



개발자 가이드

Personalize



Personalize: 개발자 가이드

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

의 상표 및 브랜드 디자인은 외 제품 또는 서비스와 함께, 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. 이 소유하지 않은 기타 모든 상표는 과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

Amazon Personalize이란 무엇인가요?	1
Amazon Personalize 요금	2
처음 사용하는 사용자를 위한 지침	2
매직 무비 머신으로 Amazon Personalize 검색	3
이 가이드 탐색	3
관련 서비스 및 솔루션 AWS	4
타사 서비스	4
자세히 알아보기	5
Amazon Personalize 및 생성형 AI	6
콘텐츠 생성기의 테마가 있는 추천	6
추천 메타데이터	7
LangChain 개인화를 위해 사전 구성된 코드	8
작동 방식	9
Personalize 워크플로우 요약	9
Personalize 용어	10
데이터 가져오기 및 관리	11
학습	13
모델 배포 및 추천	16
Personalize	17
상호작용 데이터	17
항목 데이터	18
사용자 데이터	18
작업 데이터	18
작업 상호 작용 데이터	19
Personalize 설정	20
가입하여 다음을 수행하십시오. AWS 계정	20
관리자 액세스 권한이 있는 사용자 생성	21
리전 및 엔드포인트	22
권한 설정	22
사용자에게 Amazon Personalize에 액세스할 수 있는 권한 부여	23
Personalize에 리소스에 액세스할 수 있는 권한 부여	25
Personalize에 S3 리소스에 대한 액세스 권한 부여	28
Amazon Personalize에 AWS KMS 키 사용 권한 부여	34
설정하기 AWS CLI	36

AWS SDK 설정	37
시작하기	38
시작하기 전제 조건	39
학습 데이터 생성(도메인 데이터세트 그룹)	39
학습 데이터 생성(사용자 지정 데이터세트 그룹)	40
도메인 데이터 세트 그룹 시작하기	41
도메인 데이터세트 그룹 시작하기(콘솔)	41
도메인 데이터 세트 그룹 시작하기(Java 2.x용 SDK)	52
도메인 데이터 세트 그룹 시작하기(Python용 SDK(Boto3))	61
도메인 데이터 세트 그룹 시작하기(JavaScript v3용 SDK)	66
사용자 지정 데이터세트 그룹 시작하기	74
시작하기(콘솔)	75
시작하기(AWS CLI)	86
시작하기(Python용 SDK(Boto3))	96
시작하기(Java 2.x용 SDK)	102
리소스 정리	113
리소스 정리 가이드라인	114
도메인 기반 리소스 정리	114
사용자 지정 리소스 정리	115
데이터세트 및 스키마	117
데이터세트	118
항목 상호 작용 데이터 세트	119
사용자 데이터 세트	124
항목 데이터 세트	124
작업 데이터 세트	128
작업 상호 작용 데이터 세트	130
스키마	132
스키마 서식 지적 요구 사항	132
도메인 데이터세트 및 스키마	134
사용자 지정 데이터세트 및 스키마	149
Python을 사용하여 스키마 생성하기	162
데이터 형식 지침	163
대량 데이터 형식 지침 및 요구 사항	164
상호작용 데이터 예제	164
명시적 노출의 형식 지정	166
범주형 데이터의 형식 지정	167

도메인 사용 사례 및 사용자 지정 레시피	168
사용 사례 및 레시피 기능	168
실시간 개인 맞춤	168
탐색	169
자동 업데이트	170
사용 사례 선택	172
VIDEO_ON_DEMAND 사용 사례	172
ECOMMERCE 사용 사례	176
레시피 선택	180
사용 사례별 Personalize 레시피 유형	181
Personalize 레시피	182
사용 가능한 Personalize 레시피 보기	184
USER_PERSONALIZATION	184
POPULAR_ITEMS	220
PERSONALIZED_RANKING	224
RELATED_ITEMS	233
PERSONALIZED_ACTIONS	242
USER_SEGMENTATION	247
준비 체크리스트	252
사용 사례를 Personalize 리소스와 일치시켰습니까?	252
항목 상호 작용 데이터가 충분합니까?	253
실시간 이벤트 스트리밍 아키텍처가 마련되어 있습니까?	253
데이터가 Personalize에 최적화되어 있습니까?	254
추천을 개선할 수 있는 선택적 데이터를 수집합니까?	254
추천을 테스트할 계획이 있으십니까?	255
추가 비즈니스 목표가 있나요?	255
Personalize 워크플로우	256
1단계: 데이터세트 그룹 생성	257
데이터세트 그룹 생성(콘솔)	258
데이터세트 그룹 생성(AWS CLI)	258
데이터세트 그룹 (SDK) 생성AWS	259
단계 2: 데이터 준비 및 가져오기	261
대량 데이터 준비 및 가져오기	262
개별 레코드 가져오기	296
3단계: 추천 또는 사용자 지정 리소스 생성	311
도메인 추천 생성	312

사용자 지정 리소스 생성	341
4단계: 추천 받기	400
추천 접수	400
실시간 추천 받기	401
배치 권장 사항 및 사용자 세그먼트(사용자 지정 리소스)	428
추천 관련성 유지	462
데이터세트를 최신 상태로 유지	462
도메인 추천자 유지 관리	462
사용자 지정 솔루션 유지 관리	463
이벤트 기록	464
실시간 이벤트가 추천에 미치는 영향	465
항목 상호 작용 이벤트 기록	465
항목 상호 작용 이벤트 기록 및 모델 교육을 위한 요구 사항	466
항목 상호 작용 이벤트 트래커 생성	467
PutEvents 오퍼레이션 사용	469
이벤트 지표 및 어트리뷰션 보고서	477
작업 상호 작용 이벤트 기록	479
작업 상호 작용 이벤트를 기록하기 위한 요구 사항	479
작업 상호 작용 이벤트 트래커 ID	480
오퍼레이션 사용 PutActionInteractions	480
익명 사용자를 위한 이벤트 기록	483
익명 사용자를 위한 지속적인 이벤트 이력 작성	484
타사 이벤트 추적 서비스	485
샘플 구현	485
데이터 관리	486
새 데이터가 실시간 추천에 영향을 미치는 방식	487
새 상호작용	487
새 항목	488
신규 사용자	488
새로운 작업	489
데이터세트로 더 많은 훈련 데이터 가져오기	489
개별 가져오기 작업을 통한 데이터 가져오기	490
기존 대량 데이터 업데이트	490
데이터 분석	491
데이터 분석을 위한 필수 권한	491
데이터 인사이트	492

데이터세트 인사이트 및 통계 보기	494
데이터 내보내기	495
데이터세트 내보내기 작업 권한 요구 사항	496
데이터세트 내보내기 작업 생성	498
데이터세트 스키마 바꾸기	503
지침 및 요구 사항	504
데이터세트 스키마 교체(콘솔)	505
데이터세트의 스키마 교체(AWS CLI)	506
데이터세트의 스키마 (AWS SDK) 바꾸기	506
사용자 삭제	507
지침 및 요구 사항	507
삭제할 사용자 목록 준비	508
데이터 삭제 작업 생성	509
데이터 세트 삭제	513
데이터세트 삭제(콘솔)	513
데이터세트 삭제(AWS CLI)	513
데이터세트 삭제(AWS SDK)	514
결과 필터링	515
필터 표현식	516
지침 및 요구 사항	517
필터 표현식 구조 및 요소	518
필터 표현식 예제	520
실시간 추천 필터링	526
실시간 추천 필터링(콘솔)	526
실시간 추천 필터링(AWS CLI)	532
실시간 추천 (AWS SDK) 필터링	533
배치 추천 및 사용자 세그먼트(사용자 지정 리소스) 필터링	539
입력 JSON에서 필터 값 제공	539
배치 워크플로우 필터링(콘솔)	540
배치 워크플로우 필터링(AWS SDK)	541
추천의 영향 측정	542
지표 어트리뷰션을 통한 추천 영향력 측정	542
지침 및 요구 사항	543
지표 어트리뷰션 생성	546
지표 어트리뷰션 관리	553
결과 게시 및 보기	562

A/B 테스트를 통한 권장 사항 영향 측정	568
A/B 테스트 모범 사례	568
CloudWatch를 사용한 A/B 테스트	569
검색 결과 개인화: OpenSearch	573
사용 사례 예제	574
검색 워크플로우	574
Personalize 검색 순위 플러그인 작동 방식	575
추가 정보	576
지침 및 요구 사항	576
플러그인 요구 사항	576
아마존 OpenSearch 서비스 권한 설정	577
오픈 소스 권한 설정 OpenSearch	582
플러그인 설정 OpenSearch 및 설치	584
아마존 OpenSearch 서비스 설정	584
오픈 소스 설정 OpenSearch	585
플러그인 구성	587
personalized_search_ranking 응답 프로세서에 대한 필드	588
Amazon OpenSearch 서비스로 파이프라인 생성	589
오픈소스로 파이프라인 만들기 OpenSearch	590
OpenSearch 쿼리에 플러그인 적용	590
Amazon OpenSearch 서비스 쿼리에 플러그인 적용	591
오픈소스의 쿼리에 플러그인 적용 OpenSearch	592
OpenSearch 결과를 플러그인 결과와 비교	594
아마존 OpenSearch 서비스와의 결과 비교	594
결과를 오픈소스와의 비교 OpenSearch	596
플러그인 모니터링	597
Amazon OpenSearch 서비스를 통한 플러그인 모니터링	597
오픈 소스로 플러그인 모니터링 OpenSearch	598
파이프라인 지표 예제	598
리소스에 태그 지정	600
지침 및 요구 사항	601
추가 정보	601
Personalize 리소스에 태그 추가	602
태그 추가(콘솔)	602
태그 추가(AWS CLI)	603
태그 추가 (AWS SDK)	603

Amazon Personalize 리소스에서 태그 제거	607
태그 제거 (콘솔)	607
태그 제거 (AWS CLI)	608
태그 제거 (AWS SDK)	608
IAM 정책에서 태그 사용	608
문제 해결	611
자주 묻는 질문(FAQ)	611
데이터 가져오기 및 관리	611
사용자 지정 솔루션 및 솔루션 버전 생성	613
모델 배포(사용자 지정 캠페인)	613
추천	614
추천 필터링	614
오류 메시지	615
데이터 가져오기 및 관리	615
솔루션 및 솔루션 버전 생성(사용자 지정 리소스)	616
모델 배포(사용자 지정 캠페인)	617
추천자(도메인 데이터 세트 그룹)	617
추천	617
추천 필터링	618
AWS CloudFormation를 사용하여 리소스 지정하기	619
Amazon Personalize 및 AWS CloudFormation 템플릿	619
Amazon Personalize 리소스의 예제 AWS CloudFormation 템플릿	619
CreateDatasetGroup	620
CreateDataset	620
CreateSchema	621
CreateSolution	622
AWS CloudFormation에 대해 자세히 알아보기	623
보안	624
데이터 보호	624
데이터 암호화	625
ID 및 액세스 관리	626
고객	627
ID를 통한 인증	627
정책을 사용한 액세스 관리	630
IAM으로 Personalize를 사용하는 방법	632
교차 서비스 혼동된 대리인 방지	639

자격 증명 기반 정책 예시	640
문제 해결	645
로그 및 모니터링	647
모니터링	647
CloudWatch 아마존 퍼스널라이즈용 메트릭	651
를 사용하여 Amazon Personalize API 호출을 로깅합니다. AWS CloudTrail	656
규정 준수 확인	658
복원력	659
인프라 보안	659
VPC 엔드포인트(AWS PrivateLink)	660
Personalize에 대한 인터페이스 VPC 엔드포인트 생성	661
ECS에 대한 VPC 엔드포인트 정책 생성	661
엔드포인트 및 할당량	663
Personalize 엔드포인트 및 리전	663
규정 준수	663
Service quotas	663
할당량 증가 요청	671
API 참조	673
작업	673
Personalize	676
Amazon Personalize 이벤트	914
Amazon Personalize 런타임	929
데이터 타입	945
Personalize	948
Amazon Personalize 이벤트	1099
Amazon Personalize 런타임	1114
일반적인 오류	1119
공통 파라미터	1121
문서 이력	1124
AWS 용어집	1137
.....	mcxxxviii

Amazon Personalize이란 무엇인가요?

Amazon Personalize는 데이터를 사용하여 사용자를 위한 항목 추천을 생성하는 완전 관리형 기계 학습 서비스입니다. 또한 특정 항목 또는 항목 메타데이터에 대한 사용자의 선호도를 기반으로 사용자 세그먼트를 생성할 수 있습니다.

일반적인 사용 사례는 다음과 같습니다.

- 동영상 스트리밍 앱 개인 맞춤 — 사전 구성되거나 사용자 지정이 가능한 Amazon Personalize 리소스를 사용하여 스트리밍 앱에 여러 유형의 개인 맞춤형 동영상 추천을 추가할 수 있습니다. 예: 가장 적합한 추천 제품, X와 유사한 제품, 가장 인기 있는 제품 동영상 추천 등.
- 전자상거래 앱에 제품 추천 추가 — 사전 구성되거나 사용자 지정이 가능한 Amazon Personalize 리소스를 사용하여 여러 유형의 개인 맞춤형 제품 추천을 소매 앱에 추가할 수 있습니다. 예: 추천 제품, 자주 함께 구매한 제품 및 X를 본 고객도 보는 제품 제품 추천.
- 앱에 실시간 차선책 작업 추천 추가 — 사용자 지정 가능한 Amazon Personalize 리소스를 사용하여 사용자가 행동을 기반으로 행할 가능성이 가장 높은 작업을 추천할 수 있습니다. 예를 들어, 로열티 프로그램 등록, 모바일 앱 다운로드 또는 홍보 이메일 구독을 위한 실시간 추천을 추가할 수 있습니다.
- 개인 맞춤형 이메일 생성 — 사용자 지정 가능한 Amazon Personalize 리소스를 사용하여 이메일 목록에 있는 모든 사용자를 위한 배치 추천을 생성할 수 있습니다. 그런 다음 [AWS 서비스](#) 또는 [타사 서비스](#)를 사용하여 카탈로그의 항목을 추천하는 개인 맞춤형 이메일을 사용자에게 보낼 수 있습니다.
- 타겟 마케팅 캠페인 생성 — Amazon Personalize를 사용하여 카탈로그의 항목과 상호작용할 가능성이 가장 높은 사용자 세그먼트를 생성할 수 있습니다. 그런 다음 [AWS 서비스](#) 또는 [타사 서비스](#)를 사용하여 다양한 항목을 다양한 사용자 세그먼트에 홍보하는 타겟 마케팅 캠페인을 만들 수 있습니다.
- 검색 결과 개인 맞춤 — 사용자 지정 가능한 Amazon Personalize 리소스를 사용하여 사용자에게 맞게 검색 결과를 개인 맞춤할 수 있습니다. 예를 들어 Amazon Personalize는 사용자가 생성한 검색 결과의 순위를 다시 매길 수 있습니다. [OpenSearch](#)

대부분의 사용 사례에서 Amazon Personalize는 주로 항목 상호 작용 데이터를 기반으로 추천을 생성합니다. 항목 상호 작용 데이터는 사용자가 카탈로그의 항목과 상호 작용하면서 생성됩니다. 사용자가 다른 항목을 클릭하는 경우를 예로 들 수 있습니다. 항목 상호 작용 데이터는 CSV 파일의 과거 대량 상호 작용 레코드와 사용자가 카탈로그와 상호 작용할 때 발생한 실시간 이벤트 모두에서 가져올 수 있습니다. Amazon Personalize는 장르, 가격, 성별 등 항목과 사용자의 데이터를 사용하기도 합니다. 또한 차선책 작업 시나리오에서는 작업 및 작업 상호 작용 데이터를 사용합니다.

대량 데이터를 가져올 때 Amazon 데이터 랭글러를 사용하여 40개 이상의 소스에서 SageMaker 데이터를 가져와서 Amazon Personalize에 사용할 수 있도록 준비할 수 있습니다. 자세한 설명은 [Amazon SageMaker 데이터 랭글러를 사용하여 데이터 준비 및 가져오기](#) 섹션을 참조하세요.

Amazon Personalize에는 실시간 개인 맞춤을 위한 API 작업과 대량 추천 및 사용자 세그먼트를 위한 배치 작업이 포함되어 있습니다. 비즈니스 도메인에 대한 사용 사례 최적화 추천으로 빠르게 시작하거나 구성 가능한 사용자 지정 리소스를 직접 생성할 수 있습니다.

주제

- [Amazon Personalize 요금](#)
- [Amazon Personalize를 처음 사용하는 사용자를 위한 지침](#)
- [관련 서비스 및 솔루션 AWS](#)
- [타사 서비스](#)
- [자세히 알아보기](#)

Amazon Personalize 요금

Amazon Personalize에는 최소 요금 및 선수금이 없습니다. [AWS 프리 티어](#)는 사용 가능한 AWS 지역당 최대 20GB의 데이터 처리, 적격 AWS 지역당 최대 100시간의 교육 시간, 최대 180,000개의 추천 요청의 월간 할당량을 제공합니다. 프리 티어는 처음 2개월 동안 유효합니다.

전체적인 요금 및 가격 목록은 [Amazon Personalize 요금](#) 단원을 참조하세요.

Amazon Personalize를 처음 사용하는 사용자를 위한 지침

Amazon Personalize를 처음 사용하는 경우 시작하는 데 도움이 되도록 다음 리소스에서 확인할 수 있습니다.

주제

- [매직 무비 머신으로 Amazon Personalize 검색](#)
- [이 가이드 탐색](#)

매직 무비 머신으로 Amazon Personalize 검색

매직 무비 머신은 대화형 학습 경험입니다. 이를 통해 Amazon Personalize 기능을 검색하고 추천 생성에 대해 자세히 알아볼 수 있습니다. 간략한 소개는 아래 비디오를 참조하세요. 그럼 [매직 무비 머신](#)을 사용해 보세요.

[Amazon Personalize 시작하기](#)

이 가이드 탐색

다음 단원을 읽어 보면 도움이 될 것입니다.

- 작동 방식**— 이 단원에서는 Amazon Personalize 워크플로를 소개하고 사용자를 위한 개인 맞춤형 경험을 생성하는 단계를 안내합니다. 이 단원에는 일반적인 Amazon Personalize 용어 및 정의도 포함되어 있습니다. 추천을 받기 전에 이 단원부터 시작하여 Amazon Personalize 워크플로와 용어를 잘 이해하고 있어야 합니다.
- Personalize 설정**— 이 섹션에서는 Amazon Personalize를 설정하고 AWS 계정, 사용하는 데 필요한 권한을 설정하고, Amazon Personalize를 사용하고 관리하기 위한 AWS CLI 및 AWS SDK를 설정합니다.
- 시작하기**— 이 단원에서는 간단한 영화 데이터 세트로 Amazon Personalize를 사용하기 시작합니다. 이들 자습서를 완료하여 Amazon Personalize를 직접 사용해 보세요. 도메인 데이터 세트 그룹에서 시작하거나 사용자 지정 데이터 세트 그룹에서 시작하도록 선택할 수 있습니다.
 - 도메인 데이터 세트 그룹 생성을 시작하려면 [시작하기 전제 조건](#)을 완료한 다음 [도메인 데이터 세트 그룹 시작하기](#)에서 튜토리얼을 시작하세요.
 - 사용자 지정 데이터 세트 그룹을 시작하려면 [시작하기 전제 조건](#)을 완료한 다음 [도메인 데이터 세트 그룹 시작하기](#)에서 튜토리얼을 시작하세요.
- 도메인 사용 사례 및 사용자 지정 레시피**— Amazon Personalize에서 모델을 학습시키는 데 사용할 수 있는 도메인 사용 사례 및 사용자 지정 레시피에 대해 알아봅니다. 이 정보를 사용하면 사용 사례를 Amazon Personalize의 리소스와 일치시키는 데 도움이 됩니다.
- 준비 체크리스트**— 준비 체크리스트를 검토하여 자체 데이터로 Amazon Personalize를 사용할 준비를 시작합니다. 이 체크리스트는 Amazon Personalize 기능, 요구 사항 및 데이터 지침 목록을 제공합니다. 계획을 세우는 데 도움이 될 수도 있고 Amazon Personalize에서 리소스를 생성할 때 참조로 사용할 수도 있습니다.
- Personalize 워크플로우**— 이 단원에서는 Amazon Personalize의 전체 워크플로를 소개합니다. 도메인 데이터세트 그룹 또는 사용자 지정 데이터세트 그룹을 생성하고, 데이터를 준비 및 가져오고, 추천자 또는 사용자 지정 리소스를 생성하고, 추천을 받는 방법에 대한 step-by-step 지침을 제공합니다.

7. [이벤트 기록](#) – 이 섹션에서는 항목 상호 작용 및 작업 상호 작용 이벤트를 실시간으로 기록하는 방법을 다룹니다. Amazon Personalize 리소스를 설정한 후 이 섹션을 완료하여 사용자 행동에 따라 데이터 세트를 최신 상태로 유지하는 방법을 알아봅니다.
8. [추천 및 사용자 세그먼트 필터링](#)— 이 단원에서는 추천을 필터링하는 방법을 설명합니다. 이 단원을 완료하여 필터 표현식을 구성하여 사용자 지정 기준에 따라 추천을 필터링하는 방법을 알아봅니다. 예를 들어, 사용자가 이미 구입한 제품이나 이미 시청한 영화는 추천하고 싶지 않을 수 있습니다.

관련 서비스 및 솔루션 AWS

Amazon Personalize는 다른 AWS 서비스 및 솔루션과 원활하게 통합됩니다. 예를 들어, 다음을 수행할 수 있습니다.

- 아마존 SageMaker 데이터 랭글러 (데이터 랭글러) 를 사용하여 40개 이상의 소스에서 Amazon Personalize 데이터세트로 데이터를 가져올 수 있습니다. 데이터 랭글러는 데이터를 가져오고, 준비하고, 변환하고, 분석하는 end-to-end 솔루션을 제공하는 Amazon SageMaker Studio의 기능입니다. 자세한 설명은 [Amazon SageMaker 데이터 랭글러를 사용하여 데이터 준비 및 가져오기](#) 섹션을 참조하세요.
- 항목 상호 작용 AWS Amplify 이벤트를 기록하는 데 사용합니다. Amplify에는 웹 클라이언트 애플리케이션의 이벤트를 기록하기 위한 JavaScript 라이브러리가 포함되어 있습니다. 그리고 서버 코드에서 이벤트를 기록하기 위한 라이브러리도 포함되어 있습니다. 자세한 내용은 [Amplify 설명서](#)를 참조하십시오.
- [기계 학습을 통한 개인 맞춤형 경험 유지 관리](#)를 통해 Amazon Personalize 작업을 자동화하고 예약합니다. 이 AWS 솔루션 구현은 데이터 가져오기, 솔루션 버전 학습, 배치 워크플로를 비롯한 Amazon Personalize 워크플로를 자동화해줍니다.
- Amazon CloudWatch Eviveryt를 사용하여 Amazon Personalize 권장 사항에 따라 A/B 테스트를 수행하십시오. 자세한 설명은 [CloudWatch를 사용한 A/B 테스트](#) 섹션을 참조하세요.
- Amazon Pinpoint를 사용하면 타겟 마케팅 캠페인을 생성할 수 있습니다. [Amazon Pinpoint 및 Amplify를 사용하여 마케팅 이메일 캠페인 및 웹 앱에 Amazon Personalize 추천을 추가하는 방법을 보여주는 예는 Amplify를 사용한 웹 분석 단원을 참조하세요.](#)

타사 서비스

Amazon Personalize는 다양한 타사 서비스와 잘 연동됩니다.

- **Amplitude** — Amplitude를 사용하면 사용자 행동을 추적하여 사용자의 행동을 이해할 수 있습니다. Amplitude 및 Amazon Personalize를 사용하는 방법에 대한 자세한 내용은 다음 AWS 파트너 네트워크(APN) 블로그 게시물: [Amplitude와 Amazon Personalize를 통한 개인 맞춤의 효율성 측정](#)을 참조하세요.
- **Braze** — Braze를 사용하면 카탈로그의 항목을 추천하는 개인 맞춤형 이메일을 사용자에게 보낼 수 있습니다. Braze는 시장을 선도하는 메시징 플랫폼(이메일, 푸시, SMS)입니다. Amazon Personalize와 Braze를 통합하는 방법을 보여주는 워크숍은 [Amazon Personalize 워크숍](#)을 참조하세요.
- **mParticle** — mParticle을 사용하면 앱에서 이벤트 데이터를 수집할 수 있습니다. mParticle과 Amazon Personalize를 사용하여 개인 맞춤형 제품 추천을 구현하는 방법을 보여주는 예제는 [기계 학습에 CDP의 성능을 활용하는 방법: 2부](#)를 참조하세요.
- **Optimizely**를 사용하면 Amazon Personalize 추천에서 A/B 테스트를 수행할 수 있습니다. Optimizely와 Amazon Personalize를 사용하는 방법에 대한 자세한 내용은 [강력한 기계 학습과 실험을 결합하기 위한 Optimizely와 Amazon Personalize의 통합](#)을 참조하세요.
- **세그먼트** — 세그먼트를 사용하면 Amazon Personalize로 데이터를 보낼 수 있습니다. 세그먼트를 Amazon Personalize와 통합하는 방법에 대한 자세한 내용은 [Amazon Personalize 목적지](#)를 참조하세요.

전체 파트너 목록은 [Amazon Personalize 파트너](#)를 참조하세요.

자세히 알아보기

Amazon Personalize에 대한 추가 정보는 다음 리소스에서 확인할 수 있습니다.

- Amazon Personalize가 사용 사례에 적합한지 판단하는 데 도움이 되는 참조 요약 정보는 [Amazon Personalize 샘플 리포지토리의 Amazon Personalize 치트 시트](#)를 참조하세요.
- Amazon Personalize를 사용하는 방법에 대한 동영상 시리즈는 에 있는 Amazon [Personalize 심층 분석 동영상](#) 시리즈를 참조하십시오. YouTube
- [자세한 자습서 및 코드 샘플은 리포지토리를 참조하십시오. amazon-personalize-samples GitHub](#)

Amazon Personalize 및 생성형 AI

Amazon Personalize는 생성형 인공지능(생성형 AI)과 함께 사용할 수 있습니다. Amazon Personalize 콘텐츠 생성기는 생성형 AI의 도움을 받아 관련 항목에 대한 배치 추천에 흥미로운 테마를 추가할 수 있습니다. 콘텐츠 생성기는 Amazon Personalize에서 관리하는 생성형 AI 기능입니다.

또한 Amazon Personalize 추천을 사용하여 Amazon Personalize를 생성형 AI 워크플로와 통합하고 사용자 경험을 개선할 수 있습니다. 예를 들어, 생성형 AI 프롬프트에 추천을 추가하여 각 사용자의 관심사에 맞는 마케팅 콘텐츠를 생성할 수 있습니다. 추천 콘텐츠에 대한 간결한 요약 생성하거나 채팅 봇을 통해 제품 또는 콘텐츠를 추천할 수도 있습니다.

다음 동영상은 Amazon Personalize 및 제너레이티브 AI를 사용하여 권장 사항을 향상시키는 방법을 보여줍니다.

[Amazon Personalize 및 제너레이티브 AI를 통한 권장 사항 개선](#)

다음 Amazon Personalize 기능은 생성형 AI를 사용하거나 개인 맞춤형 콘텐츠를 만드는 생성형 AI 솔루션을 구축하는 데 도움이 될 수 있습니다. [Amazon Personalize를 제너레이티브 AI와 함께 사용하는 방법을 보여주는 샘플 Jupyter 노트북은 Amazon Personalize 샘플 리포지토리의 Amazon Personalize를 사용한 제너레이티브 AI를 참조하십시오.](#)

주제

- [콘텐츠 생성기의 테마가 있는 추천](#)
- [추천 메타데이터](#)
- [LangChain 개인화를 위해 사전 구성된 코드](#)

콘텐츠 생성기의 테마가 있는 추천

Amazon Personalize 콘텐츠 생성기는 설명 테마를 배치 추천에 추가할 수 있습니다. 콘텐츠 생성기는 Amazon Personalize에서 관리하는 생성형 AI 기능입니다.

테마가 있는 배치 추천을 받으면 Amazon Personalize 콘텐츠 생성기가 유사한 각 항목 세트에 대한 설명 테마를 추가합니다. 예를 들어, 아침 식사 음식과 유사한 항목을 추천받는 경우 Amazon Personalize는 개운하게 일어나기 또는 아침 필수품과 같은 테마를 생성할 수 있습니다. 이 테마를 사용하여 함께 자주 구매하는 제품과 같은 일반적인 캐러셀 제목을 대체할 수 있습니다. 또는 새 메뉴 옵션에 대한 홍보 이메일이나 마케팅 캠페인에 테마를 통합할 수도 있습니다.

테마를 생성하려면 데이터를 항목 상호 작용 및 항목 데이터 세트로 가져오고, 유사-항목 레시피로 사용자 지정 솔루션을 만들고, 배치 추천을 생성합니다. 항목 데이터에는 항목 설명 및 제목 정보가 포함되어야 합니다. 자세한 항목 설명과 제목은 콘텐츠 생성기가 더 정확하고 흥미로운 테마를 만드는 데 도움이 됩니다.

- Amazon Personalize 워크플로에 대한 자세한 내용은 [Personalize 워크플로우](#) 섹션을 참조하세요.
- 배치 추천에 대한 자세한 내용은 [배치 권장 사항 및 사용자 세그먼트\(사용자 지정 리소스\)](#) 섹션을 참조하세요.
- 테마가 있는 추천을 생성하는 방법에 대한 자세한 내용은 [콘텐츠 생성기에서 테마가 있는 배치 추천](#) 섹션을 참조하세요.

추천 메타데이터

추천을 받으면 Amazon Personalize가 항목 데이터 세트의 각 추천 항목에 대한 메타데이터를 반환하도록 할 수 있습니다. Amazon Personalize 추천과 함께 이 메타데이터를 생성형 AI 프롬프트에 추가하여 더욱 매력적인 콘텐츠를 생성할 수 있습니다.

예를 들면, 생성형 AI를 사용하여 마케팅 이메일을 만들 수 있습니다. Amazon Personalize 추천 및 해당 메타데이터(예: 영화 장르)를 생성형 AI용 프롬프트 엔지니어링의 하나로 사용할 수 있습니다. 개인 맞춤형 프롬프트를 사용하면 생성형 AI를 통해 각 고객의 관심사에 맞는 매력적인 마케팅 이메일을 생성할 수 있습니다.

추천 메타데이터를 가져오려면 먼저 Amazon Personalize 워크플로를 완료하여 데이터를 가져오고 도메인 또는 사용자 지정 리소스를 생성해야 합니다. Amazon Personalize 추천자 또는 캠페인을 생성할 때 추천에 메타데이터를 포함하는 옵션을 활성화하세요. 추천을 받으면 포함하려는 항목 데이터 열을 지정할 수 있습니다.

- Amazon Personalize 워크플로에 대한 자세한 내용은 [Personalize 워크플로우](#) 섹션을 참조하세요.
- 추천자용 메타데이터 활성화에 대한 자세한 내용은 [추천의 메타데이터 활성화\(도메인 리소스\)](#)를 참조하세요.
- 캠페인용 메타데이터 활성화에 대한 자세한 내용은 [추천의 메타데이터 활성화\(사용자 지정 리소스\)](#)를 참조하세요.
- Amazon Personalize를 생성형 AI와 함께 사용하여 마케팅 캠페인을 생성하는 방법에 대한 자세한 내용은 [Amazon Personalize 및 생성형 AI를 통한 마케팅 솔루션 개선](#)을 참조하세요.

LangChain 개인화를 위해 사전 구성된 코드

LangChain 언어 모델을 기반으로 애플리케이션을 개발하기 위한 프레임워크입니다. Amazon Personalize용으로 빌드된 코드가 특징입니다. 이 코드를 사용하여 Amazon Personalize 추천을 생성형 AI 솔루션과 통합할 수 있습니다.

예를 들면, 다음 코드를 사용하여 사용자를 위한 Amazon Personalize 추천을 체인에 추가할 수 있습니다.

```
from aws_langchain import AmazonPersonalize
from aws_langchain import AmazonPersonalizeChain
from langchain.llms.bedrock import Bedrock

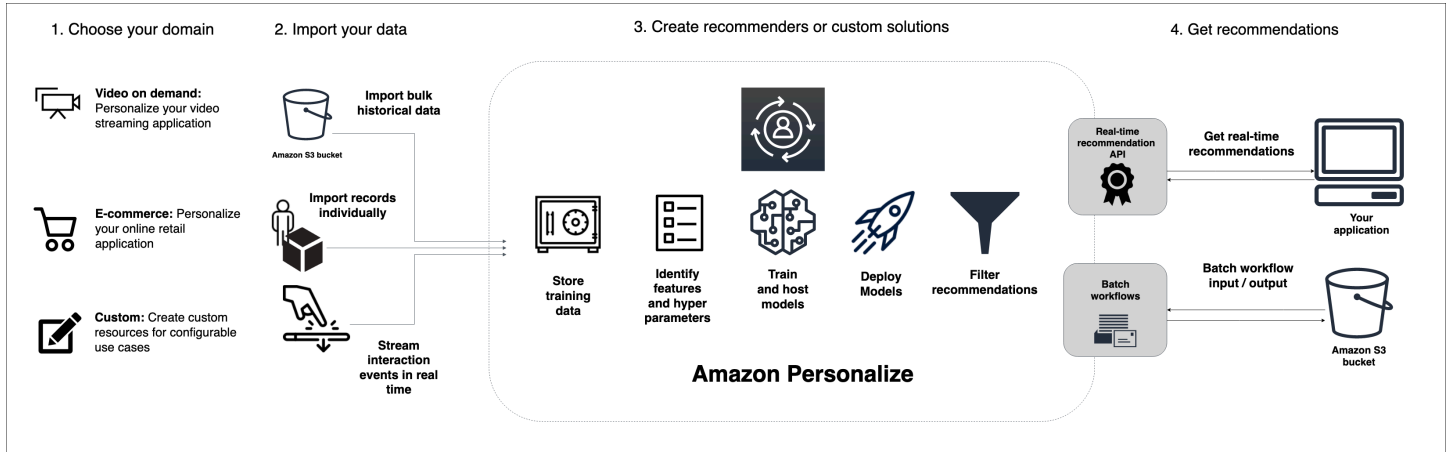
recommender_arn="RECOMMENDER_ARN"

bedrock_llm = Bedrock(model_id="anthropic.claude-v2", region_name="us-west-2")
client=AmazonPersonalize(credentials_profile_name="default", region_name="us-west-2",recommender_arn=recommender_arn)
# Create personalize chain
# Use return_direct=True if you do not want summary
chain = AmazonPersonalizeChain.from_llm(
    llm=bedrock_llm,
    client=client,
    return_direct=False
)
response = chain({'user_id': '1'})
print(response)
```

- 시작하는 LangChain 방법에 대한 자세한 내용은 LangChain 설명서의 [소개를](#) 참조하십시오.
- [고급 LangChain 코드 샘플을 포함하여 Amazon Personalize용으로 구축된 코드를 사용하는 방법에 대한 자세한 내용은 샘플 리포지토리의 AWS Amazon LangChain Personalize 확장 프로그램을 참조하십시오.](#)

작동 방식

Personalize는 사용자 데이터를 사용하여 도메인 기반 또는 사용자 지정 가능한 추천 모델을 학습시킵니다. 애플리케이션에서 비공개 추천 API를 사용하여 실시간 추천을 요청합니다. Personalize는 배치 워크플로우가 항목 추천 및 사용자 세그먼트를 받는 것도 지원합니다.



주제

- [Personalize 워크플로우 요약](#)
- [Personalize 용어](#)
- [Personalize가 사용할 수 있는 데이터 유형](#)

Personalize 워크플로우 요약

Personalize 워크플로는 다음과 같습니다.

1. 데이터세트 그룹 생성

데이터세트 그룹은 Personalize 리소스를 위한 컨테이너입니다. 생성하는 데이터세트 그룹 유형에 따라 Personalize 워크플로의 3단계에서 생성할 수 있는 리소스가 결정됩니다.

- 도메인 데이터세트 그룹을 사용하면 VIDEO_ON_DEMAND 또는 ECOMMERCE 도메인 사용 사례에 대한 추천자를 생성할 수 있습니다. Personalize는 이러한 추천자의 구성, 학습 및 업데이트를 관리합니다. 도메인 데이터세트 그룹으로 시작하는 경우에도 사용자 지정 리소스를 추가할 수 있습니다. 도메인 데이터세트 그룹에서는 액션 및 액션 인터랙션 데이터세트를 비롯한 차선택 액션 리소스를 만들 수 없습니다.

- 사용자 지정 데이터세트 그룹을 사용하면 사용자 지정 리소스만 생성할 수 있습니다. 여기에는 솔루션, 솔루션 버전, 캠페인이 포함됩니다. 이들 리소스의 경우 구성, 업데이트 및 재학습을 더 잘 제어할 수 있습니다.

2. [데이터 준비 및 가져오기](#)

상호 작용, 항목, 사용자, 작업, 작업 상호 작용 레코드를 데이터 세트(데이터용 Amazon Personalize 컨테이너)로 가져옵니다. 레코드를 대량으로 또는 개별적으로 가져올 수 있습니다. 대량 데이터를 가져올 때 Amazon 데이터 랭글러를 사용하여 40개 이상의 소스에서 SageMaker 데이터를 가져와서 Amazon Personalize에 사용할 수 있도록 준비할 수 있습니다. 자세한 정보는 [Amazon SageMaker 데이터 랭글러를 사용하여 데이터 준비 및 가져오기](#)을 참조하세요.

Personalize 데이터세트로 데이터를 가져온 후에는, 데이터를 분석하고, S3 버킷으로 내보내고, 업데이트하거나, 데이터세트를 삭제하여 데이터를 삭제할 수 있습니다. 자세한 내용은 [데이터셋의 훈련 데이터 관리](#) 단원을 참조하세요.

3. [도메인 추천자 또는 사용자 지정 리소스 생성](#)

데이터를 가져온 후에는 도메인 추천자(도메인 데이터세트 그룹의 경우) 또는 사용자 지정 리소스(사용자 지정 데이터세트 그룹의 경우)를 만들어 데이터에 따라 모델을 학습시킵니다. 이들 리소스를 사용하여 추천을 생성할 수 있습니다.

4. [추천 받기](#)

추천자 또는 사용자 지정 캠페인을 사용하여 추천을 받습니다. 사용자 지정 데이터세트 그룹을 사용하면 배치 추천 또는 사용자 세그먼트를 가져올 수도 있습니다.

Personalize 워크플로를 처음으로 완료한 후에는 데이터를 최신 상태로 유지하고 모든 사용자 지정 솔루션을 정기적으로 재학습시킵니다. 이를 통해 모델은 사용자의 가장 최근 활동을 학습하고 추천의 관련성을 유지하고 개선할 수 있습니다. 자세한 내용은 [추천 관련성 유지](#) 단원을 참조하세요.

Personalize 용어

이 단원에서는 Personalize에서 사용되는 용어를 소개합니다.

주제

- [데이터 가져오기 및 관리](#)
- [학습](#)
- [모델 배포 및 추천](#)

데이터 가져오기 및 관리

다음 용어는 Personalize의 데이터 가져오기, 내보내기 및 형식 지정과 관련이 있습니다.

액션 데이터셋

액션에 대한 메타데이터를 담은 컨테이너입니다. 액션은 모바일 앱 설치, 로열티 프로그램 가입 등 사용자에게 추천할 수 있는 참여 또는 수익 창출 활동입니다. 액션의 메타데이터에는 액션의 만료 타임스탬프, 값, 반복 빈도 데이터, 범주별 메타데이터가 포함될 수 있습니다. 이 유형의 데이터는 [에서만](#) 사용됩니다. [차선책-작업 레시피](#)

액션 인터랙션 데이터셋

사용자와 액션 간의 상호작용을 통해 수집하는 과거 및 실시간 데이터를 담은 컨테이너입니다. 각 작업 상호 작용은 userID, actionID, 타임스탬프, 이벤트 유형 및 상호 작용에 대한 추가 데이터(예: 범주별 메타데이터)로 구성됩니다. 이 유형의 데이터는 [에서만](#) 사용됩니다. [차선책-작업 레시피](#).

컨텍스트 메타데이터

이벤트(예: 클릭)가 발생할 때 사용자의 브라우징 컨텍스트(예: 사용한 디바이스 또는 위치)에 대해 수집하는 상호작용 데이터. 컨텍스트 메타데이터는 신규 및 기존 사용자에게 대한 추천 관련성을 개선할 수 있습니다.

데이터셋

Personalize에 업로드하는 데이터를 위한 컨테이너입니다. Amazon Personalize 데이터 세트에는 사용자, 아이템, 항목 상호 작용 데이터 세트 및 작업의 다섯 가지 유형이 있습니다.

데이터셋 그룹

데이터셋, 도메인 추천자, 사용자 지정 리소스를 포함한 Personalize 리소스를 위한 컨테이너입니다. 데이터셋 그룹에서는 리소스가 독립적인 컬렉션으로 정리되어 있는데, 한 데이터셋 그룹의 리소스는 다른 데이터셋 그룹의 리소스에 영향을 미칠 수 없습니다. 데이터셋 그룹은 도메인 데이터셋 그룹 또는 사용자 지정 데이터셋 그룹일 수 있습니다.

도메인 데이터셋 그룹

다양한 비즈니스 도메인 및 사용 사례에 맞게 사전 구성된 리소스가 포함된 데이터셋 그룹입니다. Personalize는 학습 모델 및 배포의 수명 주기를 관리합니다. 도메인 데이터셋 그룹을 생성할 때는 비즈니스 도메인을 선택하고, 데이터를 가져오고, 각 사용 사례에 대한 추천자를 생성합니다. 애플리케이션에서 추천자를 사용하여 작업과 관련된 권장 사항을 얻을 수 있습니다.

GetRecommendations

도메인 데이터세트 그룹으로 시작하는 경우에도 사용자 지정 사용 사례에 대한 레시피로 학습한 솔루션 및 솔루션 버전과 같은 사용자 지정 리소스를 추가할 수 있습니다.

사용자 지정 데이터세트 그룹

솔루션, 솔루션 버전, 필터, 캠페인, 배치 추론 작업 등 사용자 지정 리소스만 포함하고 있는 데이터세트 그룹입니다. 캠페인을 사용하여 GetRecommendations 운영과 관련된 추천을 받을 수 있습니다. 학습 모델 및 배포의 수명 주기를 관리합니다. 사용자 지정 데이터세트 그룹으로 시작했다면 나중에 도메인과 연결할 수 없습니다. 대신 새 도메인 데이터세트 그룹을 생성하세요.

데이터세트 내보내기 작업

데이터세트의 레코드를 S3 버킷에 있는 하나 이상의 CSV 파일로 출력하는 레코드 내보내기 도구입니다. 출력 CSV 파일에는 데이터세트 스키마의 필드와 일치하는 열 이름을 가진 헤더 행이 포함됩니다.

데이터세트 가져오기 작업

S3 버킷에 있는 CSV 파일의 데이터로 Personalize 데이터세트를 채우는 대량 가져오기 도구입니다.

이벤트

클릭, 구매 또는 비디오 시청과 같은 사용자 작업으로, Amazon Personalize 항목 상호 작용 데이터 세트에 이를 기록하고 업로드합니다. CSV 파일에서 이벤트를 대량으로 가져오고, Personalize 콘솔을 사용하여 점진적으로, 실시간으로 이벤트를 가져올 수 있습니다.

명시적 노출

Amazon Personalize 항목 상호 작용 데이터 세트에 수동으로 추가하는 항목 목록입니다. Personalize가 추천 데이터에서 자동으로 추출하는 암시적 노출과 달리 명시적 노출에 포함할 내용은 사용자가 선택합니다.

암시적 노출

애플리케이션이 사용자에게 보여주는 추천. 항목 상호 작용 데이터 세트에 수동으로 추가하는 명시적 노출과 달리 Amazon Personalize는 추천 데이터에서 암시적 노출을 자동으로 도출합니다.

노출 데이터

사용자가 특정 항목을 클릭, 시청, 구매 등을 통해 특정 항목과 상호작용할 때 사용자에게 제시되는 항목 목록입니다. Personalize는 노출 데이터를 사용하여 사용자가 동일한 품목을 선택하거나 무시한 빈도를 기반으로 사용자에 대한 새 항목의 관련성을 계산합니다.

이벤트 상호작용 데이터세트

사용자와 항목 간의 상호작용([이벤트](#)라고 함)을 통해 수집하는 과거 및 실시간 데이터를 저장하는 컨테이너입니다. 상호작용 데이터에는 이벤트 유형 데이터와 [상황별](#) 메타데이터가 포함될 수 있습니다.

항목 데이터세트

가격, 장르, 재고 여부 등 항목에 대한 메타데이터를 담은 컨테이너입니다.

반복 빈도

Actions 데이터세트로 가져올 수 있는 액션 메타데이터 유형입니다. 반복 빈도 데이터는 작업 상호 작용 데이터 세트의 사용자 기록을 기반으로 사용자가 상호 작용한 후 Amazon Personalize가 특정 작업을 추천할 때까지 기다려야 하는 기간을 지정합니다.

스키마

Personalize에 데이터 구조를 알려주는 [Apache Avro](#) 형식의 JSON 객체입니다. Personalize는 스키마를 사용하여 데이터를 파싱합니다.

사용자 데이터세트

연령, 성별, 로열티 멤버십과 같은 사용자에 대한 메타데이터를 담은 컨테이너입니다.

학습

다음 용어는 Personalize에서 모델을 학습시키는 것과 관련이 있습니다.

item-to-item 유사성 (SIMS) 레시피

상호작용 데이터세트의 데이터를 사용하여 지정된 항목과 유사한 항목을 추천하는 [RELATED_ITEMS](#) 레시피입니다. SIMS 레시피는 가격이나 색상과 같은 항목 메타데이터를 일치시키는 대신 사용자가 항목과 상호작용하는 방식을 기반으로 유사도를 계산합니다.

항목-친화도

항목 상호 작용 데이터 세트와 항목 데이터 세트의 데이터를 사용하여 사용자가 항목과 상호 작용할 가능성에 따라 지정하는 각 항목에 대한 사용자 세그먼트를 만드는 USER_SEGMENTATION 레시피입니다.

항목-속성-친화도

항목 상호 작용 데이터 세트와 항목 데이터 세트의 데이터를 사용하여 사용자가 속성을 통해 항목과 상호 작용할 가능성에 따라 지정하는 각 항목 속성에 대한 사용자 세그먼트를 만드는 USER_SEGMENTATION 레시피입니다.

차선택-작업 레시피

이 레시피는 사용자를 위한 차선택을 위한 실시간 추천을 생성합니다. 사용자를 위한 차선택 작업은 사용자가 수행할 가능성이 가장 높은 작업입니다. 로열티 프로그램에 등록하거나, 앱을 다운로드하거나, 신용카드를 신청하는 경우를 예로 들 수 있습니다. 자세한 내용은 [차선택-작업 레시피](#) 단원을 참조하세요.

맞춤형 랭킹 V2 레시피

[PERSONALIZED_RANKING](#) 레시피로, 특정 사용자의 예상 관심 수준을 기반으로 제공하는 항목 컬렉션의 순위를 매깁니다. 이 레시피는 변환기 기반 아키텍처를 사용하여 항목 상호 작용 데이터, 항목 메타데이터 및 사용자 메타데이터에서 학습하는 모델을 학습시킵니다. Personalized-Ranking-v2 레시피를 사용하여 특정 사용자에게 맞게 개인화된 항목 또는 검색 결과의 선별된 목록 순서를 개인화할 수 있습니다. 최대 5백만 개의 항목을 학습할 수 있으며 이전 버전보다 짧은 지연 시간으로 관련성이 더 높은 추천을 생성할 수 있습니다.

개인 맞춤형-순위 레시피

[PERSONALIZED_RANKING](#) 레시피로, 특정 사용자의 예상 관심 수준을 기반으로 제공하는 항목 컬렉션의 순위를 매깁니다. 개인 맞춤형-순위 레시피를 사용하면 특정 사용자에게 맞게 개인 맞춤형 항목 또는 검색 결과의 선별된 목록 순서를 개인 맞춤할 수 있습니다.

인기도-집계 레시피

고유 [사용자와 가장 많이 상호작용하는 항목을 추천하는 USER_PERSONALIZATION](#) 레시피입니다.

추천자

추천을 생성하는 도메인 데이터세트 그룹 도구입니다. 도메인 데이터세트 그룹에 대한 추천자를 만들고 애플리케이션에서 사용하여 API로 실시간 추천을 받을 수 있습니다. GetRecommendations 추천을 생성하고 나면 사용 사례를 지정하고 Personalize에서 사용 사례에 가장 적합한 구성으로 추천을 지원하는 모델을 교육합니다.

레시피

사용자가 상호작용할 항목을 예측하거나(USER_PERSONALIZED 레시피의 경우), 사용자가 관심을 보인 특정 항목과 유사한 항목을 계산하거나(RELATED_ITEMS 레시피의 경

우), 특정 사용자에게 대한 예측 관심도를 기반으로 제공하는 항목 컬렉션의 순위를 매기도록 (PERSONALIZED_RANKING 레시피의 경우) 사전 구성된 Personalize 알고리즘입니다.

솔루션

Personalize가 추천을 생성하는 데 사용하는 레시피, 사용자 지정 파라미터 및 학습된 모델(솔루션 버전).

솔루션 버전

Personalize에서 솔루션의 일부로 생성한 학습된 모델입니다. 캠페인에 솔루션 버전을 배포하여 추천을 요청하는 데 사용하는 개인 맞춤 API를 활성화합니다.

학습 모드

솔루션 버전을 만들 때 수행할 학습 범위입니다. 두 가지 모드, 즉 FULL 모드와 UPDATE 모드가 있습니다. FULL 모드는 데이터세트 그룹 내 데이터세트의 전체 학습 데이터를 기반으로 완전히 새로운 솔루션 버전을 생성합니다. UPDATE는 기존 솔루션 버전을 점진적으로 업데이트하여 마지막 학습 이후 추가한 새 항목을 추천합니다.

Note

Amazon Personalize는 사용자 맞춤화-v2, 사용자 맞춤화 또는 차세대 베스트 액션을 통해 FULL 교육 모드로 학습된 최신 솔루션 버전을 자동으로 업데이트합니다. [자동 업데이트](#) 섹션을 참조하십시오.

사용자 개인화-v2 레시피

사용자의 선호도에 따라 [사용자가 상호작용할 항목을 추천하는 USER_PERSONALIZATION](#) 레시피입니다. 이 레시피는 변환기 기반 아키텍처를 사용하여 항목 상호 작용 데이터, 항목 메타데이터 및 사용자 메타데이터에서 학습하는 모델을 학습시킵니다. 최대 5백만 개의 항목을 학습시킬 수 있으며 이전 버전보다 짧은 지연 시간으로 더 관련성 높은 추천을 생성할 수 있습니다.

사용자-개인 맞춤 레시피

사용자가 상호작용할 항목을 예측하는 계층적 순환 신경망(HRNN) 기반 [USER_PERSONALIZATION](#) 레시피입니다. 사용자-개인 맞춤 레시피는 항목 탐색 및 노출 데이터를 사용하여 새 항목에 대한 추천을 생성할 수 있습니다.

모델 배포 및 추천

다음 용어는 모델을 배포하고 사용하여 추천을 생성하는 것과 관련이 있습니다.

조치 최적화 기간

Amazon Personalize가 사용자가 취할 가능성이 가장 높은 조치를 예측할 때 사용하는 기간입니다. 예를 들어 작업 최적화 기간이 14일인 경우 Amazon Personalize는 사용자가 향후 14일 동안 취할 가능성이 가장 높은 작업을 예측합니다. 를 사용하여 솔루션을 생성할 때 작업 최적화 기간을 구성합니다. [차선책-작업 레시피](#)

배치 추론 작업

S3 버킷에서 배치 입력 데이터를 가져오고 솔루션 버전을 사용하여 추천을 생성하고 S3 버킷으로 추천을 내보내는 도구입니다. 출력 데이터에 대한 다른 위치(폴더 또는 다른 S3 버킷)를 사용하는 것이 좋습니다. 또한 배치 워크플로우를 사용하여 실시간 업데이트가 필요하지 않은 대규모 데이터 세트에서 추천을 가져올 수 있습니다.

배치 세그먼트 작업

S3 버킷에서 배치 입력 데이터를 가져오고, 솔루션 버전을 사용하여 사용자 세그먼트를 생성하고, S3 버킷으로 사용자 세그먼트를 내보내는 도구입니다. 출력 데이터에 대한 다른 위치(폴더 또는 다른 S3 버킷)를 사용하는 것이 좋습니다. 사용자_세분화 레시피로 뒷받침되는 솔루션과 함께 배치 세그먼트 작업을 사용하면 사용자가 다른 항목 또는 항목 속성이 다른 항목과 상호작용할 가능성에 따라 사용자 세그먼트를 생성할 수 있습니다.

campaign

애플리케이션 사용자를 위한 실시간 추천을 생성하기 위한 전용 트랜잭션 용량이 프로비저닝된 배포된 솔루션 버전(학습된 모델)입니다. 캠페인을 생성한 후 `getRecommendations` 또는 `getPersonalizedRankingAPI`를 사용하여 추천을 가져올 수 있습니다.

항목 탐색

탐색을 사용하면 추천에 새 항목 또는 작업, 상호 작용이 거의 없는 항목 또는 작업, 이전 행동을 기준으로 사용자와 관련성이 낮은 항목 또는 작업 등 일반적으로 사용자에게 추천될 가능성이 낮은 일부 항목 또는 작업이 포함됩니다.

지표 어트리뷰션

항목 권장 사항의 영향을 측정하는 데 사용하는 도구입니다. 지표 어트리뷰션은 가져온 항목 상호 작용 및 항목 데이터, 지정한 지표를 기반으로 보고서를 생성합니다. 예를 들어, 사용자가 본 영화의 총 길이 또는 총 클릭 이벤트 수가 여기에 해당합니다.

추천

Personalize에서 사용자가 상호작용할 것으로 예측하는 항목의 목록입니다. 사용된 Personalize 레시피에 따라 추천은 항목 목록(USER_PERSONALIZED 레시피 및 RELATED_ITEMS 레시피)이거나 제공한 항목 컬렉션의 순위(PERSONALIZED_RANKING 레시피)일 수 있습니다.

사용자 세그먼트

Personalize에서 사용자가 카탈로그와 상호작용할 것으로 예측하는 사용자 목록입니다. 사용된 사용자_세분화 레시피에 따라 항목(항목-선호도 레시피) 항목 메타데이터(항목-속성-선호도 레시피)를 기반으로 사용자 세그먼트를 생성합니다. 사용자 세그먼트는 배치 세그먼트 작업을 통해 생성합니다.

Personalize가 사용할 수 있는 데이터 유형

다음 주제에서는 Personalize로 가져올 수 있는 다양한 유형의 데이터를 소개합니다.

주제

- [상호작용 데이터](#)
- [항목 데이터](#)
- [사용자 데이터](#)
- [작업 데이터](#)
- [작업 상호 작용 데이터](#)

상호작용 데이터

상호작용은 기록한 다음 교육 데이터로 가져오는 이벤트입니다. Personalize는 주로 상호작용 데이터를 기반으로 추천을 생성합니다. 상호작용 데이터에는 다음이 포함될 수 있습니다.

- 이벤트 유형 및 이벤트 값 데이터
- 컨텍스트 메타데이터
- 노출 데이터

상호 작용 데이터를 항목 상호 작용 데이터 세트로 가져옵니다. 상호작용 데이터세트에 대한 자세한 내용은 [항목 상호 작용 데이터 세트](#) 단원을 참조하세요.

항목 데이터

Personalize에서 사용할 수 있는 항목 메타데이터에는 다음이 포함됩니다.

- 각 항목에 대한 수치 데이터(예: 가격).
- 각 항목에 대한 범주형 메타데이터(예: 항목의 장르 또는 색상).
- 각 항목의 생성 타임스탬프 데이터.
- 제품 설명 또는 영화 시놉시스와 같은 비정형 텍스트 메타데이터.

항목에 대한 메타데이터를 항목 데이터세트로 가져옵니다. 데이터세트에 대한 자세한 내용은 [항목 데이터 세트](#) 단원을 참조하세요.

사용자 데이터

Personalize에서 사용할 수 있는 사용자 메타데이터에는 다음이 포함될 수 있습니다.

- 연령과 같은 각 사용자에 대한 수치 데이터.
- 성별 또는 로열티 멤버십 상태와 같은 각 사용자에 대한 범주형 메타데이터.

사용자에 대한 메타데이터를 사용자 데이터세트로 가져옵니다. 사용자 데이터세트에 대한 자세한 내용은 [사용자 데이터 세트](#) 단원을 참조하세요.

작업 데이터

Amazon Personalize에서 사용할 수 있는 작업 데이터는 다음과 같습니다.

- 각 작업의 비즈니스 가치 또는 중요성.
- 각 작업에 대한 범주형 메타데이터(예: 계절성 또는 작업 독점성).
- Amazon Personalize에서 각 작업의 추천을 중단해야 하는 시기를 지정하는 작업 만료 타임스탬프 데이터.
- Amazon Personalize에서 사용자가 작업과 상호 작용한 후 각 작업을 추천하기 전에 기다려야 하는 시간을 지정하는 반복 빈도 데이터.

작업에 대한 데이터를 작업 데이터 세트로 가져옵니다. 도메인 데이터세트 그룹에서는 액션 및 액션 인터랙션 데이터세트를 비롯한 차선책 액션 리소스를 만들 수 없습니다. 작업 데이터 세트에 대한 자세한 내용은 [작업 데이터 세트](#) 섹션을 참조하세요.

작업 상호 작용 데이터

Amazon Personalize가 사용자와 작업의 상호 작용에서 사용할 수 있는 데이터에는 다음이 포함됩니다.

- 이벤트 유형 데이터
- 범주형 메타데이터

상호 작용 데이터를 작업 상호 작용 데이터 세트로 가져옵니다. 도메인 데이터세트 그룹에는 액션 및 액션 인터랙션 데이터세트를 비롯한 차선책 액션 리소스를 생성할 수 없습니다. 작업 상호 작용 데이터 세트에 대한 자세한 내용은 [작업 상호 작용 데이터 세트](#) 섹션을 참조하세요.

Personalize 설정

Personalize를 사용하기 전에 관리자 사용자의 Amazon Web Services(AWS) 계정이 있어야 합니다. 필요한 권한을 설정한 후에는 Amazon Personalize 콘솔, AWS CLI() 또는 SDK를 통해 Amazon AWS Command Line Interface Personalize에 액세스할 수 있습니다. AWS

주제

- [가입하여 다음을 수행하십시오. AWS 계정](#)
- [관리자 액세스 권한이 있는 사용자 생성](#)
- [리전 및 엔드포인트](#)
- [권한 설정](#)
- [설정: AWS CLI](#)
- [AWS SDK 설정](#)

가입하여 다음을 수행하십시오. AWS 계정

계정이 없는 경우 다음 단계를 완료하여 계정을 만드세요. AWS 계정

가입하려면 AWS 계정

1. <https://portal.aws.amazon.com/billing/signup>을 여세요.
2. 온라인 지시 사항을 따르세요.

등록 절차 중에는 전화를 받고 키패드로 인증 코드를 입력하는 과정이 있습니다.

에 AWS 계정가입하면 AWS 계정 루트 사용자a가 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스 액세스 권한이 있습니다. 보안 모범 사례는 사용자에게 관리 액세스 권한을 할당하고, 루트 사용자만 사용하여 [루트 사용자 액세스 권한이 필요한 작업](#)을 수행하는 것입니다.

AWS 가입 절차가 완료된 후 확인 이메일을 보냅니다. 언제든지 <https://aws.amazon.com/>으로 가서 내 계정(My Account)을 선택하여 현재 계정 활동을 보고 계정을 관리할 수 있습니다.

관리자 액세스 권한이 있는 사용자 생성

등록한 AWS 계정후에는 일상적인 작업에 루트 사용자를 사용하지 않도록 관리 사용자를 보호하고 AWS IAM Identity Center활성화하고 생성하십시오 AWS 계정 루트 사용자.

보안을 유지하세요. AWS 계정 루트 사용자

1. 루트 사용자를 선택하고 AWS 계정 이메일 주소를 입력하여 계정 [AWS Management Console](#)소유자로 로그인합니다. 다음 페이지에서 비밀번호를 입력합니다.

루트 사용자를 사용하여 로그인하는 데 도움이 필요하면AWS 로그인 사용 설명서의 [루트 사용자 로 로그인](#)을 참조하세요.

2. 루트 사용자의 다중 인증(MFA)을 활성화합니다.

지침은 IAM [사용 설명서의 AWS 계정 루트 사용자 \(콘솔\)에 대한 가상 MFA 디바이스 활성화를 참조](#)하십시오.

관리자 액세스 권한이 있는 사용자 생성

1. IAM Identity Center를 활성화합니다.

지침은 AWS IAM Identity Center 사용 설명서의 [AWS IAM Identity Center설정](#)을 참조하세요.

2. IAM Identity Center에서 사용자에게 관리자 액세스 권한을 부여합니다.

를 ID 소스로 사용하는 방법에 대한 자습서는 사용 [설명서의 기본값으로 IAM Identity Center 디렉터리사용자 액세스 구성](#)을 참조하십시오. IAM Identity Center 디렉터리 AWS IAM Identity Center

관리 액세스 권한이 있는 사용자로 로그인

- IAM IDentity Center 사용자로 로그인하려면 IAM IDentity Center 사용자를 생성할 때 이메일 주소로 전송된 로그인 URL을 사용합니다.

IAM Identity Center 사용자를 사용하여 [로그인하는 데 도움이 필요하면 사용 설명서의 AWS 액세스 포털 로그인](#)을 참조하십시오.AWS 로그인

추가 사용자에게 액세스 권한 할당

1. IAM Identity Center에서 최소 권한 적용 모범 사례를 따르는 권한 세트를 생성합니다.

지침은 AWS IAM Identity Center 사용 설명서의 [Create a permission set](#)를 참조하세요.

2. 사용자를 그룹에 할당하고, 그룹에 Single Sign-On 액세스 권한을 할당합니다.

지침은 AWS IAM Identity Center 사용 설명서의 [Add groups](#)를 참조하세요.

리전 및 엔드포인트

엔드포인트는 웹 서비스의 진입점인 URL입니다. 각 엔드포인트는 특정 AWS 리전과 연결되어 있습니다. 특정 캠페인의 모든 Amazon Personalize 구성 요소 (데이터 세트, 솔루션 AWS CLI, 캠페인, 이벤트 트래커)를 동일한 지역에 생성해야 하므로 Amazon Personalize 콘솔, Amazon Personalize SDK의 기본 지역에 주의를 기울이십시오. Personalize에서 지원하는 리전 및 엔드포인트는 [리전 및 엔드포인트](#) 단원을 참조하세요.

권한 설정

사용자, 그룹 또는 역할에 Personalize 리소스와 상호작용할 수 있는 권한을 부여해야만 합니다. 또한 Personalize에서 생성한 리소스에 액세스하고 사용자를 대신하여 작업을 수행할 수 있는 권한을 Personalize에 부여해야 합니다.

권한을 설정하려면

1. 사용자, 그룹 또는 역할에 Personalize 리소스와 상호작용하고 Personalize에 역할을 전달할 수 있는 권한을 부여합니다. [사용자에게 Amazon Personalize에 액세스할 수 있는 권한 부여](#) 단원을 참조하세요.
2. Personalize에 Personalize의 리소스에 액세스할 수 있는 권한과 사용자를 대신하여 작업을 수행할 수 있는 권한을 부여합니다. [Personalize에 리소스에 액세스할 수 있는 권한 부여](#)를 참조하세요.
3. Personalize 서비스 역할의 신뢰 정책을 수정하여 [혼동되는 대리인](#) 문제를 방지합니다. 신뢰 관계 정책 예제는 [교차 서비스 혼동된 대리인 방지](#) 단원을 참조하세요. 역할의 신뢰 정책을 수정하는 방법에 대한 자세한 내용은 [역할 수정](#) 단원을 참조하세요.
4. 암호화에 AWS Key Management Service (AWS KMS)를 사용하는 경우 Amazon Personalize와 Amazon Personalize IAM 서비스 역할에 키를 사용할 권한을 부여해야 합니다. 자세한 정보는 [Amazon Personalize에 AWS KMS 키 사용 권한 부여](#)를 참조하세요.
5. IAM 및 S3 버킷 정책을 사용하여 S3 리소스에 대한 액세스 권한을 Personalize에 부여하려면 [Personalize에 S3 리소스에 대한 액세스 권한 부여](#)의 단계를 완료합니다.

주제

- [사용자에게 Amazon Personalize에 액세스할 수 있는 권한 부여](#)
- [Personalize에 리소스에 액세스할 수 있는 권한 부여](#)
- [Personalize에 S3 리소스에 대한 액세스 권한 부여](#)
- [Amazon Personalize에 AWS KMS 키 사용 권한 부여](#)

사용자에게 Amazon Personalize에 액세스할 수 있는 권한 부여

사용자에게 Amazon Personalize에 대한 액세스 권한을 제공하려면 Amazon Personalize 리소스에 액세스할 권한을 부여하고 역할을 Amazon Personalize에 전달하는 IAM 정책을 생성합니다. 그런 다음 사용자, 그룹 또는 역할에 권한을 추가할 때 해당 정책을 사용합니다.

사용자를 위한 새 IAM 정책 생성

Amazon Personalize 리소스에 대한 전체 액세스 권한을 Amazon Personalize에 제공하는 IAM 정책을 생성합니다.

JSON 정책 편집기를 사용하여 정책을 생성하려면

1. AWS Management Console에 로그인하여 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 왼쪽의 탐색 창에서 정책을 선택합니다.

정책을 처음으로 선택하는 경우 관리형 정책 소개 페이지가 나타납니다. 시작하기를 선택합니다.

3. 페이지 상단에서 정책 생성을 선택합니다.
4. 정책 편집기 섹션에서 JSON 옵션을 선택합니다.
5. 다음 JSON 정책 문서를 입력합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "personalize:*"
      ],
      "Resource": "*"
    }
  ],
}
```

```

    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "personalize.amazonaws.com"
        }
      }
    }
  ]
}

```

6. 다음을 선택합니다.

Note

언제든지 시각적 편집기 옵션과 JSON 편집기 옵션 간에 전환할 수 있습니다. 그러나 변경을 적용하거나 시각적 편집기에서 다음을 선택한 경우 IAM은 시각적 편집기에 최적화되도록 정책을 재구성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [정책 재구성](#)을 참조하세요.

7. 검토 및 생성 페이지에서 생성하는 정책에 대한 정책 이름과 설명(선택 사항)을 입력합니다. 이 정책에 정의된 권한을 검토하여 정책이 부여한 권한을 확인합니다.
8. 정책 생성을 선택하고 새로운 정책을 저장합니다.

Amazon Personalize에서 작업을 수행하는 데 필요한 권한만 부여하기 위해 사용자에게 필요한 작업만 포함하도록 이전 정책을 수정합니다. Amazon Personalize 작업의 전체 목록은 [Amazon Personalize에 사용되는 작업, 리소스 및 조건 키](#)를 참조하세요.

Amazon Personalize에 대한 액세스 권한 제공

사용자에게 권한을 제공할 때 새 IAM 정책을 연결합니다.

액세스 권한을 제공하려면 사용자, 그룹 또는 역할에 권한을 추가합니다.

- AWS IAM Identity Center의 사용자 및 그룹:

권한 세트를 생성합니다. AWS IAM Identity Center 사용 설명서의 [권한 세트 생성](#)의 지침을 따르세요.

- 자격 증명 공급자를 통해 IAM에서 관리되는 사용자:

아이덴티티 페더레이션을 위한 역할을 생성합니다. IAM 사용 설명서의 [서드 파티 자격 증명 공급자의 역할 만들기\(연합\)](#)의 지침을 따르세요.

- IAM 사용자:

- 사용자가 맡을 수 있는 역할을 생성합니다. IAM 사용 설명서에서 [IAM 사용자의 역할 생성](#)의 지침을 따르세요.
- (권장되지 않음) 정책을 사용자에게 직접 연결하거나 사용자를 사용자 그룹에 추가합니다. IAM 사용 설명서에서 [사용자\(콘솔\)에 권한 추가](#)의 지침을 따르세요.

Personalize에 리소스에 액세스할 수 있는 권한 부여

Personalize에 리소스에 대한 액세스 권한을 부여하려면 Personalize에 Personalize 리소스에 대한 전체 액세스 권한을 부여하는 IAM 정책을 생성해야 합니다. 또는 관리형 정책을 사용할 수도 있습니다. `AWS AmazonPersonalizeFullAccess AmazonPersonalizeFullAccess`필요한 것보다 더 많은 권한을 제공합니다. 필요한 권한만 부여하는 새 IAM 정책을 생성하는 것이 좋습니다. 관리형 정책에 대한 자세한 정보는 [AWS 관리형 정책](#) 단원을 참조하세요.

정책을 작성한 후 Personalize에 대한 IAM 역할을 만들어 여기에 정책을 연결합니다.

주제

- [Personalize에 대한 새 IAM 정책 생성](#)
- [Personalize에 대한 IAM 역할 생성](#)

Personalize에 대한 새 IAM 정책 생성

Personalize 리소스에 대한 전체 액세스 권한을 Personalize에 제공하는 IAM 정책을 생성합니다.

JSON 정책 편집기를 사용하여 정책을 생성하려면

1. 에 AWS Management Console 로그인하고 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 왼쪽의 탐색 창에서 정책을 선택합니다.

정책을 처음으로 선택하는 경우 관리형 정책 소개 페이지가 나타납니다. 시작하기를 선택합니다.

3. 페이지 상단에서 정책 생성을 선택합니다.
4. 정책 편집기 섹션에서 JSON 옵션을 선택합니다.
5. 다음 JSON 정책 문서를 입력합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "personalize:*"
      ],
      "Resource": "*"
    }
  ]
}
```

6. 다음을 선택합니다.

Note

언제든지 시각적 편집기 옵션과 JSON 편집기 옵션 간에 전환할 수 있습니다. 그러나 변경을 적용하거나 시각적 편집기에서 다음을 선택한 경우 IAM은 시각적 편집기에 최적화되도록 정책을 재구성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [정책 재구성](#)을 참조하십시오.

7. 검토 및 생성 페이지에서 생성하는 정책에 대한 정책 이름과 설명(선택 사항)을 입력합니다. 이 정책에 정의된 권한을 검토하여 정책이 부여한 권한을 확인합니다.
8. 정책 생성을 선택하고 새로운 정책을 저장합니다.

Personalize에 대한 IAM 역할 생성

Amazon Personalize를 사용하려면 Amazon Personalize에 대한 AWS Identity and Access Management 서비스 역할을 생성해야 합니다. 서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하는 것으로 가정하는 [IAM 역할](#)입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 [AWS 서비스에 대한 권한을 위임할 역할 생성](#)을 참

조하세요. Personalize에 대한 서비스 역할을 생성한 후 필요에 따라 [추가 서비스 역할 권한](#)에 나열된 추가 권한을 부여합니다.

Personalize에 대한 서비스 역할을 생성하려면(IAM 콘솔)

1. AWS Management Console [로그인](#)하고 <https://console.aws.amazon.com/iam/> 에서 IAM 콘솔을 [열](#)니다.
2. IAM 콘솔의 탐색 창에서 역할을 선택하고 역할 생성을 선택합니다.
3. 신뢰할 수 있는 엔터티 유형에 AWS 서비스를 선택합니다.
4. 서비스 또는 사용 사례의 경우 Amazon Personalize를 선택한 다음 사용 사례 개인화를 선택합니다.
5. 다음을 선택합니다.
6. 이전 절차에서 생성한 스키마를 선택합니다.
7. (선택 사항) [권한 경계](#)를 선택합니다. 이는 서비스 역할에서 가능한 고급 기능이며 서비스 링크된 역할은 아닙니다.
 - a. 권한 경계 설정 섹션을 열고 최대 역할 권한을 관리하기 위한 권한 경계 사용을 선택합니다.

IAM에는 계정의 AWS 관리형 및 고객 관리형 정책 목록이 포함되어 있습니다.
 - b. 정책을 선택하여 권한 경계를 사용하세요.
8. 다음을 선택합니다.
9. 역할의 목적을 식별하는 데 도움이 되는 역할 이름이나 역할 이름 접미사를 입력합니다.

Important

역할 이름을 지정할 때는 다음 사항에 유의하세요.

- 역할 이름은 사용자 내에서 고유해야 AWS 계정하며 대소문자를 구분하여 고유할 수 없습니다.

예를 들어, 이름이 **PRODRole**과 **prodrole**, 두 가지로 지정된 역할을 만들지 마십시오. 역할 이름이 정책 또는 ARN의 일부로 사용되는 경우 역할 이름은 대소문자를 구분합니다. 그러나 로그인 프로세스와 같이 콘솔에서 역할 이름이 고객에게 표시되는 경우에는 역할 이름이 대소문자를 구분하지 않습니다.

- 다른 엔터티가 역할을 참조할 수 있기 때문에 역할이 생성된 후에는 역할 이름을 편집할 수 없습니다.

10. (선택 사항) 설명에 역할에 대한 설명을 입력합니다.
11. (선택 사항) 역할에 대한 사용 사례와 권한을 편집하려면 1단계: 신뢰할 수 있는 엔터티 선택 또는 2단계: 권한 추가 섹션에서 편집을 선택합니다.
12. (선택 사항) 태그를 키-값 페어로 연결하여 역할을 식별, 구성 또는 검색합니다. IAM에서 태그 사용에 대한 자세한 내용을 알아보려면 IAM 사용 설명서의 [IAM 리소스에 태그 지정](#)을 참조하세요.
13. 역할을 검토한 다음 역할 생성을 선택합니다.

Amazon Personalize에 대한 역할을 생성하고 나면 [Amazon S3 버킷과 모든 AWS KMS 키에 대한 액세스 권한을 부여](#)할 준비가 된 것입니다.

추가 서비스 역할 권한

역할을 만들고 Personalize에 리소스에 액세스할 수 있는 권한을 부여한 후 다음과 같이 합니다.

1. Personalize 서비스 역할의 신뢰 정책을 수정하여 [혼동되는 대리인](#) 문제를 방지합니다. 신뢰 관계 정책 예제는 [교차 서비스 혼동된 대리인 방지](#) 단원을 참조하세요. 역할의 신뢰 정책을 수정하는 방법에 대한 자세한 내용은 [역할 수정](#) 단원을 참조하세요.
2. 암호화에 AWS Key Management Service (AWS KMS) 를 사용하는 경우 Amazon Personalize 와 Amazon Personalize IAM 서비스 역할에 키를 사용할 권한을 부여해야 합니다. 자세한 정보는 [Amazon Personalize에 AWS KMS 키 사용 권한 부여](#)을 참조하세요.

Personalize에 S3 리소스에 대한 액세스 권한 부여

Personalize에 S3 버킷에 대한 액세스 권한을 부여하려면 다음과 같이 합니다.

1. 아직 권한을 설정하지 않았다면 Personalize가 사용자를 대신하여 Personalize의 리소스에 액세스할 수 있도록 하는 [권한 설정](#)의 단계를 따라 합니다.
2. S3 버킷에 대한 액세스를 허용하는 Personalize 서비스 역할([Personalize에 대한 IAM 역할 생성](#) 참조)에 정책을 연결합니다. 자세한 내용은 [S3 정책을 Personalize 서비스 역할에 연결](#) 단원을 참조하세요.
3. Personalize가 데이터에 액세스할 수 있도록 데이터 파일이 들어 있는 S3 버킷에 버킷 정책을 연결합니다. 자세한 정보는 [S3 버킷에 Personalize 액세스 정책 연결](#)을 참조하세요.
4. 암호화에 AWS Key Management Service (AWS KMS) 를 사용하는 경우 Amazon Personalize 와 Amazon Personalize IAM 서비스 역할에 키를 사용할 권한을 부여해야 합니다. 자세한 정보는 [Amazon Personalize에 AWS KMS 키 사용 권한 부여](#)을 참조하세요.

Note

Amazon Personalize는 AWS VPC와 통신하지 않으므로, Amazon Personalize는 VPC 액세스만 허용하는 Amazon S3 버킷과 상호 작용할 수 없습니다.

주제

- [S3 정책을 Personalize 서비스 역할에 연결](#)
- [S3 버킷에 Personalize 액세스 정책 연결](#)

S3 정책을 Personalize 서비스 역할에 연결

S3 정책을 Personalize 역할에 연결하려면 다음과 같이 합니다.

1. [IAM 콘솔 \(https://console.aws.amazon.com/iam/\)](https://console.aws.amazon.com/iam/) 에 로그인합니다.
2. 탐색 창에서 정책을 선택한 다음 정책 생성을 선택합니다.
3. JSON 탭을 선택하고 다음과 같이 정책을 업데이트합니다. bucket-name을 사용 중인 버킷의 이름으로 바꿉니다. 데이터세트 가져오기 작업 또는 데이터 삭제 작업에 다음 정책을 사용할 수 있습니다. 배치 워크플로를 사용하거나 데이터세트 내보내기 작업을 생성하는 경우 Amazon Personalize에는 추가 권한이 필요합니다. [배치 워크플로우에 대한 서비스 역할 정책](#) 또는 [데이터 세트를 내보내는 데 사용되는 S3 버킷 정책](#)을 참조하세요.

```
{
  "Version": "2012-10-17",
  "Id": "PersonalizeS3BucketAccessPolicy",
  "Statement": [
    {
      "Sid": "PersonalizeS3BucketAccessPolicy",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}
```

}

4. 다음: 태그를 선택합니다. 원하는 경우 태그를 추가하고 검토를 선택합니다.
5. 정책에 이름을 지정합니다.
6. (선택 사항) 설명에 이 정책을 설명하는 짧은 문장을 입력합니다(예: **Allow Amazon Personalize to access its Amazon S3 bucket.**)
7. 정책 생성(Create policy)을 선택합니다.
8. 탐색 창에서 역할을 선택하고 Personalize에 대해 생성한 역할을 선택합니다. [Personalize에 대한 IAM 역할 생성](#)를 참조하세요.
9. 권한 탭에서 정책 연결을 선택합니다.
10. 목록에 정책을 표시하려면 필터 정책 필터 상자에 정책 이름의 일부를 입력합니다.
11. 이 절차의 앞 부분에서 정책 옆에 있는 확인란을 선택합니다.
12. 정책 연결을 선택합니다.

Personalize에서 역할을 사용할 준비가 되기 전에 데이터가 들어 있는 S3 버킷에 버킷 정책도 연결해야 합니다. [S3 버킷에 Personalize 액세스 정책 연결](#)를 참조하세요.

배치 워크플로우에 대한 서비스 역할 정책

배치 워크플로우를 완료하려면 Personalize에 S3 버킷에 액세스하고 파일을 추가할 수 있는 권한이 필요합니다. Personalize 역할에 다음 정책을 연결하려면 위 단계를 따라 합니다. bucket-name을 사용 중인 버킷의 이름으로 바꿉니다. 배치 워크플로우에 대한 자세한 내용은 [배치 권장 사항 및 사용자 세그먼트\(사용자 지정 리소스\)](#) 단원을 참조하세요.

```
{
  "Version": "2012-10-17",
  "Id": "PersonalizeS3BucketAccessPolicy",
  "Statement": [
    {
      "Sid": "PersonalizeS3BucketAccessPolicy",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",

```



```

        "arn:aws:s3:::bucket-name/*"
    ]
}
]
}

```

데이터세트 내보내기를 위한 서비스 역할 정책

데이터세트를 내보내려면 Personalize 서비스 역할에 S3 버킷에 PutObject 및 ListBucket 작업을 사용할 권한이 있어야 합니다. 다음 예제 정책은 Personalize에 PutObject 및 ListBucket 권한을 부여합니다. *bucket-name*을 사용 중인 버킷의 이름으로 바꾸고 정책을 Personalize의 서비스 역할에 연결합니다. 정책을 서비스 역할에 연결하는 방법에 대한 자세한 내용은 [S3 정책을 Personalize 서비스 역할에 연결](#) 단원을 참조하세요.

```

{
  "Version": "2012-10-17",
  "Id": "PersonalizeS3BucketAccessPolicy",
  "Statement": [
    {
      "Sid": "PersonalizeS3BucketAccessPolicy",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}

```

S3 버킷에 Personalize 액세스 정책 연결

Personalize에 S3 버킷에 액세스할 수 있는 권한이 필요합니다. 데이터세트 가져오기 작업 또는 데이터 삭제 작업에 다음 정책을 사용할 수 있습니다. *bucket-name*을 사용 중인 버킷의 이름으로 바꿉니다. 배치 워크플로우에 대한 자세한 내용은 [배치 워크플로우에 대한 S3 버킷 정책](#) 단원을 참조하세요.

S3 버킷 정책에 대한 자세한 내용은 [S3 버킷 정책을 추가하려면 어떻게 해야 하나요?](#) 단원을 참조하세요.

```
{
  "Version": "2012-10-17",
  "Id": "PersonalizeS3BucketAccessPolicy",
  "Statement": [
    {
      "Sid": "PersonalizeS3BucketAccessPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "personalize.amazonaws.com"
      },
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}
```

배치 워크플로우에 대한 S3 버킷 정책

배치 워크플로우의 경우 Personalize는 S3 버킷에 액세스하고 S3 버킷에 파일을 추가할 권한이 필요합니다. 버킷에 다음 정책을 연결합니다. bucket-name을 사용 중인 버킷의 이름으로 바꿉니다.

버킷에 S3 버킷 정책을 추가하는 방법에 대한 자세한 내용은 [S3 버킷 정책을 추가하려면 어떻게 해야 합니까?](#) 단원을 참조하세요. 배치 워크플로우에 대한 자세한 내용은 [배치 권장 사항 및 사용자 세그먼트\(사용자 지정 리소스\)](#) 단원을 참조하세요.

```
{
  "Version": "2012-10-17",
  "Id": "PersonalizeS3BucketAccessPolicy",
  "Statement": [
    {
      "Sid": "PersonalizeS3BucketAccessPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "personalize.amazonaws.com"
      },
      "Action": [
```

```

        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
    ],
    "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
    ]
}
]
}

```

데이터세트를 내보내는 데 사용되는 S3 버킷 정책

데이터세트를 내보내려면 Personalize에 S3 버킷에 PutObject 및 ListBucket 작업을 사용할 권한이 있어야 합니다. 다음 예제 정책은 Personalize 원칙에 PutObject 및 ListBucket 권한을 부여합니다. *bucket-name*을 사용 중인 버킷의 이름으로 바꾸고 정책을 버킷에 연결합니다. S3 버킷 정책을 버킷에 추가하는 방법에 대한 자세한 내용은 Simple Storage Service 사용 설명서의 [S3 버킷 정책을 추가하려면 어떻게 해야 하나요?](#) 단원을 참조하세요.

```

{
  "Version": "2012-10-17",
  "Id": "PersonalizeS3BucketAccessPolicy",
  "Statement": [
    {
      "Sid": "PersonalizeS3BucketAccessPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "personalize.amazonaws.com"
      },
      "Action": [
        "s3:PutObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}

```

Amazon Personalize에 AWS KMS 키 사용 권한 부여

Amazon Personalize 콘솔 또는 API를 사용할 때 AWS Key Management Service(AWS KMS) 키를 지정하거나 AWS KMS 키를 사용하여 Amazon S3 버킷을 암호화하는 경우 Amazon Personalize에 키 사용 권한을 부여해야 합니다. 권한을 부여하려면 서비스 역할에 연결된 AWS KMS 키 정책 및 IAM 정책이 Amazon Personalize에 키 사용 권한을 부여해야 합니다. 이는 Amazon Personalize에서 다음을 생성하는 경우에 적용됩니다.

- 데이터 세트 그룹
- 데이터 세트 가져오기 작업 (AWS KMS 키 정책만 권한을 부여해야 함)
- 데이터 세트 내보내기 작업
- 배치 추론 작업
- 배치 세그먼트 작업
- 지표 어트리뷰션

AWS KMS 키 정책 및 IAM 정책이 다음 작업을 수행할 권한을 부여해야 합니다.

- Decrypt
- GenerateDataKey
- DescribeKey
- CreateGrant(키 정책에만 필요)
- ListGrants

리소스를 생성한 후 AWS KMS 키 권한을 취소하면 필터를 만들거나 추천을 받을 때 문제가 발생할 수 있습니다. AWS KMS 키 정책에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 [AWS KMS에서 키 정책 사용](#) 단원을 참조하세요. IAM 정책 생성에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 정책 생성](#) 단원을 참조하세요. 역할에 IAM 정책을 연결하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 자격 증명 권한 추가 및 제거](#) 단원을 참조하세요.

주제

- [예제 키 정책](#)
- [예제 IAM 정책](#)

예제 키 정책

다음 예제 키 정책에서는 Amazon Personalize와 사용자 역할에 이전 Amazon Personalize 작업에 대한 최소한의 권한을 부여합니다. 데이터 세트 그룹을 생성할 때 키를 지정하고 데이터 세트에서 데이터를 내보내려는 경우 키 정책에 GenerateDataKeyWithoutPlaintext 작업이 포함되어야 합니다.

```
{
  "Version": "2012-10-17",
  "Id": "key-policy-123",
  "Statement": [
    {
      "Sid": "Allow use of the key",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<account-id>:role/<personalize-role-name>",
        "Service": "personalize.amazonaws.com"
      },
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "kms:DescribeKey",
        "kms:CreateGrant",
        "kms:ListGrants"
      ],
      "Resource": "*"
    }
  ]
}
```

예제 IAM 정책

다음 예제 IAM 정책은 이전 Amazon Personalize 작업에 필요한 최소한의 AWS KMS 권한을 역할에 부여합니다. 데이터 세트 가져오기 작업의 경우 AWS KMS 키 정책만 권한을 부여하면 됩니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey",

```

```

        "kms:DescribeKey",
        "kms:ListGrants"
    ],
    "Resource": "*"
}
]
}

```

설정: AWS CLI

AWS Command Line Interface (AWS CLI) 는 Amazon Personalize를 비롯한 AWS 서비스를 관리하기 위한 통합 개발자 도구입니다. 이 도구를 설치하는 것이 좋습니다.

1. 를 설치하려면 AWS Command Line Interface 인터페이스 사용 설명서의 [설치](#) 지침을 따르십시오. AWS CLI AWS Command Line Interface
2. 를 호출하도록 프로필을 AWS CLI 구성하고 설정하려면 사용 [AWS Command Line Interface 설명서의 AWS CLI 구성의](#) 지침을 따르십시오. AWS CLI
3. AWS CLI 프로필이 제대로 구성되었는지 확인하려면 다음 명령을 실행합니다.

```
aws configure --profile default
```

프로파일이 올바르게 구성된 경우 다음과 유사한 출력이 표시됩니다.

```

AWS Access Key ID [*****52FQ]:
AWS Secret Access Key [*****xgyZ]:
Default region name [us-west-2]:
Default output format [json]:

```

4. Amazon Personalize와 함께 사용하도록 구성되어 AWS CLI 있는지 확인하려면 다음 명령을 실행합니다.

```
aws personalize help
```

그리고

```
aws personalize-runtime help
```

그리고

```
aws personalize-events help
```

AWS CLI 가 올바르게 구성된 경우 Amazon Personalize, Amazon Personalize 런타임 및 Amazon Personalize 이벤트에 지원되는 AWS CLI 명령 목록이 표시됩니다.

를 설정했는데 Amazon Personalize 명령을 인식하지 못하는 경우 를 업데이트하십시오. AWS CLI AWS CLI를 업데이트하려면 다음 명령을 실행합니다. AWS CLI

```
pip3 install awscli --upgrade --user
```

자세한 내용은 [필을 사용하여 AWS CLI 설치](#) 단원을 참조하세요.

AWS SDK 설정

사용하려는 AWS SDK를 다운로드하고 설치합니다. 이 가이드에서는 파이썬용 SDK (Boto3), 자바 2.x용 SDK, v3용 SDK의 예를 제공합니다. JavaScript 다른 AWS SDK에 대한 자세한 내용은 [Amazon Web Services용 도구를](#) 참조하십시오. Amplify 설정에 대한 자세한 내용은 [AWS Amplify](#)를 참조하세요.

- [AWS SDK for Python \(Boto3\)](#)

Python용 SDK(Boto3)를 설치하려면 Boto3 설명서의 [퀵스타트](#) 지침을 따릅니다.

- [Java 2.x용 SDK](#)

Java 2.x용 SDK 설정에 대한 자세한 내용은 AWS SDK for Java 2.x 개발자 안내서의 [Java 2.x용 SDK 시작하기](#) 항목을 참조하세요.

Personalize의 코드 예제는 [AWS SDK 예제](#) 리포지토리의 [Personalize 자바 코드 샘플](#)을 참조하세요.

- [AWS SDK for JavaScript v3](#)

JavaScript v3용 SDK 설정에 대해 알아보려면 개발자 [안내서의 시작하기 AWS SDK for JavaScript](#) 항목을 참조하십시오. AWS SDK for JavaScript

[Amazon Personalize의 코드 예제는 SDK 예제 리포지토리의 v3용 JavaScript SDK에 대한 Amazon Personalize 코드 예제를 참조하십시오.](#) AWS

시작하기

다음 섹션은 Amazon Personalize 콘솔 AWS CLI 및 SDK를 사용하여 Amazon Personalize를 시작하는데 도움이 됩니다. AWS 이 자습서에서는 600명의 사용자가 제공한 9,700개 영화에 대한 100,000개의 영화 평점으로 구성된 과거 데이터를 사용합니다.

자습서를 단순화하기 위해:

- 작은 데이터셋을 사용합니다. 이는 리소스에서 생성되는 모든 지표에 부정적인 영향을 미칠 수 있습니다. 이 자습서는 Personalize 워크플로우를 소개하는 역할을 할 뿐 반드시 최고 성능의 모델을 생성하지는 않습니다.
- Amazon은 단지 항목 상호 작용 데이터 세트를 만들 뿐이며, 사용자가 영화를 봤다는 사실을 신뢰할 뿐 영화를 어떻게 평가했는지는 신뢰하지 않습니다. 그러면 학습 데이터의 준비가 간소화됩니다.
- 실시간 사용자 상호작용 이벤트는 기록하지 않습니다. 사용자 이벤트 캡처에 대한 자세한 정보는 [이벤트 기록](#) 단원을 참조하세요.

도메인 데이터셋 그룹에서 시작하거나 사용자 지정 데이터셋 그룹에서 시작하도록 선택할 수 있습니다.

- 도메인 데이터셋 그룹은 도메인을 기반으로 다양한 사용 사례에 최적화된 리소스를 제공합니다. 도메인 데이터셋 그룹 생성을 시작하려면 [시작하기 전제 조건](#)을 완료한 다음 [도메인 데이터 세트 그룹 시작하기](#)에서 자습서를 완료합니다.
- 사용자 지정 데이터셋 그룹을 사용하면 사용자 지정 리소스만 생성하고 구성할 수 있습니다. 사용자 지정 리소스 및 [사용자 맞춤화-v2 레시피](#) 레시피로 사용자에게 개인화된 영화 추천을 제공하기 시작하려면 을 완료한 다음 에서 [시작하기 전제 조건](#) 자습서를 시작하십시오. [사용자 지정 데이터셋 그룹 시작하기](#)

시작하기 연습을 마쳤을 때 불필요한 요금이 발생하지 않도록 하려면 [리소스 정리](#)의 단계에 따라 생성한 리소스를 삭제합니다.

주제

- [시작하기 전제 조건](#)
- [도메인 데이터 세트 그룹 시작하기](#)
- [사용자 지정 데이터셋 그룹 시작하기](#)
- [리소스 정리](#)

시작하기 전제 조건

다음 단계는 시작하기 연습의 전제 조건을 나타냅니다.

1. Personalize가 사용자를 대신하여 리소스에 액세스할 수 있도록 권한을 설정합니다. 여기에는 Personalize에 대한 서비스 역할을 생성하고 IAM 정책을 통해 Personalize 리소스에 대한 액세스 권한을 부여하는 작업이 포함됩니다. 자세한 내용은 [Personalize에 리소스에 액세스할 수 있는 권한 부여](#) 단원을 참조하세요.
2. 학습 데이터를 준비해 데이터를 S3 버킷에 업로드합니다.
 - 도메인 데이터세트 그룹 자습서는 [학습 데이터 생성\(도메인 데이터세트 그룹\)](#) 단원을 참조하세요.
 - 사용자 지정 데이터세트 그룹 자습서는 [학습 데이터 생성\(사용자 지정 데이터세트 그룹\)](#) 단원을 참조하세요.
3. [Personalize에 S3 리소스에 대한 액세스 권한 부여](#)에 지정된 대로 Personalize 서비스 역할에 S3 리소스에 액세스할 수 있는 권한을 부여합니다.

학습 데이터 생성(도메인 데이터세트 그룹)

학습 데이터를 생성하려면 영화 평점 데이터를 다운로드하여 수정하고 Simple Storage Service(S3) 버킷에 저장합니다. 그런 다음, 버킷에서 읽을 수 있는 권한을 Personalize에 부여합니다.

학습 데이터를 생성하려면

1. 교육 및 개발 권장 목록에서 영화 등급 zip 파일인 [ml-latest-small.zip](#) 를 다운로드하고 압축을 풉니다 (F. 맥스웰 하퍼, 조셉 A. 콘스탄, 2015. [MovieLens](#) MovieLens 데이터세트: 역사와 맥락. 인터랙티브 인텔리전트 시스템에서의 ACM 트랜잭션 (TII) 5, 4:19:1 — 19:19. <https://doi.org/10.1145/2827872>).
2. ratings.csv 파일을 엽니다. 이 파일에는 이 자습서의 상호작용 데이터가 들어 있습니다.
 - a. 평점 열을 삭제합니다.
 - b. userId 및 movieId 열의 이름을 각각 USER_ID 및 ITEM_ID로 바꿉니다.
 - c. EVENT_TYPE 열을 추가하고 모든 레코드의 값을 watch로 설정합니다. Microsoft Excel을 사용하는 경우 열의 첫 번째 셀에 watch를 입력한 다음 셀의 오른쪽 하단 모서리를 두 번 클릭하여 모든 레코드에 대해 EVENT_TYPE을 설정할 수 있습니다. 헤더가 다음과 같아야 합니다.

USER_ID, ITEM_ID, TIMESTAMP, EVENT_TYPE

Personalize가 데이터를 인식하려면 이 열이 표시된 것과 정확히 일치해야 합니다. 데이터의 처음 몇 줄은 다음과 같은 형식이어야 합니다.

```
USER_ID, ITEM_ID, TIMESTAMP, EVENT_TYPE
1, 1, 964982703, watch
1, 3, 964981247, watch
1, 6, 964982224, watch
1, 47, 964983815, watch
1, 50, 964982931, watch
....
....
```

ratings.csv 파일을 저장합니다.

3. S3 버킷에 ratings.csv를 업로드합니다. 자세한 내용은 Simple Storage Service 사용 설명서의 [드래그 앤 드롭을 사용하여 파일 및 폴더 업로드](#) 단원을 참조하세요.
4. Personalize에 버킷의 데이터를 읽을 수 있는 권한을 부여합니다. 자세한 내용은 [Personalize에 S3 리소스에 대한 액세스 권한 부여](#) 단원을 참조하세요.

학습 데이터 생성(사용자 지정 데이터세트 그룹)

학습 데이터를 생성하려면 영화 평점 데이터를 다운로드하여 수정하고 Simple Storage Service(S3) 버킷에 저장합니다. 그런 다음, 버킷에서 읽을 수 있는 권한을 Personalize에 부여합니다.

1. 영화 등급 압축 파일인 [ml-latest-small.zip](#) 파일을 다운로드하고 압축을 풉니다 (F. [MovieLens](#) 맥스 웰 하퍼, 조셉 A. 콘스탄, 2015. MovieLens 데이터세트: 역사와 맥락. 인터랙티브 인텔리전트 시스템에서의 ACM 트랜잭션 (TII) 5, 4:19:1 — 19:19. <https://doi.org/10.1145/2827872>).
2. ratings.csv 파일을 엽니다. 이 파일에는 이 자습서의 상호작용 데이터가 들어 있습니다.
 - a. 평점 열을 삭제합니다.
 - b. 헤더 행을 다음으로 바꿉니다.

USER_ID, ITEM_ID, TIMESTAMP

Personalize가 데이터를 인식하려면 이 헤더가 표시된 것과 정확히 일치해야 합니다.

ratings.csv 파일을 저장합니다.

3. S3 버킷에 ratings.csv를 업로드합니다. 자세한 내용은 Simple Storage Service 사용 설명서의 [드래그 앤 드롭을 사용하여 파일 및 폴더 업로드](#) 단원을 참조하세요.
4. Personalize에 버킷의 데이터를 읽을 수 있는 권한을 부여합니다. 자세한 내용은 [Personalize에 S3 리소스에 대한 액세스 권한 부여](#) 단원을 참조하세요.

도메인 데이터 세트 그룹 시작하기

이 시작하기 자습서에서는 VIDEO_ON_DEMAND 도메인의 도메인 데이터 세트 그룹을 만들고, CSV 파일에서 상호작용 데이터를 가져온 다음, 가장 적합한 추천 제품 사용 사례를 사용하여 추천자를 생성합니다. 그런 다음 추천자를 사용하여 사용자를 위한 개인 맞춤형 영화 추천을 받습니다. 이 자습서에서는 600명의 사용자가 제공한 9,700개 영화에 대한 100,000개의 영화 평점으로 구성된 과거 데이터를 사용합니다.

시작하려면 [시작하기 전제 조건](#)을 완료한 다음 Amazon Personalize 리소스를 생성하는 방법에 따라, [도메인 데이터세트 그룹 시작하기\(콘솔\)](#), [도메인 데이터 세트 그룹 시작하기\(Python용 SDK\(Boto3\)\)](#), [도메인 데이터 세트 그룹 시작하기\(Java 2.x용 SDK\)](#) 또는 [도메인 데이터 세트 그룹 시작하기\(JavaScript v3용 SDK\)](#) 로 이동합니다.

시작하기 연습을 마쳤을 때 불필요한 요금이 발생하지 않도록 하려면 [리소스 정리](#)의 단계에 따라 생성한 리소스를 삭제합니다.

주제

- [도메인 데이터세트 그룹 시작하기\(콘솔\)](#)
- [도메인 데이터 세트 그룹 시작하기\(Java 2.x용 SDK\)](#)
- [도메인 데이터 세트 그룹 시작하기\(Python용 SDK\(Boto3\)\)](#)
- [도메인 데이터 세트 그룹 시작하기\(JavaScript v3용 SDK\)](#)

도메인 데이터세트 그룹 시작하기(콘솔)

이 연습에서는 Personalize 콘솔을 사용하여 도메인 데이터세트 그룹과, 지정한 사용자에게 대해 추천 영화를 반환하는 추천자를 생성합니다.

이 연습을 시작하기 전에 [시작하기 전제 조건](#) 단원을 복습합니다.

시작하기 연습을 마쳤을 때 불필요한 요금이 발생하지 않도록 하려면 [리소스 정리](#)의 단계에 따라 생성한 리소스를 삭제합니다.

1단계: 도메인 데이터세트 그룹 생성

이 절차에서는 VIDEO_ON_DEMAND 도메인의 도메인 데이터 세트 그룹을 만들고, VIDEO_ON_DEMAND 도메인의 기본 스키마를 사용하여 항목 상호 작용 데이터 세트를 만들고, [학습 데이터 생성\(도메인 데이터세트 그룹\)](#)에서 생성한 항목 상호 작용 데이터를 가져옵니다.

도메인 데이터세트 그룹을 생성하려면

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 탐색 창에서 데이터세트 그룹 생성을 선택합니다.
3. 데이터세트 그룹 세부 정보에서 데이터세트 그룹의 이름을 지정합니다.
4. 도메인의 경우 온디맨드 비디오를 선택합니다. 선택한 도메인에 따라 데이터를 가져올 때 사용하는 기본 스키마가 결정됩니다. 또한 추천자가 사용할 수 있는 사용 사례도 결정됩니다. 다음과 유사한 화면이 표시되어야 합니다.

[Amazon Personalize](#) > Create dataset group

Create dataset group [Info](#)

A dataset group is a container for Amazon Personalize resources, including datasets, domain recommenders, and custom resources.

Dataset group details

Name

The name you enter here distinguishes this dataset group from others.

gs-vod-dataset-group

The dataset group name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Domain

Choose a domain for your use cases.

E-commerce

Grow your business by recommending the right products at the right time.

Video on demand

Increase engagement by recommending relevant content to your users.

Custom

Create and manage custom resources for your use cases.

► Tags - optional (0) [Info](#)

A tag is an administrative label that you assign to resources to make it easier to manage them. Each tag consists of a key and an optional value. Use tags to search and filter your resources or track your costs.

Cancel

Create group

- 데이터세트 그룹 생성을 선택합니다. 개요 페이지가 나타납니다. [2단계: 데이터 가져오기](#)로 이동합니다.

2단계: 데이터 가져오기

이 절차에서는 기본 VIDEO_ON_DEMAND 도메인 스키마를 사용하여 항목 상호 작용 데이터 세트를 생성합니다. 그런 다음 [학습 데이터 생성\(도메인 데이터세트 그룹\)](#)에서 생성한 항목 상호 작용 데이터를 가져옵니다.

데이터를 가져오려면

1. 개요 페이지의 1단계에서 데이터세트를 만들고 데이터를 가져오고 데이터세트 만들기를 선택한 다음 항목 상호작용 데이터세트를 선택합니다.
2. Personalize 데이터세트로 직접 데이터 가져오기를 선택하고 다음을 선택합니다.
3. 항목 상호 작용 스키마 구성 페이지에서 데이터 세트 이름에 대해 항목 상호 작용 데이터 세트의 이름을 입력합니다.
4. 데이터세트 스키마의 경우 도메인의 기존 기본 스키마를 수정하여 새 도메인 스키마 만들기를 선택하고 스키마의 이름을 입력합니다. 스키마 정의가 업데이트되어 VIDEO_ON_DEMAND 도메인의 기본 스키마가 표시됩니다. 스키마는 변경하지 않고 그대로 둡니다. 다음과 유사한 화면이 표시되어야 합니다.

Configure item interactions schema [Info](#)

Dataset details

Dataset name

The name you enter here can help you distinguish this dataset import job from others.

The dataset name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Dataset schema

The schema you provide allows Amazon Personalize to understand and import your data.

- Create a new domain schema by modifying the existing default schema for your domain
- Use an existing domain related schema

Schema name

The name you enter here can help you distinguish this schema from others.

The schema name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Schema definition

Verify your data structure matches the following schema.

```
1 {
2   "type": "record",
3   "name": "Interactions",
4   "namespace": "com.amazonaws.personalize.schema",
5   "fields": [
6     {
7       "name": "USER_ID",
8       "type": "string"
9     },
10    {
11      "name": "ITEM_ID",
12      "type": "string"
13    },
14    {
15      "name": "TIMESTAMP",
16      "type": "long"
17    },

```

5. 다음을 선택합니다. 항목 상호 작용 데이터 세트 가져오기 작업 구성 페이지가 나타납니다.
6. 항목 상호 작용 데이터 세트 가져오기 작업 구성 페이지에서 데이터 가져오기 소스를 변경하지 않고 S3에서 데이터 가져오기로 그대로 둡니다.
7. 데이터세트 가져오기 작업 이름의 경우 가져오기 작업 이름을 지정합니다.
8. 데이터 가져오기 소스에서 Amazon Simple Storage Service (S3) 의 데이터 저장 위치를 지정합니다. 다음 구문을 사용합니다.

s3://<name of your S3 bucket>/<folder path>/<CSV filename>

9. IAM 역할에서 IAM 서비스 역할에 대해 사용자 지정 IAM 역할 ARN 입력을 선택하고 [Personalize에 대한 IAM 역할 생성](#)에서 생성한 역할의 리소스 이름(ARN)을 입력합니다. 다음과 유사한 화면이 표시되어야 합니다.

Configure item interactions dataset import job Info

Dataset import job details

Data import source

Import data from S3
Specify the location where your data is stored in S3.

Incrementally import data with APIs
Incrementally import item interactions data with the event ingestion SDK.


Dataset import job name

The name you enter here can help you distinguish this dataset import job from others.

`my-dataset-import-job-name`

The dataset import job name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ (hyphen).

Data import source

 **Additional S3 bucket policy required**
In addition to the IAM service role defined above, Amazon Personalize also requires you to add a bucket policy to the S3 bucket containing your data files so that it can process them. Follow the instructions [described here](#) to add the required bucket policy to your S3 bucket.

Data location Info

Choose the S3 location of your data.

`s3://bucket/path-to-your-data/`

Your file needs to be in a CSV format and reflect the schema.

IAM Role

IAM service role

Amazon Personalize requires permissions to access your S3 bucket. Choose an existing role with access or create a role in the IAM console with the [AmazonPersonalizeFullAccess](#) IAM policy attached.

Enter a custom IAM role ARN ▼

Custom IAM role ARN

`arn:aws:iam::YourAccountID:role/YourRole`

10. 데이터를 가져오려면 가져오기 시작을 선택합니다. 도메인 데이터세트 그룹의 개요 페이지가 나타납니다. 데이터세트 설정 섹션에서 가져오기의 상태를 확인합니다. 상태가 Interaction data active이면 [3단계: 추천자 생성](#)로 이동합니다.

3단계: 추천자 생성

이 절차에서는 VIDEO_ON_DEMAND 도메인의 가장 적합한 추천 제품 사용 사례에 대한 추천자를 생성합니다.

추천자를 생성하려면

1. 도메인 데이터 세트 그룹에 대한 개요 페이지의 3단계에서 온디맨드 비디오 추천자 사용 탭을 선택하고 추천자 생성을 선택합니다.
2. 사용 사례 선택 페이지에서 가장 적합한 제품을 선택하고 추천자 이름을 입력합니다. 다음과 유사한 화면이 표시되어야 합니다.

Choose use case [Info](#)

You use recommenders to get recommendations for specific e-commerce use cases. Amazon Personalize trains the models backing each recommender with the optimal configurations for these use cases.

Video on demand recommenders

Video on demand use cases to create recommenders for

Amazon Personalize will create a recommender for each selected use case.

Because you watched X

Get recommendations for videos that other users also watched based on a video you specify.

Recommender name

my-recommender-name

The recommender name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

More like X

Get recommendations for videos that are similar to a video you specify.

Recommender name

my-recommender-name

The recommender name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Most popular

Get recommendations for videos that have been watched by the most users.

Recommender name

my-recommender-name

The recommender name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Top picks for you

Get personalized content recommendations for a user you specify.

Recommender name

my-recommender-name

The recommender name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Trending now - new

Get recommendations for videos that are trending now.

Recommender name

my-recommender-name

The recommender name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

3. 다음을 선택합니다.
4. 고급 구성 페이지의 필드를 변경하지 않고 그대로 두고 다음을 선택합니다.
5. 추천자 세부 정보를 검토하고 추천자 생성을 선택하여 추천자를 생성하십시오.

추천자 페이지에서 각 추천자의 상태를 모니터링할 수 있습니다. 추천자 상태가 활성화되면 이 상태를 사용하여 추천을 받을 수 있습니다. [4단계: 추천 받기](#)

4단계: 추천 받기

이 절차에서는 이전 단계에서 생성한 추천자를 사용하여 추천을 받습니다.

추천을 받으려면

1. 도메인 데이터세트 그룹의 개요 페이지의 탐색 창에서 추천자를 선택합니다.
2. 추천자 페이지에서 추천자를 선택합니다.
3. 오른쪽 상단에서 테스트를 선택합니다.
4. 추천 파라미터에 사용자 ID를 입력합니다. 다른 필드는 바꾸지 않고 그대로 둡니다.
5. 추천 받기를 선택합니다. 해당 사용자의 상위 25개 추천이 포함된 표가 나타납니다. 다음과 유사한 화면이 표시되어야 합니다.

Test recommender

Recommendation parameters

User ID

This is the USER_ID you want to get personalized re-ranked item recommendations for. This USER_ID needs to be present in your user-interactions or user dataset.

Filter name- *optional*

Choose an existing filter to apply to your recommendations or create a new filter.



[Create new filter](#)

► **Promotion - *optional* info**

Define additional business rules to promote a subset of items in recommendations. The promotion filter you specify applies to these items instead of any filter you specify above.

Recommendations (25)

Up to 25 recommendations are displayed. If you applied a promotion, promoted items are distributed randomly.

Recommendation ID

RID-4d12cd84-7d83-4dd9-b849-158b3e8f9ab8

Item ID

592

380

2571

590

150

296

318

780

도메인 데이터 세트 그룹 시작하기(Java 2.x용 SDK)

이 가이드에서는 Java 2.x용 SDK를 사용하여 VIDEO_ON_DEMAND 도메인에 대한 도메인 데이터 세트 그룹을 생성하는 방법을 보여줍니다. 이 자습서에서는 가장 적합한 추천 제품 사용 사례를 위한 추천자를 생성합니다.

이 시작하기 연습을 완료한 후 불필요한 비용이 발생하지 않도록 하려면 [리소스 정리](#)에서 자습서에서 생성한 리소스를 삭제하는 방법에 대한 정보를 참조하세요.

필수 조건

이 자습서를 완료하기 위한 필수 조건 단계는 다음과 같습니다.

- [시작하기 전제 조건](#)를 완료하여 필요한 권한을 설정하고 학습 데이터를 생성합니다. [도메인 데이터 세트 그룹 시작하기\(콘솔\)](#)를 완료한 경우에도 동일한 소스 데이터를 재사용할 수 있습니다. 고유한 소스 데이터를 사용하는 경우 필수 조건 단계와 같이 데이터의 형식이 지정되었는지 확인합니다.
- 개발자 안내서의 설정 프로시저에 지정된 대로 Java 2.x용 SDK 환경 AWS 및 자격 증명을 [설정합니다](#). AWS SDK for Java 2.x AWS SDK for Java 2.x

튜토리얼

다음에 이어지는 단계에서는 Amazon Personalize 패키지를 사용하도록 프로젝트를 설정하고 Amazon Personalize Java 2.x용 SDK 클라이언트를 생성합니다. 그런 다음 데이터를 가져오고, 가장 적합한 추천 제품 사용 사례를 위한 추천자를 생성한 다음 추천을 받습니다.

1단계: Amazon Personalize 패키지를 사용하도록 프로젝트 설정

필수 조건을 완료한 후, Amazon Personalize 종속 항목을 pom.xml 파일에 추가하고 Amazon Personalize 패키지를 가져옵니다.

1. pom.xml 파일에 다음 종속 항목을 추가합니다. 최신 버전 번호가 예제 코드와 다를 수 있습니다.

```
<dependency>
  <groupId>software.amazon.awssdk</groupId>
  <artifactId>personalize</artifactId>
  <version>2.16.83</version>
</dependency>
<dependency>
  <groupId>software.amazon.awssdk</groupId>
```

```
<artifactId>personalizeruntime</artifactId>
<version>2.16.83</version>
</dependency>
<dependency>
  <groupId>software.amazon.awssdk</groupId>
  <artifactId>personalizeevents</artifactId>
  <version>2.16.83</version>
</dependency>
```

2. 프로젝트에 다음 가져오기 명령문을 추가합니다.

```
// import client packages
import software.amazon.awssdk.services.personalize.PersonalizeClient;
import software.amazon.awssdk.services.personalizeruntime.PersonalizeRuntimeClient;
// Amazon Personalize exception package
import software.amazon.awssdk.services.personalize.model.PersonalizeException;
// schema packages
import software.amazon.awssdk.services.personalize.model.CreateSchemaRequest;
// dataset group packages
import software.amazon.awssdk.services.personalize.model.CreateDatasetGroupRequest;
import software.amazon.awssdk.services.personalize.model.DescribeDatasetGroupRequest;
// dataset packages
import software.amazon.awssdk.services.personalize.model.CreateDatasetRequest;
// dataset import job packages
import
  software.amazon.awssdk.services.personalize.model.CreateDatasetImportJobRequest;
import software.amazon.awssdk.services.personalize.model.DataSource;
import software.amazon.awssdk.services.personalize.model.DatasetImportJob;
import
  software.amazon.awssdk.services.personalize.model.DescribeDatasetImportJobRequest;
// recommender packages
import software.amazon.awssdk.services.personalize.model.CreateRecommenderRequest;
import software.amazon.awssdk.services.personalize.model.CreateRecommenderResponse;
import software.amazon.awssdk.services.personalize.model.DescribeRecommenderRequest;
// get recommendations packages
import
  software.amazon.awssdk.services.personalizeruntime.model.GetRecommendationsRequest;
import
  software.amazon.awssdk.services.personalizeruntime.model.GetRecommendationsResponse;
import software.amazon.awssdk.services.personalizeruntime.model.PredictedItem;
// Java time utility package
import java.time.Instant;
```

2단계: Amazon Personalize 클라이언트 생성

Amazon Personalize 종속 항목을 pom.xml 파일에 추가하고 필요한 패키지를 가져온 후 다음과 같은 Amazon Personalize 클라이언트를 생성합니다.

```
PersonalizeClient personalizeClient = PersonalizeClient.builder()
    .region(region)
    .build();

PersonalizeRuntimeClient personalizeRuntimeClient = PersonalizeRuntimeClient.builder()
    .region(region)
    .build();
```

3단계: 데이터 가져오기

Amazon Personalize 클라이언트를 초기화한 후에는 [시작하기 전제 조건](#) 완료 시 생성한 과거 데이터를 가져옵니다. 과거 데이터를 Amazon Personalize로 가져오려면 다음과 같이 합니다.

1. 다음 Avro 스키마를 작업 디렉터리에 JSON 파일로 저장합니다. 이 스키마는 [학습 데이터 생성\(도메인 데이터세트 그룹\)](#)을 완료할 때 만든 CSV 파일의 열과 일치합니다.

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "EVENT_TYPE",
      "type": "string"
    },
    {
      "name": "TIMESTAMP",
      "type": "long"
    }
  ]
}
```



```

    ],
    "version": "1.0"
  }

```

2. 다음 `createDomainSchema` 메서드를 사용하여 Amazon Personalize에서 도메인 스키마를 생성합니다. Amazon Personalize 서비스 클라이언트, 스키마 이름, 도메인에 대한 `VIDEO_ON_DEMAND`, 이전 단계에서 생성한 스키마 JSON 파일의 파일 경로를 파라미터로 전달합니다. 메서드가 새 스키마의 Amazon 리소스 이름(ARN)을 반환합니다. 나중에 사용하기 위해 이 ARN을 저장합니다.

```

    public static String createDomainSchema(PersonalizeClient personalizeClient,
        String schemaName, String domain,
        String filePath) {

        String schema = null;
        try {
            schema = new String(Files.readAllBytes(Paths.get(filePath)));
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }

        try {
            CreateSchemaRequest createSchemaRequest = CreateSchemaRequest.builder()
                .name(schemaName)
                .domain(domain)
                .schema(schema)
                .build();

            String schemaArn =
                personalizeClient.createSchema(createSchemaRequest).schemaArn();

            System.out.println("Schema arn: " + schemaArn);

            return schemaArn;

        } catch (PersonalizeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }

```

3. 데이터 세트 그룹을 생성합니다. 다음 `createDomainDatasetGroup` 메서드를 사용하여 도메인 데이터 세트 그룹을 생성합니다. Amazon Personalize 서비스 클라이언트, 데이터 세트 그룹 이름을

파라미터로 전달하고 도메인에 대해 VIDEO_ON_DEMAND을 전달합니다. 메서드가 새 데이터 세트 그룹의 ARN을 반환합니다. 나중에 사용하기 위해 이 ARN을 저장합니다.

```
public static String createDomainDatasetGroup(PersonalizeClient
personalizeClient,
    String datasetGroupName,
    String domain) {

    try {
        CreateDatasetGroupRequest createDatasetGroupRequest =
CreateDatasetGroupRequest.builder()
            .name(datasetGroupName)
            .domain(domain)
            .build();

        return
personalizeClient.createDatasetGroup(createDatasetGroupRequest).datasetGroupArn();
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}
```

4. 항목 상호 작용 데이터 세트를 생성합니다. 다음 createDataset 메서드를 사용하여 항목 상호 작용 데이터 세트를 생성합니다. Amazon Personalize 서비스 클라이언트, 데이터 세트 이름, 스키마의 ARN, 데이터 세트 그룹의 ARN, 데이터 세트 유형에 대한 Interactions을 파라미터로 전달합니다. 메서드가 새 데이터 세트의 ARN을 반환합니다. 나중에 사용하기 위해 이 ARN을 저장합니다.

```
public static String createDataset(PersonalizeClient personalizeClient,
    String datasetName,
    String datasetGroupArn,
    String datasetType,
    String schemaArn) {
    try {
        CreateDatasetRequest request = CreateDatasetRequest.builder()
            .name(datasetName)
            .datasetGroupArn(datasetGroupArn)
            .datasetType(datasetType)
            .schemaArn(schemaArn)
            .build();

        String datasetArn = personalizeClient.createDataset(request)
            .datasetArn();
    }
}
```

```

        System.out.println("Dataset " + datasetName + " created.");
        return datasetArn;

    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}

```

5. 데이터 세트 가져오기 작업을 사용하여 데이터를 가져옵니다. 다음 `createPersonalizeDatasetImportJob` 메서드를 사용하여 데이터 세트 가져오기 작업을 생성합니다.

Amazon Personalize 서비스 클라이언트, 작업 이름, 상호작용 데이터 세트의 ARN을 파라미터로 전달합니다. 학습 데이터를 저장한 Amazon S3 버킷 경로(`s3://bucket name/folder name/ratings.csv`)와 서비스 역할의 ARN을 전달합니다. [시작하기 전제 조건](#)의 일부로 이 역할을 생성했습니다. 메서드가 데이터 세트 가져오기 작업의 ARN을 반환합니다. 나중에 사용할 수 있도록 저장할 수도 있습니다.

```

public static String createPersonalizeDatasetImportJob(PersonalizeClient
personalizeClient,
    String jobName,
    String datasetArn,
    String s3BucketPath,
    String roleArn) {

    long waitInMilliseconds = 60 * 1000;
    String status;
    String datasetImportJobArn;

    try {
        DataSource importDataSource = DataSource.builder()
            .dataLocation(s3BucketPath)
            .build();

        CreateDatasetImportJobRequest createDatasetImportJobRequest =
        CreateDatasetImportJobRequest.builder()
            .datasetArn(datasetArn)
            .dataSource(importDataSource)
            .jobName(jobName)
            .roleArn(roleArn)

```

```

        .build());

        datasetImportJobArn =
personalizeClient.createDatasetImportJob(createDatasetImportJobRequest)
        .datasetImportJobArn();
        DescribeDatasetImportJobRequest describeDatasetImportJobRequest =
DescribeDatasetImportJobRequest.builder()
        .datasetImportJobArn(datasetImportJobArn)
        .build();

        long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

        while (Instant.now().getEpochSecond() < maxTime) {

            DatasetImportJob datasetImportJob = personalizeClient
                .describeDatasetImportJob(describeDatasetImportJobRequest)
                .datasetImportJob();

            status = datasetImportJob.status();
            System.out.println("Dataset import job status: " + status);

            if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
                break;
            }
            try {
                Thread.sleep(waitInMilliseconds);
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
        return datasetImportJobArn;

    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}

```

4단계: 추천자 생성

데이터 세트 가져오기 작업이 완료되면 추천자를 만들 준비가 된 것입니다. 추천자를 생성하려면 다음 `createRecommender` 메서드를 사용합니다. Amazon Personalize 서비스 클

라이언트, 추천자 이름, 데이터세트 그룹의 Amazon 리소스 이름(ARN), 레시피 ARN에 대한 `arn:aws:personalize:::recipe/aws-vod-top-picks`을 파라미터로 전달합니다. 메서드가 새 추천자의 ARN을 반환합니다. 나중에 사용하기 위해 이 ARN을 저장합니다.

```
public static String createRecommender(PersonalizeClient personalizeClient,
    String name,
    String datasetGroupArn,
    String recipeArn) {

    long maxTime = 0;
    long waitInMilliseconds = 30 * 1000; // 30 seconds
    String recommenderStatus = "";

    try {
        CreateRecommenderRequest createRecommenderRequest =
        CreateRecommenderRequest.builder()
            .datasetGroupArn(datasetGroupArn)
            .name(name)
            .recipeArn(recipeArn)
            .build();

        CreateRecommenderResponse recommenderResponse = personalizeClient
            .createRecommender(createRecommenderRequest);
        String recommenderArn = recommenderResponse.recommenderArn();
        System.out.println("The recommender ARN is " + recommenderArn);

        DescribeRecommenderRequest describeRecommenderRequest =
        DescribeRecommenderRequest.builder()
            .recommenderArn(recommenderArn)
            .build();

        maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

        while (Instant.now().getEpochSecond() < maxTime) {

            recommenderStatus =
            personalizeClient.describeRecommender(describeRecommenderRequest).recommender()
                .status();
            System.out.println("Recommender status: " + recommenderStatus);

            if (recommenderStatus.equals("ACTIVE") ||
            recommenderStatus.equals("CREATE FAILED")) {
                break;
            }
        }
    }
}
```

```
    }
    try {
        Thread.sleep(waitInMilliseconds);
    } catch (InterruptedException e) {
        System.out.println(e.getMessage());
    }
}
return recommenderArn;

} catch (PersonalizeException e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
return "";
}
```

5단계: 추천 받기

추천자를 만든 후에는 이를 사용하여 추천을 받을 수 있습니다. 사용자에게 대한 추천을 받으려면 다음 `getRecs` 메서드를 사용합니다. Amazon Personalize 런타임 클라이언트, 이전 단계에서 생성한 추천자의 Amazon 리소스 이름(ARN) 및 사용자 ID(예: 123)를 파라미터로 전달합니다. 메서드가 추천 항목 목록을 화면에 인쇄합니다.

```
public static void getRecs(PersonalizeRuntimeClient personalizeRuntimeClient,
    String recommenderArn,
    String userId) {

    try {
        GetRecommendationsRequest recommendationsRequest =
        GetRecommendationsRequest.builder()
            .recommenderArn(recommenderArn)
            .numResults(20)
            .userId(userId)
            .build();

        GetRecommendationsResponse recommendationsResponse =
        personalizeRuntimeClient
            .getRecommendations(recommendationsRequest);
        List<PredictedItem> items = recommendationsResponse.itemList();

        for (PredictedItem item : items) {
            System.out.println("Item Id is : " + item.itemId());
            System.out.println("Item score is : " + item.score());
        }
    }
}
```

```

    }
  } catch (AwsServiceException e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
  }
}

```

도메인 데이터 세트 그룹 시작하기(Python용 SDK(Boto3))

이 가이드에서는 Python용 SDK(Boto3)를 사용하여 VIDEO_ON_DEMAND 도메인에 대한 도메인 데이터 세트 그룹을 만드는 방법을 보여줍니다. 이 자습서에서는 가장 적합한 추천 제품 사용 사례를 위한 추천자를 생성합니다.

이 시작하기 연습을 완료한 후 불필요한 비용이 발생하지 않도록 하려면 이 자습서에서 생성한 리소스를 삭제합니다. 자세한 내용은 [리소스 정리](#) 단원을 참조하세요.

주제

- [사전 조건](#)
- [튜토리얼](#)
- [주피터\(IPython\) 노트북으로 Amazon Personalize API 사용 시작하기](#)

사전 조건

다음은 이 안내서에서 Python 예제를 사용하기 위한 필수 조건 단계입니다.

- [시작하기 전제 조건](#)를 완료하여 필요한 권한을 설정하고 학습 데이터를 생성합니다. 고유한 소스 데이터를 사용하는 경우 필수 조건 단계와 같이 데이터의 형식이 지정되었는지 확인합니다.
- [AWS SDK 설정](#)에 지정된 대로 AWS SDK for Python (Boto3) 환경을 설정합니다.

튜토리얼

다음에 이어지는 단계에서는 환경을 확인하고 Amazon Personalize용 Python용 SDK(Boto3) 클라이언트를 생성합니다. 그런 다음 데이터를 가져오고, 가장 적합한 추천 제품 사용 사례를 위한 추천자를 생성한 다음 추천을 받습니다.

1단계: Python 환경을 확인하고 boto3 클라이언트 만들기

필수 조건을 완료한 후 다음 Python 예제를 실행하여 환경이 올바르게 구성되었는지 확인합니다. 또한 이 코드는 이 자습서에서 사용하는 Amazon Personalize boto3 클라이언트를 생성합니다. 환경이 올바

르게 구성된 경우 사용 가능한 레시피 목록이 표시되고 이 자습서의 다른 Python 예제를 실행할 수 있습니다.

```
import boto3

personalizeRt = boto3.client('personalize-runtime')
personalize = boto3.client('personalize')

response = personalize.list_recipes()

for recipe in response['recipes']:
    print (recipe)
```

2단계: 데이터 가져오기

Amazon Personalize boto3 클라이언트를 생성하고 환경을 확인했으면, [시작하기 전제 조건](#) 완료 시 생성한 과거 데이터를 가져옵니다. 과거 데이터를 Amazon Personalize로 가져오려면 다음과 같이 합니다.

1. 다음 코드를 사용하여 Amazon Personalize에서 스키마를 생성합니다. `gs-domain-interactions-schema`을 스키마의 이름으로 바꿉니다.

```
import json
schema = {
    "type": "record",
    "name": "Interactions",
    "namespace": "com.amazonaws.personalize.schema",
    "fields": [
        {
            "name": "USER_ID",
            "type": "string"
        },
        {
            "name": "ITEM_ID",
            "type": "string"
        },
        {
            "name": "EVENT_TYPE",
            "type": "string"
        },
        {
            "name": "TIMESTAMP",
```



```

        "type": "long"
    }
],
"version": "1.0"
}

create_interactions_schema_response = personalize.create_schema(
    name='gs-domain-interactions-schema',
    schema=json.dumps(schema),
    domain='VIDEO_ON_DEMAND'
)

interactions_schema_arn = create_interactions_schema_response['schemaArn']
print(json.dumps(create_interactions_schema_response, indent=2))

```

2. 다음 코드를 사용하여 데이터 세트 그룹을 생성합니다. `dataset group name`을 데이터 세트 그룹의 이름으로 바꿉니다.

```

response = personalize.create_dataset_group(
    name = 'dataset group name',
    domain = 'VIDEO_ON_DEMAND'
)
dsg_arn = response['datasetGroupArn']

description = personalize.describe_dataset_group(datasetGroupArn = dsg_arn)
['datasetGroup']

print('Name: ' + description['name'])
print('ARN: ' + description['datasetGroupArn'])
print('Status: ' + description['status'])

```

3. 다음 코드를 사용하여 새 데이터 세트 그룹에 항목 상호 작용 데이터 세트를 만듭니다. 데이터 세트에 이름을 지정하고 이전 단계의 `schema_arn` 및 `dataset_group_arn`를 입력합니다.

```

response = personalize.create_dataset(
    name = 'interactions-dataset-name',
    schemaArn = interactions_schema_arn,
    datasetGroupArn = dsg_arn,
    datasetType = 'INTERACTIONS'
)

dataset_arn = response['datasetArn']

```

4. 다음 코드를 사용한 데이터 세트 가져오기 작업으로 데이터를 가져옵니다. 코드는 `describe_dataset_import_job` 메서드를 사용하여 작업 상태를 추적합니다.

작업 이름, 이전 단계의 `dataset_arn`, 학습 데이터를 저장한 Amazon S3 버킷 경로 (`s3://bucket name/folder name/ratings.csv`), IAM 서비스 역할의 ARN 등을 파라미터로 전달합니다. [시작하기 전제 조건](#)의 일부로 이 역할을 생성했습니다. 버킷에 액세스할 수 있는 권한이 Amazon Personalize에 필요합니다. 액세스 권한 부여에 대한 자세한 내용은 [Personalize에 S3 리소스에 대한 액세스 권한 부여](#) 단원을 참조하세요.

```
import time
response = personalize.create_dataset_import_job(
    jobName = 'JobName',
    datasetArn = 'dataset_arn',
    dataSource = {'dataLocation': 's3://bucket/file.csv'},
    roleArn = 'role_arn'
)

dataset_interactions_import_job_arn = response['datasetImportJobArn']

description = personalize.describe_dataset_import_job(
    datasetImportJobArn = dataset_interactions_import_job_arn)['datasetImportJob']

print('Name: ' + description['jobName'])
print('ARN: ' + description['datasetImportJobArn'])
print('Status: ' + description['status'])

max_time = time.time() + 3*60*60 # 3 hours
while time.time() < max_time:
    describe_dataset_import_job_response = personalize.describe_dataset_import_job(
        datasetImportJobArn = dataset_interactions_import_job_arn
    )
    status = describe_dataset_import_job_response["datasetImportJob"]['status']
    print("Interactions DatasetImportJob: {}".format(status))

    if status == "ACTIVE" or status == "CREATE FAILED":
        break

    time.sleep(60)
```

4단계: 추천자 생성

데이터 세트 가져오기 작업이 완료되면 추천자를 만들 준비가 된 것입니다. 추천자를 생성하려면 다음 코드를 사용합니다. 추천자 이름, 데이터 세트 그룹의 Amazon 리소스 이름(ARN), 레시피 ARN에 대한 `arn:aws:personalize:::recipe/aws-vod-top-picks`을 파라미터로 전달합니다. 코드는 `describe_committer` 메서드를 사용하여 추천자의 상태를 추적합니다.

```
import time
create_recommender_response = personalize.create_recommender(
    name = 'gs-python-top-picks',
    recipeArn = 'arn:aws:personalize:::recipe/aws-vod-top-picks',
    datasetGroupArn = dsg_arn
)
recommender_arn = create_recommender_response['recommenderArn']

print('Recommender ARN:' + recommender_arn)
max_time = time.time() + 3*60*60 # 3 hours
while time.time() < max_time:

    version_response = personalize.describe_recommender(
        recommenderArn = recommender_arn
    )
    status = version_response["recommender"]["status"]

    if status == "ACTIVE":
        print("Creation succeeded for {}".format(recommender_arn))

    elif status == "CREATE FAILED":
        print("Creation failed for {}".format(recommender_arn))

    if status == "ACTIVE":
        break
    else:
        print("Recommender creation is still in progress")

    time.sleep(60)
```

5단계: 추천 받기

추천자를 만든 후에는 이를 사용하여 다음 코드를 통해 추천을 받을 수 있습니다. 이전 단계에서 생성한 추천자의 Amazon 리소스 이름(ARN)과 사용자 ID(예: 123)를 파라미터로 전달합니다. 메서드가 추천 항목 목록을 인쇄합니다.

```

response = personalizeRt.get_recommendations(
    recommenderArn = "arn:aws:personalize:us-west-2:014025156336:recommender/gs-python-
top-picks-89",
    userId = '123'
)
print("Recommended items")
for item in response['itemList']:
    print (item['itemId'])

```

주피터(IPython) 노트북으로 Amazon Personalize API 사용 시작하기

주피터 노트북으로 도메인 데이터 세트 그룹 생성을 시작하려면 [Amazon Personalize 샘플](#) 리포지토리의 [notebooks_managed_domain](#) 폴더에 있는 노트북 시리즈를 복제하거나 다운로드합니다. 노트북에서는 Amazon Personalize를 사용하여 학습 데이터를 가져오고, 추천자를 생성하고, 추천을 받는 과정을 안내합니다.

Note

노트북을 시작하기 전에 [README.md](#)의 단계에 따라 환경을 구축해야 합니다.

도메인 데이터 세트 그룹 시작하기(JavaScript v3용 SDK)

이 가이드에서는 AWS SDK for JavaScript v3를 사용하여 VIDEO_ON_DEMAND 도메인에 대한 도메인 데이터 세트 그룹을 만드는 방법을 보여줍니다. 이 자습서에서는 가장 적합한 추천 제품 사용 사례를 위한 추천자를 생성합니다.

GitHub에서 이 자습서에 사용된 코드를 보려면 AWS SDK 코드 예제 리포지토리에서 [JavaScript v3용 SDK의 Amazon Personalize 코드 예제](#)를 참조하세요.

이 시작하기 연습을 완료한 후 불필요한 비용이 발생하지 않도록 하려면 이 자습서에서 생성한 리소스를 삭제합니다. 자세한 내용은 [리소스 정리](#) 단원을 참조하세요.

주제

- [사전 조건](#)
- [튜토리얼](#)

사전 조건

이 자습서를 완료하기 위한 필수 조건 단계는 다음과 같습니다.

- [시작하기 전제 조건](#)를 완료하여 필요한 권한을 설정하고 학습 데이터를 생성합니다. [도메인 데이터 세트 그룹 시작하기\(콘솔\)](#)를 완료한 경우에도 동일한 소스 데이터를 재사용할 수 있습니다. 고유한 소스 데이터를 사용하는 경우 필수 조건 단계와 같이 데이터의 형식이 지정되었는지 확인합니다.
- AWS SDK for JavaScript 개발자 안내서의 [JavaScript용 SDK 설정](#) 절차에 지정된 대로 JavaScript용 SDK 및 AWS 보안 인증을 설정합니다.

튜토리얼

다음 단계에서는 필수 종속 항목을 설치합니다. 그런 다음 데이터 세트 그룹을 만들고, 데이터를 가져 오고, 가장 적합한 추천 제품 사용 사례를 위한 추천자를 생성한 다음 추천을 받습니다.

Node.js를 사용하는 경우 샘플을 JavaScript 파일로 저장한 다음 `node <fileName.js>`을 실행하여 각 코드 샘플을 실행할 수 있습니다.

1단계: Amazon Personalize 종속 항목 설치

필수 조건을 완료한 후에는 다음과 같은 Amazon Personalize 종속 항목을 설치합니다.

- `@aws-sdk/client-personalize`
- `@aws-sdk/client-personalize-runtime`
- `@aws -sdk/client-personalize-events`(이 자습서에서는 선택 사항이지만 추천자를 만든 후 [이벤트를 기록](#)하려는 경우 필수)

다음은 사용할 수 있는 `package.json` 파일의 예제입니다. Node.js를 사용하여 종속 항목을 설치하려면 `package.json` 파일을 저장한 위치로 이동하여 `npm install`를 실행합니다.

```
{
  "name": "personalize-js-project",
  "version": "1.0.0",
  "description": "personalize operations",
  "type": "module",
  "author": "Author Name <email@address.com>",
  "license": "ISC",
  "dependencies": {
    "@aws-sdk/client-personalize": "^3.350.0",
    "@aws-sdk/client-personalize-events": "^3.350.0",
  }
}
```

```

    "@aws-sdk/client-personalize-runtime": "^3.350.0",
    "fs": "^0.0.1-security"
  },
  "compilerOptions": {
    "resolveJsonModule": true,
    "esModuleInterop": true
  }
}

```

2단계: Amazon Personalize 클라이언트 생성

종속 항목을 설치한 후 Amazon Personalize 클라이언트를 생성합니다. 이 자습서의 코드 샘플에서는 `libs`라는 디렉터리에 `personalizeClients.js`라는 이름의 파일에 클라이언트를 생성한다고 가정합니다.

다음은 `personalizeClient.js` 파일의 예제입니다.

```

import { PersonalizeClient } from "@aws-sdk/client-personalize";
import { PersonalizeRuntimeClient } from "@aws-sdk/client-personalize-runtime";
import { PersonalizeEventsClient } from "@aws-sdk/client-personalize-events";
// Set your AWS region.
const REGION = "region"; //e.g. "us-east-1"

const personalizeClient = new PersonalizeClient({ region: REGION});
const personalizeEventsClient = new PersonalizeEventsClient({ region: REGION});
const personalizeRuntimeClient = new PersonalizeRuntimeClient({ region: REGION});

export { personalizeClient, personalizeEventsClient, personalizeRuntimeClient };

```

3단계: 데이터 가져오기

Amazon Personalize 클라이언트를 생성한 후에는 [시작하기 전제 조건](#) 완료 시 생성한 과거 데이터를 가져옵니다. 과거 데이터를 Amazon Personalize로 가져오려면 다음과 같이 합니다.

1. 다음 Avro 스키마를 작업 디렉터리에 JSON 파일로 저장합니다. 이 스키마는 [학습 데이터 생성\(도메인 데이터세트 그룹\)](#)을 완료할 때 만든 CSV 파일의 열과 일치합니다.

```

{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [

```

```
{
  "name": "USER_ID",
  "type": "string"
},
{
  "name": "ITEM_ID",
  "type": "string"
},
{
  "name": "EVENT_TYPE",
  "type": "string"
},
{
  "name": "TIMESTAMP",
  "type": "long"
}
],
"version": "1.0"
}
```

2. 다음 `createDomainSchema.js` 코드를 사용하여 Amazon Personalize에서 도메인 스키마를 생성합니다. `SCHEMA_PATH`을 방금 생성한 `schema.json` 파일의 경로로 바꿉니다. `createSchemaParam`을 업데이트하여 스키마의 이름을 지정하고 `domain`에 대해 `VIDEO_ON_DEMAND`을 지정합니다.

```
// Get service clients module and commands using ES6 syntax.
import { CreateSchemaCommand } from
  "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

import fs from 'fs';

let schemaFilePath = "SCHEMA_PATH";
let mySchema = "";

try {
  mySchema = fs.readFileSync(schemaFilePath).toString();
} catch (err) {
  mySchema = 'TEST' // for unit tests.
}
```

```
// Set the domain schema parameters.
export const createDomainSchemaParam = {
  name: 'NAME', /* required */
  schema: mySchema, /* required */
  domain: 'DOMAIN' /* required for a domain dataset group, specify ECOMMERCE or
VIDEO_ON_DEMAND */
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
CreateSchemaCommand(createDomainSchemaParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

3. 다음 `createDomainDatasetGroup.js` 코드를 사용하여 Amazon Personalize에서 도메인 데이터 세트 그룹을 생성합니다. `domainDatasetGroupParams`을 업데이트하여 데이터 세트 그룹의 이름을 지정하고 `domain`에 대해 `VIDEO_ON_DEMAND`을 지정합니다.

```
// Get service clients module and commands using ES6 syntax.
import { CreateDatasetGroupCommand } from
"@aws-sdk/client-personalize";
import { personalizeClient } from "../libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the domain dataset group parameters.
export const domainDatasetGroupParams = {
  name: 'NAME', /* required */
  domain: 'DOMAIN' /* required for a domain dsG, specify ECOMMERCE or
VIDEO_ON_DEMAND */
}

export const run = async () => {
  try {
```



```

    const response = await personalizeClient.send(new
CreateDatasetGroupCommand(domainDatasetGroupParams));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();

```

4. 다음 `createDataset.js` 코드를 사용하여 Amazon Personalize에서 항목 상호 작용 데이터 세트를 생성합니다. `createDatasetParam`을 업데이트하여 방금 생성한 데이터 세트 그룹 및 스키마의 Amazon 리소스 이름(ARN)을 지정하고, 데이터 세트에 이름을 지정한 다음, `datasetType`에 대해 `Interactions`을 지정합니다.

```

// Get service clients module and commands using ES6 syntax.
import { CreateDatasetCommand } from
"@aws-sdk/client-personalize";
import { personalizeClient } from "../libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the dataset's parameters.
export const createDatasetParam = {
  datasetGroupArn: 'DATASET_GROUP_ARN', /* required */
  datasetType: 'DATASET_TYPE', /* required */
  name: 'NAME', /* required */
  schemaArn: 'SCHEMA_ARN' /* required */
}

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
CreateDatasetCommand(createDatasetParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();

```

5. 다음 `createDatasetImportJob.js` 코드를 사용하여 데이터를 가져옵니다.

`datasetImportJobParam`을 업데이트하여 다음을 지정합니다.

- 작업 이름을 지정하고 상호작용 데이터 세트의 ARN을 지정합니다.
- `dataLocation`의 경우, 학습 데이터를 저장한 Amazon S3 버킷 경로(`s3://bucket name/folder name/ratings.csv`)를 지정합니다.
- `roleArn`의 경우, Amazon Personalize 서비스 역할에 대한 Amazon 리소스 이름을 지정합니다. [시작하기 전제 조건](#)의 일부로 이 역할을 생성했습니다.

```
// Get service clients module and commands using ES6 syntax.
import {CreateDatasetImportJobCommand } from
  "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the dataset import job parameters.
export const datasetImportJobParam = {
  datasetArn: 'DATASET_ARN', /* required */
  dataSource: { /* required */
    dataLocation: 'S3_PATH'
  },
  jobName: 'NAME', /* required */
  roleArn: 'ROLE_ARN' /* required */
}

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
      CreateDatasetImportJobCommand(datasetImportJobParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

4단계: 추천자 생성

데이터 세트 가져오기 작업이 완료되면 추천자를 만들 준비가 된 것입니다. 추천자를 생성하려면 다음 `createRecommender.js` 코드를 사용합니다. 다음과 같이 `createRecommenderParam`을 업데이트합니다. 추천자 이름을 지정하고, 데이터 세트 그룹의 ARN을 지정하고, `recipeArn`에 대해 `arn:aws:personalize:::recipe/aws-vod-top-picks`을 지정합니다.

```
// Get service clients module and commands using ES6 syntax.
import { CreateRecommenderCommand } from
  "@aws-sdk/client-personalize";
import { personalizeClient } from "../libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the recommender's parameters.
export const createRecommenderParam = {
  name: 'NAME', /* required */
  recipeArn: 'RECIPE_ARN', /* required */
  datasetGroupArn: 'DATASET_GROUP_ARN' /* required */
}

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
    CreateRecommenderCommand(createRecommenderParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

5단계: 추천 받기

추천자를 만든 후에는 이를 사용하여 추천을 받을 수 있습니다. 사용자에게 대한 추천을 받으려면 다음 `getRecommendations.js` 코드를 사용합니다. `getRecommendationsParam`을 업데이트하여 이전 단계에서 생성한 추천자의 ARN을 지정하고 사용자 ID(예: 123)를 지정합니다.

```
// Get service clients module and commands using ES6 syntax.
import { GetRecommendationsCommand } from
```

```

"@aws-sdk/client-personalize-runtime";
import { personalizeRuntimeClient } from "./libs/personalizeClients.js";
// Or, create the client here.
// const personalizeRuntimeClient = new PersonalizeRuntimeClient({ region: "REGION"});

// Set the recommendation request parameters.
export const getRecommendationsParam = {
  recommenderArn: 'RECOMMENDER_ARN', /* required */
  userId: 'USER_ID', /* required */
  numResults: 15 /* optional */
}

export const run = async () => {
  try {
    const response = await personalizeRuntimeClient.send(new
    GetRecommendationsCommand(getRecommendationsParam));
    console.log("Success!", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();

```

사용자 지정 데이터세트 그룹 시작하기

Important

이 자습서에서는 자동 학습을 사용하는 솔루션을 만듭니다. 기본적으로 모든 새 솔루션은 자동 학습을 사용합니다. 자동 교육을 사용하면 솔루션이 활성화되는 동안 교육 비용이 발생합니다. 불필요한 비용을 피하려면 완료 후 솔루션을 삭제해야 합니다. 자세한 정보는 [리소스 정리](#)를 참조하세요.

이 시작 가이드에서는 맞춤 데이터세트 그룹과 [사용자 맞춤화-v2 레시피](#) 레시피를 사용하여 사용자에게 개인화된 영화 추천을 제공하는 방법을 보여줍니다. 이 자습서에서는 600명의 사용자가 제공한 9,700개 영화에 대한 100,000개의 영화 평점으로 구성된 과거 데이터를 사용합니다.

시작하려면 [시작하기 전제 조건](#)을 완료한 다음 [시작하기\(콘솔\)](#), [시작하기\(AWS CLI\)](#), [시작하기\(Python용 SDK\(Boto3\)\)](#) 또는 [시작하기\(Java 2.x용 SDK\)](#) 중 하나로 진행합니다.

시작하기 연습을 마쳤을 때 불필요한 요금이 발생하지 않도록 하려면 [리소스 정리](#)의 단계에 따라 생성한 리소스를 삭제합니다.

주제

- [시작하기\(콘솔\)](#)
- [시작하기\(AWS CLI\)](#)
- [시작하기\(Python용 SDK\(Boto3\)\)](#)
- [시작하기\(Java 2.x용 SDK\)](#)

시작하기(콘솔)

이 연습에서는 Personalize 콘솔을 사용하여 지정한 사용자에게 대해 추천 영화를 반환하는 솔루션으로 사용자 지정 데이터세트 그룹을 생성합니다. 이 연습을 시작하기 전에 [시작하기 전제 조건](#) 단원을 복습합니다.

시작하기 연습을 마쳤을 때 불필요한 요금이 발생하지 않도록 하려면 [리소스 정리](#)의 단계에 따라 생성한 리소스를 삭제합니다.

1단계: 데이터세트 그룹 및 데이터세트 생성

이 절차에서는 가장 먼저 데이터세트 그룹을 생성합니다. 다음으로, 데이터 세트 그룹에 Amazon Personalize 항목 상호 작용 데이터 세트를 생성합니다.

데이터세트 그룹과 데이터세트를 생성하려면

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터세트 그룹 생성을 선택합니다.
3. 데이터세트 그룹 세부 정보의 데이터세트 그룹 이름에서 데이터세트 그룹의 이름을 지정합니다.
4. 도메인의 경우 사용자 지정을 선택합니다. 다음과 유사한 화면이 표시되어야 합니다.

Create dataset group [Info](#)

A dataset group is a container for Amazon Personalize resources, including datasets, domain recommenders, and custom resources.

Dataset group details

Name
The name you enter here distinguishes this dataset group from others.

The dataset group name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ (hyphen).

Domain
Choose a domain for your use cases.

E-commerce
Grow your business by recommending the right products at the right time.

Video on demand
Increase engagement by recommending relevant content to your users.

Custom
Create and manage custom resources for your use cases.

▶ **Tags - optional (0)** [Info](#)

A tag is a label that you assign to an **resource**. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your **costs**.

Cancel
Create group

5. 그룹 생성을 선택합니다. 개요 페이지가 나타납니다.
6. 1단계에서 데이터셋을 만들고 데이터를 가져오고 데이터셋 생성을 선택한 다음 항목 상호작용 데이터셋을 선택합니다.
7. Personalize 데이터셋으로 직접 데이터 가져오기를 선택하고 다음을 선택합니다.
8. 항목 상호 작용 데이터 세트 구성 페이지에서 데이터 세트 이름에 데이터 세트의 이름을 지정합니다.
9. 데이터셋 스키마의 경우 새 스키마 생성을 선택합니다. 스키마 정의 섹션에는 최소한의 항목 상호 작용 스키마가 표시됩니다. 스키마는 이전에 ratings.csv 파일에 추가한 헤더와 일치하므로 아무 것도 변경할 필요가 없습니다. 학습 데이터를 생성하지 않은 경우 [시작하기 전제 조건](#) 단원을 참조하세요.
10. 스키마 이름의 경우 새 스키마의 이름을 지정합니다. 다음과 유사한 화면이 표시되어야 합니다.

[Amazon Personalize](#) > [Dataset groups](#) > [custom-dataset-group-name](#) > [Import item interactions data](#)

Step 1
[Choose import method](#)

Step 2
Configure item interactions schema

Step 3
Configure item interactions dataset import job

Configure item interactions schema [Info](#)

Dataset details

Dataset name

The name you enter here can help you distinguish this dataset import job from others.

The dataset name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ (hyphen).

Dataset schema

The schema you provide allows Amazon Personalize to understand and import your data.

- Create new schema
 Use an existing schema

Schema name

The name you enter here can help you distinguish this schema from others.

The schema name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ (hyphen).

Schema definition

Verify your data structure matches the following schema.

```

1  {
2    "type": "record",
3    "name": "Interactions",
4    "namespace": "com.amazon.personalize.schema",
5    "fields": [
6      {
7        "name": "USER_ID",
8        "type": "string"
9      },
10     {
11       "name": "ITEM_ID",
12       "type": "string"
13     },
14     {
15       "name": "TIMESTAMP",
16       "type": "long"
17     }
18   ],
19   "version": "1.0"
20 }

```

JSON Line 1, Column 2 0 Errors: 0 0 Warnings: 0

► Tags - optional (0) [Info](#)

A tag is a label that you assign to an resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your costs.

[Cancel](#)

[Previous](#)

[Next](#)

11. 다음을 선택합니다. 항목 상호 작용 데이터 세트 가져오기 작업 구성 페이지가 나타납니다. 그 다음, [2단계: 항목 상호 작용 데이터 가져오기](#)을 완료하여 상호작용 데이터를 가져옵니다.

2단계: 항목 상호 작용 데이터 가져오기

데이터 세트를 만들었으니 이제 항목 상호 작용 데이터를 데이터 세트로 가져올 차례입니다.

항목 상호 작용 데이터를 가져오려면

1. 항목 상호 작용 데이터 세트 가져오기 작업 구성 페이지에서 데이터 가져오기 소스에 대해 S3에서 데이터 가져오기를 선택합니다.
2. 데이터세트 가져오기 작업 이름의 경우, 가져오기 작업의 이름을 지정합니다.
3. 추가 S3 버킷 정책 필요 대화 상자에서 Personalize 권한을 부여하지 않은 경우 지침에 따라 [필수 S3 버킷 정책을 추가](#)합니다.
4. 데이터 위치의 경우 Simple Storage Service(S3)에서 영화 데이터 파일을 저장할 위치를 지정합니다. 다음 구문을 사용합니다.

s3://<name of your S3 bucket>/<folder path>/filename.csv

5. IAM 역할 섹션에서 IAM 서비스 역할에 대해 사용자 지정 IAM 역할 ARN 입력을 선택합니다.
6. 사용자 지정 IAM 역할 ARN의 경우 [Personalize에 대한 IAM 역할 생성](#)에서 생성한 역할을 지정합니다.

데이터세트 가져오기 작업 세부 정보 및 IAM 역할 섹션은 다음과 비슷해야 합니다.

Configure item interactions dataset import job [Info](#)

Dataset import job details

Data import source

Import data from S3
Specify the location where your data is stored in S3.

Incrementally import data with APIs
Incrementally import item interactions data with the event ingestion SDK.

Dataset import job name

The name you enter here can help you distinguish this dataset import job from others.

gs-dataset-import-job

The dataset import job name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Data import source

Additional S3 bucket policy required
In addition to the IAM service role defined above, Amazon Personalize also requires you to add a bucket policy to the S3 bucket containing your data files so that it can process them. Follow the instructions [described here](#) to add the required bucket policy to your S3 bucket.

Data location [Info](#)

Choose the S3 location of your data.

s3://bucket-name/ratings.csv

Your file needs to be in a CSV format and reflect the schema.

IAM Role

IAM service role

Amazon Personalize requires permissions to access your S3 bucket. Choose an existing role with access or create a role in the IAM console with the [AmazonPersonalizeFullAccess](#) IAM policy attached.

Enter a custom IAM role ARN

Custom IAM role ARN

arn: :iam::accountNumber:role/roleName

After you import data from S3, you can still incrementally import data with the Amazon Personalize console, the AWS Command Line Interface (CLI), or the SDKs.

Publish event metrics to S3 - optional

When you create a metric attribution, reports related to this import job can be published to S3 for analysis with your tool of choice.

To track and publish metrics for events, you must create a metric attribution and define event metrics.

[Create metric attribution](#)

Publish metrics from this import job (you have not created a metric attribution)

- 이벤트 지표를 S3와 Tags에 게시 섹션을 변경하지 않고 그대로 두고 가져오기 시작을 선택합니다. 데이터 가져오기 작업이 시작되고 개요 페이지가 표시됩니다. 처음에는 상태가 생성 대기 중이고(그 다음은 생성 진행 중), 솔루션 생성 버튼이 비활성화되어 있습니다.

데이터 가져오기 작업을 마치면 상태가 활성으로 변경되고 솔루션 생성 버튼이 활성화됩니다.

- 데이터를 가져왔으니 이제 [3단계: 솔루션 생성](#)에서 솔루션을 만들 준비가 되었습니다.

3단계: 솔루션 생성

이 자습서에서는 [2단계: 항목 상호 작용 데이터 가져오기](#) 가져온 데이터세트를 사용하여 모델을 학습 시킵니다. 학습된 모델을 솔루션 버전이라고 합니다.

Important

이 자습서에서는 자동 학습을 사용하는 솔루션을 만듭니다. 자동 교육을 사용하면 솔루션이 활성화되는 동안 교육 비용이 발생합니다. 불필요한 비용을 피하려면 완료 후 솔루션을 삭제해야 합니다. 자세한 정보는 [리소스 정리](#)를 참조하세요.

솔루션을 생성하려면

- 데이터세트 그룹의 개요 페이지, 3단계에서 교육 및 추천 리소스를 설정하고 솔루션 생성을 선택합니다.
- 솔루션 이름의 경우 솔루션에 대한 이름을 지정합니다.
- 솔루션 유형에서는 항목 추천을 선택합니다.
- 레시피에서 선택하세요aws-user-personalization-v2.

다음과 유사한 화면이 표시되어야 합니다.

Specify solution details

Choose your solution type and choose the recipe to use in training.

Solution details

Solution name
The solution name that you enter here can help you distinguish this solution from others.

The solution name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Solution type [Info](#)
Choose the type of solution you want to create. The type determines what recipes are available for solution creation.

Item recommendation
Create a solution that generates item recommendations.
Supports Generative AI

Action recommendation - new
Create a solution that predicts the actions your users will most likely take.

You must create an Actions dataset to create an Action recommendation solution.

User segmentation
Create a solution that predicts groups of users based on item input data.

Recipe
Recipes are preconfigured algorithms tailored to specific use cases.

Recommends items a user will interact with based on their preferences. This recipe u...

Info **Analyze data before training** Analyze data [↗](#)

Before training, make sure you analyze your data. Data analysis includes item count statistics as well as actions you can take to meet training requirements and improve recommendations.

▶ Tags - optional (0) [Info](#)

A tag is a label that you assign to an resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your costs.

Cancel
Next

5. 다음을 선택합니다. 교육 구성 필드는 변경하지 않고 그대로 두십시오. 생성한 솔루션은 7일마다 새 모델을 자동으로 학습시키고 가장 최근의 항목 상호 작용 데이터에 더 많은 가중치를 부여합니다.
6. 다음을 선택하고 솔루션에 대한 세부 정보를 검토합니다.
7. 솔루션 생성을 선택하면 솔루션의 세부정보 페이지가 표시됩니다. 솔루션을 생성한 후 Amazon Personalize는 한 시간 내에 첫 번째 솔루션 버전 생성을 시작합니다. 교육이 시작되면 세부 정보 페이지의 솔루션 버전 섹션에 표시되며 교육 상태를 모니터링할 수 있습니다.

솔루션 버전 상태가 활성화되면 [4단계: 캠페인 생성](#)으로 이동할 준비가 된 것입니다.

4단계: 캠페인 생성

이 절차에서는 이전 단계에서 생성한 솔루션 버전을 배포하는 캠페인을 생성합니다.

캠페인을 생성하려는 경우

1. 탐색 창에서 사용자 지정 리소스를 확장하고 캠페인을 선택합니다.
2. 캠페인 생성을 선택합니다. 새 캠페인 생성 페이지가 나타납니다.
3. 캠페인 세부 정보의 캠페인 이름에서 캠페인의 이름을 지정합니다.
4. 솔루션의 경우 이전 단계에서 만든 솔루션을 선택합니다.
5. 최신 솔루션 버전 자동 사용을 선택합니다. 다른 모든 필드는 변경하지 마십시오.

다음과 유사한 화면이 표시되어야 합니다.

Create new campaign

Campaign details

Campaign name
The text you enter here appears in the Campaign dashboard and detail page. It can help you distinguish this campaign from others.

The campaign name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Solution
Choose the solution the campaign uses to generate recommendations.

getting-started-solution

Recipe: user-personalization-v2 Automatic training: On

Last updated: May 24, 2024, 19:24 (UTC-7:00)

Automatically use the latest solution version | [Info](#)

Automatically use the latest solution version - *new*
Choose this option to have the campaign automatically use the latest active solution version. If you don't, you must manually update the campaign each time you want to deploy a new solution version.

Minimum provisioned transactions per second | [Info](#)
The minimum amount of throughput in transactions per second (TPS) that is provisioned for this campaign.

Enter a number from 1-500.

i If you create an Items dataset, you can configure your campaign to include metadata in recommendations. [Create items dataset](#) ↗

▶ **Tags - optional (0)** [Info](#)

A tag is a label that you assign to an resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your costs.

Cancel Create campaign

6. 캠페인 생성을 선택합니다. 캠페인 생성이 시작되고 개인 맞춤 API 섹션이 있는 캠페인 세부 정보 페이지가 표시됩니다.

캠페인을 만드는 데 몇 분 정도 걸릴 수 있습니다. Personalize에서 캠페인 생성을 마치면 페이지가 업데이트되어 캠페인 결과 테스트 섹션이 표시됩니다. 다음과 유사한 화면이 표시되어야 합니다.

getting-started-campaign

[Delete](#)
[Update](#)

Personalization API
Details

Campaign inference

To get recommendations for this campaign in your application, use the `getRecommendations` API call. You can learn more about the usage and requirements for this API call in the documentation and the other links listed below.

- [Amazon Personalize GetRecommendations Developer Guide](#)

Campaign ARN [Info](#)
arn:aws:personalize:us-west-2:123456789012:campaign/getting-started-campaign

Test campaign results

Before you use your campaign in your application, you can view the recommendations it generates here.

User ID [Info](#)
This is the user ID of the user you want to see campaign results for. This user ID needs to be obtained from your item-interactions or user dataset.

Filter name - optional
Choose an existing filter to apply to your user segments or create a new filter.

None
▼

↻
View

[Create new filter](#)

▶ **Promotion - optional** [Info](#)
Define additional business rules to promote a subset of items in recommendations. The promotion filter you specify applies to these items instead of any filter you specify above.

▶ **Additional metadata - optional** [Info](#)
Choose the item metadata you want to include in recommendations.

Get recommendations

5단계: 추천 받기

이 절차에서는 이전 단계에서 생성한 캠페인을 사용하여 추천을 받습니다.

추천을 받으려면

1. 캠페인 결과 테스트의 사용자 ID에서 평점 데이터세트의 값(예: **83**)을 지정합니다. 다른 모든 필드는 변경하지 않고 그대로 두십시오.
2. 추천 받기를 선택합니다. 추천 패널에는 추천 항목에 대한 항목 ID와 점수가 나열됩니다.

다음과 유사한 화면이 표시되어야 합니다.

getting-started-campaign

[Delete](#)
[Update](#)



[Personalization API](#)
[Details](#)

Campaign inference

To get recommendations for this campaign in your application, use the `getRecommendations` API call. You can learn more about the usage and requirements for this API call in the documentation and the other links listed below.

- [Amazon Personalize GetRecommendations Developer Guide](#)

Campaign ARN [Info](#)

arn:aws:personalize:us-west-2:123456789012:campaign/getting-started-campaign

Test campaign results

Before you use your campaign in your application, you can view the recommendations it generates here.

User ID [Info](#)

This is the user ID of the user you want to see campaign results for. This user ID needs to be obtained from your item-interactions or user dataset.

Filter name - optional

Choose an existing filter to apply to your user segments or create a new filter.

 [View](#)

[Create new filter](#)

Promotion - optional [Info](#)

Define additional business rules to promote a subset of items in recommendations. The promotion filter you specify applies to these items instead of any filter you specify above.

Additional metadata - optional [Info](#)

Choose the item metadata you want to include in recommendations.

[Get recommendations](#)

Recommendations (25) [Info](#)

Recommendations include up to 25 items or actions. Any promoted items are distributed randomly. For some custom recipes, the reason column lists if the item is included through exploration or as a popular item placeholder.

Recommendation ID

[Copy](#) RID-35-402c-ba0b-5fb86a83f81a-CID-4598c7

Item ID	Score	Recommendation reason
318	0.0032059	-
589	0.0027560	-
593	0.0023468	-
356	0.0022429	-
527	0.0022052	-
296	0.0021353	-
457	0.0020748	-
50	0.0020146	-
150	0.0019477	-
1	0.0019159	-
780	0.0019149	-
480	0.0018700	-
592	0.0018420	-
1221	0.0018401	-
110	0.0017960	-
590	0.0017893	-
1198	0.0016929	-
47	0.0016593	-
시작하기(콘솔)	0.0016585	-
364	0.0015704	-
500	0.0015517	-
7361	0.0015233	-

시작하기(AWS CLI)

이 연습에서는 AWS Command Line Interface (AWS CLI) 를 사용하여 Amazon Personalize를 살펴봅니다. 지정한 사용자 ID에 대해 추천 영화를 반환하는 캠페인을 생성합니다.

이 연습을 시작하기 전에 다음 작업을 수행해야 합니다.

- [시작하기 전제 조건](#) 시작하기 검토.
- 예 [설정: AWS CLI](#) 지정된 대로 AWS CLI를 설정합니다.

시작하기 연습을 마쳤을 때 불필요한 요금이 발생하지 않도록 하려면 [리소스 정리](#)의 단계에 따라 생성한 리소스를 삭제합니다.

Note

이 연습의 AWS CLI 명령은 Linux에서 테스트되었습니다. Windows에서의 AWS CLI 명령 사용에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 [AWS Command Line Interface에 대한 파라미터 값 지정](#) 단원을 참조하세요.

1단계: 학습 데이터 가져오기

아래 단계를 수행해 데이터세트 그룹을 생성하고, 이 그룹에 데이터세트를 추가한 다음 영화 평점 데이터를 사용하여 이 데이터세트를 채웁니다.

1. 다음 명령을 실행하여 데이터세트 그룹을 생성합니다. [AWS Key Management Service](#) 키 ARN 및 해당 키에 대한 액세스 권한이 있는 IAM 역할의 ARN을 입력 파라미터로 전달하면 데이터세트 그룹을 암호화할 수 있습니다. API에 대한 자세한 내용은 [CreateDatasetGroup](#) 단원을 참조하세요.

```
aws personalize create-dataset-group --name MovieRatingDatasetGroup --kms-key-arn arn:aws:kms:us-west-2:01234567890:key/1682a1e7-a94d-4d92-bbdf-837d3b62315e --role-arn arn:aws:iam::01234567890:KMS-key-access
```

데이터세트 그룹 ARN이 예를 들어, 다음과 같이 표시됩니다.

```
{
  "datasetGroupArn": "arn:aws:personalize:us-west-2:acct-id:dataset-group/MovieRatingDatasetGroup"
}
```


`describe-dataset-group` 명령을 사용하여 생성한 데이터세트 그룹을 표시하고, 반환되는 데이터세트 그룹 ARN을 지정합니다.

```
aws personalize describe-dataset-group \
--dataset-group-arn arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieRatingDatasetGroup
```

데이터세트 그룹과 해당 속성이 예를 들어, 다음과 같이 표시됩니다.

```
{
  "datasetGroup": {
    "name": "MovieRatingDatasetGroup",
    "datasetGroupArn": "arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieRatingDatasetGroup",
    "status": "ACTIVE",
    "creationDateTime": 1542392161.262,
    "lastUpdatedDateTime": 1542396513.377
  }
}
```

Note

그룹에서 데이터세트를 생성하려면 데이터세트 그룹의 `status`가 `ACTIVE`로 표시될 때까지 기다립니다. 이 작업은 일반적으로 빠르게 진행됩니다.

데이터세트 그룹 ARN이 기억나지 않을 경우, `list-dataset-groups` 명령을 사용하면 생성한 모든 데이터세트 그룹이 ARN과 함께 표시됩니다.

```
aws personalize list-dataset-groups
```

Note

`describe-object` 및 `list-objects` 명령은 대부분의 Personalize 객체에 사용할 수 있습니다. 이러한 명령은 이 연습의 나머지 부분에는 표시되지 않지만 사용할 수 있습니다.

- 다음 코드를 MovieRatingSchema.json 파일에 저장해 스키마 파일을 JSON 형식으로 생성합니다. 이 스키마는 이전에 ratings.csv에 추가한 헤더와 일치합니다. 스키마 이름은 Interactions로, Amazon Personalize에서 인식하는 데이터 세트 유형 중 하나와 일치합니다. 자세한 정보는 [스키마](#)을 참조하세요.

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "TIMESTAMP",
      "type": "long"
    }
  ],
  "version": "1.0"
}
```

- 다음 명령을 실행하여 스키마를 생성합니다. 이전 단계에서 저장한 파일을 지정합니다. 이 예제에서는 이 파일을 현재 폴더에 속한 것으로 표시합니다. API에 대한 자세한 내용은 [CreateSchema](#) 단원을 참조하세요.

```
aws personalize create-schema \
  --name MovieRatingSchema \
  --schema file://MovieRatingSchema.json
```

스키마의 리소스 이름(ARN)이 예를 들어, 다음과 같이 표시됩니다.

```
{
  "schemaArn": "arn:aws:personalize:us-west-2:acct-id:schema/MovieRatingSchema"
}
```

4. 다음 명령을 실행하여 빈 데이터셋을 생성합니다. 이전 단계에서 반환된 데이터셋 그룹 ARN 및 스키마 ARN을 입력합니다. `dataset-type`은 이전 단계의 스키마 `name`와 일치해야 합니다. API에 대한 자세한 내용은 [CreateDataset](#) 단원을 참조하세요.

```
aws personalize create-dataset \
  --name MovieRatingDataset \
  --dataset-group-arn arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieRatingDatasetGroup \
  --dataset-type Interactions \
  --schema-arn arn:aws:personalize:us-west-2:acct-id:schema/MovieRatingSchema
```

데이터셋 ARN이 예를 들어, 다음과 같이 표시됩니다.

```
{
  "datasetArn": "arn:aws:personalize:us-west-2:acct-id:dataset/
MovieRatingDatasetGroup/INTERACTIONS"
}
```

5. 데이터셋에 학습 데이터를 추가합니다.
 - a. 다음 명령을 실행하여 데이터셋 가져오기 작업을 생성합니다. 이전 단계에서 반환된 데이터셋 ARN 및 S3 버킷 이름을 입력합니다. 에서 생성한 AWS Identity and Access Management (IAM) 역할 ARN을 입력합니다. [Personalize에 대한 IAM 역할 생성](#) API에 대한 자세한 내용은 [CreateDatasetImportJob](#) 단원을 참조하세요.

```
aws personalize create-dataset-import-job \
  --job-name MovieRatingImportJob \
  --dataset-arn arn:aws:personalize:us-west-2:acct-id:dataset/
MovieRatingDatasetGroup/INTERACTIONS \
  --data-source dataLocation=s3://bucketname/ratings.csv \
  --role-arn roleArn
```

데이터셋 가져오기 작업 ARN이 예를 들어, 다음과 같이 표시됩니다.

```
{
  "datasetImportJobArn": "arn:aws:personalize:us-west-2:acct-id:dataset-import-
job/MovieRatingImportJob"
}
```

- b. `describe-dataset-import-job` 명령을 사용하여 상태를 확인합니다. 이전 단계에서 반환된 데이터세트 가져오기 작업 ARN을 제공합니다. API에 대한 자세한 내용은 [DescribeDatasetImportJob](#) 단원을 참조하세요.

```
aws personalize describe-dataset-import-job \
  --dataset-import-job-arn arn:aws:personalize:us-west-2:acct-id:dataset-
  import-job/MovieRatingImportJob
```

상태를 포함하여 데이터세트 가져오기 작업의 속성이 표시됩니다. 처음에 `status`는 `CREATE PENDING`으로 표시됩니다. 예를 들면 다음과 같이 표시됩니다.

```
{
  "datasetImportJob": {
    "jobName": "MovieRatingImportJob",
    "datasetImportJobArn": "arn:aws:personalize:us-west-2:acct-id:dataset-
  import-job/MovieRatingImportJob",
    "datasetArn": "arn:aws:personalize:us-west-2:acct-id:dataset/
  MovieRatingDatasetGroup/INTERACTIONS",
    "dataSource": {
      "dataLocation": "s3://<bucketname>/ratings.csv"
    },
    "roleArn": "role-arn",
    "status": "CREATE PENDING",
    "creationDateTime": 1542392161.837,
    "lastUpdatedDateTime": 1542393013.377
  }
}
```

상태가 `ACTIVE`로 표시되면 데이터세트 가져오기가 완료된 것이고, 그러면 지정된 데이터세트를 사용하여 모델을 학습시킬 준비가 된 것입니다.

Note

가져오기에는 시간이 필요합니다. 데이터세트를 사용하여 모델을 학습시키려면 먼저 데이터세트 가져오기가 완료될 때까지 기다려야 합니다.

2단계: 솔루션 생성(모델 학습)

모델을 학습시키려면 [CreateSolution](#) 작업을 사용하여 모델을 학습시키기 위한 구성을 생성하고 자동 학습은 그대로 두십시오. 솔루션은 한 시간 내에 첫 번째 솔루션 학습을 자동으로 시작합니다.

레시피와 학습 데이터를 사용하여 모델을 학습시킵니다. Personalize는 미리 정의된 레시피 세트를 제공합니다. 자세한 정보는 [레시피 선택](#)을 참조하세요. 이 연습에서는 사용자 개인화-v2 레시피를 사용합니다.

1. 다음 명령을 실행하여 모델을 학습시키기 위한 구성을 생성합니다. 이 명령은 자동 학습을 사용하는 솔루션을 생성합니다. 7일마다 새 솔루션 버전이 자동으로 생성됩니다 (기본값).

```
aws personalize create-solution \
  --name MovieSolution \
  --dataset-group-arn arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieRatingDatasetGroup \
  --recipe-arn arn:aws:personalize:::recipe/aws-user-personalization-v2 \
  --perform-auto-training \
  --solution-config "{\"autoTrainingConfig\": {\"schedulingExpression\": \"rate(7
days)\"}\"}
```

솔루션 ARN이 예를 들어, 다음과 같이 표시됩니다.

```
{
  "solutionArn": "arn:aws:personalize:us-west-2:acct-id:solution/MovieSolution"
}
```

2. `describe-solution` 명령을 사용하여 생성 상태를 확인합니다. 이전 단계에서 반환된 솔루션 ARN을 입력합니다. API에 대한 자세한 내용은 [DescribeSolution](#) 단원을 참조하세요.

```
aws personalize describe-solution \
  --solution-arn arn:aws:personalize:us-west-2:acct-id:solution/MovieSolution
```

솔루션의 속성과 생성 status가 표시됩니다. 예:

```
{
  "solution": {
    "name": "MovieSolution",
    "solutionArn": "arn:aws:personalize:us-west-2:acct-id:solution/
MovieSolution",
    "performHPO": false,
```

```

    "performAutoML": false,
    "recipeArn": "arn:aws:personalize::recipe/aws-user-personalization-v2",
    "datasetGroupArn": "arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieRatingDatasetGroup",
    "solutionConfig": {
      "algorithmHyperParameters": {
        "apply_recency_bias": "true"
      },
      "featureTransformationParameters": {},
      "autoTrainingConfig": {
        "schedulingExpression": "rate(7 days)"
      }
    },
    "status": "ACTIVE",
    "creationDateTime": "2021-05-12T16:27:59.819000-07:00",
    "lastUpdatedDateTime": "2021-05-12T16:27:59.819000-07:00"
  }
}

```

3. 자동 교육을 사용하면 솔루션이 활성화된 후 1년 이내에 솔루션 버전 교육이 시작됩니다. 교육이 시작된 후 다음 [ListSolutionVersions](#) 명령을 사용하여 솔루션 버전의 Amazon 리소스 이름 (ARN) 을 가져올 수 있습니다.

```
aws personalize list-solution-versions --solution-arn arn:aws:personalize:us-west-2:acct-id:solution/MovieSolution
```

4. `describe-solution-version` 명령을 사용하여 솔루션 버전의 교육 상태를 확인합니다. 이전 단계에서 반환된 솔루션 버전 ARN을 입력합니다. API에 대한 자세한 내용은 [DescribeSolutionVersion](#) 단원을 참조하세요.

```
aws personalize describe-solution-version \
  --solution-version-arn arn:aws:personalize:us-west-2:acct-id:solution/
MovieSolution/version-id
```

솔루션 버전의 속성과 학습 status가 표시됩니다. 처음에 상태는 CREATE PENDING으로 표시 됩니다. 예를 들면 다음과 같이 표시됩니다.

```

{
  "solutionVersion": {
    "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/
MovieSolution/<version-id>",
    ...
  }
}

```

```

    "status": "CREATE PENDING"
  }
}

```

5. 최신 솔루션 버전 status가 ACTIVE로 표시되면 학습이 완료된 것입니다.

이제 학습 지표를 검토하고 솔루션 버전을 사용하여 캠페인을 만들 수 있습니다.

Note

학습에는 시간이 필요합니다. 캠페인에서 이 솔루션 버전을 사용하려면 학습이 완료될 때까지 기다려야 합니다(솔루션 버전의 학습 상태가 ACTIVE로 표시됨).

6. 솔루션의 지표를 살펴보면 솔루션 버전의 성능을 확인할 수 있습니다. 다음 명령을 실행하여 솔루션 버전에 대한 지표를 가져옵니다. 이전에 반환된 솔루션 버전 ARN을 제공합니다. API에 대한 자세한 내용은 [GetSolutionMetrics](#) 단원을 참조하세요.

```

aws personalize get-solution-metrics \
  --solution-version-arn arn:aws:personalize:us-west-2:acct-id:solution/
  MovieSolution/version-id

```

샘플 응답이 다음과 같이 표시됩니다.

```

{
  "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/www-
  solution/<version-id>",
  "metrics": {
    "coverage": 0.0485,
    "mean_reciprocal_rank_at_25": 0.0381,
    "normalized_discounted_cumulative_gain_at_10": 0.0363,
    "normalized_discounted_cumulative_gain_at_25": 0.0984,
    "normalized_discounted_cumulative_gain_at_5": 0.0175,
    "precision_at_10": 0.0107,
    "precision_at_25": 0.0207,
    "precision_at_5": 0.0107
  }
}

```

3단계: 캠페인 생성(솔루션 배포)

추천을 받으려면 먼저 솔루션 버전을 배포해야 합니다. 솔루션 배포를 캠페인 생성이라고도 합니다. 캠페인을 생성하면 클라이언트 애플리케이션이 [GetRecommendationsAPI](#)를 사용하여 추천을 가져올 수 있습니다.

1. 다음 명령을 실행하여 캠페인을 생성합니다. 이전 단계에서 반환된 솔루션 버전 ARN을 입력합니다. API에 대한 자세한 내용은 [CreateCampaign](#) 단원을 참조하세요.

```
aws personalize create-campaign \
  --name MovieRecommendationCampaign \
  --solution-version-arn arn:aws:personalize:us-west-2:acct-id:solution/
MovieSolution/version-id \
  --min-provisioned-tps 1
```

샘플 응답이 다음과 같이 표시됩니다.

```
{
  "campaignArn": "arn:aws:personalize:us-west-2:acct-id:campaign/
MovieRecommendationCampaign"
}
```

2. 다음 명령을 실행하여 배포 상태를 확인합니다. 이전 단계에서 반환된 캠페인 ARN을 입력합니다. API에 대한 자세한 내용은 [DescribeCampaign](#) 단원을 참조하세요.

```
aws personalize describe-campaign \
  --campaign-arn arn:aws:personalize:us-west-2:acct-id:campaign/
MovieRecommendationCampaign
```

샘플 응답이 다음과 같이 표시됩니다.

```
{
  "campaign": {
    "name": "MovieRecommendationCampaign",
    "campaignArn": "arn:aws:personalize:us-west-2:acct-id:campaign/
MovieRecommendationCampaign",
    "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/
MovieSolution/<version-id>",
    "minProvisionedTPS": "1",
    "creationDateTime": 1543864775.923,
    "lastUpdatedDateTime": 1543864791.923,
```



```

    "status": "CREATE_IN_PROGRESS"
  }
}

```

Note

캠페인에서 추천 항목을 받으려면 status가 ACTIVE로 표시될 때까지 기다립니다.

4단계: 추천 받기

get-recommendations 명령을 실행하여 추천을 받습니다. 이전 단계에서 반환된 캠페인 ARN을 입력합니다. 요청 시 영화 평점 데이터세트의 사용자 ID를 지정합니다. API에 대한 자세한 내용은 [GetRecommendations](#) 단원을 참조하세요.

Note

일부 레시피는 GetRecommendationsAPI를 지원하지 않습니다. 자세한 정보는 [레시피 선택](#)을 참조하세요.

이 단계에서 호출하는 AWS CLI 명령은 이전 단계와 다릅니다. personalize-runtime

```

aws personalize-runtime get-recommendations \
  --campaign-arn arn:aws:personalize:us-west-2:acct-id:campaign/
  MovieRecommendationCampaign \
  --user-id 123

```

응답에서 캠페인은 사용자가 좋아할 수 있는 항목 추천 목록(영화 ID)을 반환합니다. 이 목록은 사용자와의 관련성을 기준으로 내림차순으로 정렬됩니다.

```

{
  "itemList": [
    {
      "itemId": "14"
    },
    {
      "itemId": "15"
    },
    {

```

```

        "itemId": "275"
    },
    {
        "itemId": "283"
    },
    {
        "itemId": "273"
    },
    ...
]
}

```

시작하기(Python용 SDK(Boto3))

이 자습서에서는 Python용 SDK(Boto3)를 사용해 처음부터 끝까지 Personalize 워크플로우를 완료하는 방법을 보여줍니다.

이 시작하기 연습을 완료한 후 불필요한 비용이 발생하지 않도록 하려면 이 자습서에서 생성한 리소스를 삭제합니다. 자세한 내용은 [리소스 정리](#) 단원을 참조하세요.

주제

- [사전 조건](#)
- [튜토리얼](#)
- [주피터\(IPython\) 노트북으로 Personalize API 사용 시작하기](#)

사전 조건

다음은 이 안내서에서 Python 예제를 사용하기 위한 필수 조건 단계입니다.

- [시작하기 전제 조건](#)를 완료하여 필요한 권한을 설정하고 학습 데이터를 생성합니다. 고유한 소스 데이터를 사용하는 경우 필수 조건 단계와 같이 데이터의 형식이 지정되었는지 확인합니다.
- 예 지정된 대로 AWS SDK for Python (Boto3) 환경을 설정합니다. [AWS SDK 설정](#).

튜토리얼

다음에 이어지는 단계에서는 환경을 확인하고 Personalize용 Python용 SDK(Boto3) 클라이언트를 생성합니다. 그런 다음 데이터를 가져와서 캠페인이 포함된 솔루션 버전을 생성 및 배포하고 추천을 받습니다.

1단계: Python 환경을 확인하고 boto3 클라이언트 만들기

필수 조건을 완료한 후 다음 Python 예제를 실행하여 환경이 올바르게 구성되었는지 확인합니다. 또한 이 코드는 이 자습서에서 사용하는 Personalize boto3 클라이언트를 생성합니다. 환경이 올바르게 구성된 경우 사용 가능한 레시피 목록이 표시되고 이 자습서의 다른 Python 예제를 실행할 수 있습니다.

```
import boto3

personalizeRt = boto3.client('personalize-runtime')
personalize = boto3.client('personalize')

response = personalize.list_recipes()

for recipe in response['recipes']:
    print (recipe)
```

2단계: 데이터 가져오기

Personalize boto3 클라이언트를 생성하고 환경을 확인했으면, [시작하기 전제 조건](#) 완료 시 생성한 과거 데이터를 가져옵니다. 과거 데이터를 Personalize로 가져오려면 다음과 같이 합니다.

1. 다음 코드를 사용하여 Personalize에서 스키마를 생성합니다. `getting-started-schema`을 스키마의 이름으로 바꿉니다.

```
import json
schema = {
    "type": "record",
    "name": "Interactions",
    "namespace": "com.amazonaws.personalize.schema",
    "fields": [
        {
            "name": "USER_ID",
            "type": "string"
        },
        {
            "name": "ITEM_ID",
            "type": "string"
        },
        {
            "name": "TIMESTAMP",
            "type": "long"
        }
    ]
}
```

```

    ],
    "version": "1.0"
  }

  create_interactions_schema_response = personalize.create_schema(
      name='getting-started-schema',
      schema=json.dumps(schema)
  )

  interactions_schema_arn = create_interactions_schema_response['schemaArn']
  print(json.dumps(create_interactions_schema_response, indent=2))

```

2. 다음 코드를 사용하여 데이터세트 그룹을 생성합니다. `dataset group name`을 데이터세트 그룹의 이름으로 바꿉니다.

```

response = personalize.create_dataset_group(name = 'dataset group name')
dataset_group_arn = response['datasetGroupArn']

description = personalize.describe_dataset_group(datasetGroupArn = dataset_group_arn)
['datasetGroup']

print('Name: ' + description['name'])
print('ARN: ' + description['datasetGroupArn'])
print('Status: ' + description['status'])

```

3. 다음 코드를 사용하여 새 데이터 세트 그룹에 항목 상호 작용 데이터 세트를 만듭니다. 데이터세트에 이름을 지정하고 이전 단계의 `schema_arn` 및 `dataset_group_arn`를 입력합니다.

```

response = personalize.create_dataset(
    name = 'datase_name',
    schemaArn = 'schema_arn',
    datasetGroupArn = 'dataset_group_arn',
    datasetType = 'Interactions'
)

dataset_arn = response['datasetArn']

```

4. 다음 코드를 사용한 데이터세트 가져오기 작업으로 데이터를 가져옵니다. 코드는 `describe_dataset_import_job` 메서드를 사용하여 작업 상태를 추적합니다.

작업 이름, 이전 단계의 `dataset_arn`, 학습 데이터를 저장한 S3 버킷 경로(`s3://bucket name/folder name/ratings.csv`), IAM 서비스 역할의 ARN 등을 파라미터로 전달합니다. [시작](#)

[하기 전제 조건](#)의 일부로 이 역할을 생성했습니다. 버킷에 액세스할 수 있는 권한이 Personalize에 필요합니다. [Personalize에 S3 리소스에 대한 액세스 권한 부여](#) 단원을 참조하세요.

```
import time
response = personalize.create_dataset_import_job(
    jobName = 'JobName',
    datasetArn = 'dataset_arn',
    dataSource = {'dataLocation': 's3://bucket/file.csv'},
    roleArn = 'role_arn',
    importMode = 'FULL'
)

dataset_interactions_import_job_arn = response['datasetImportJobArn']

description = personalize.describe_dataset_import_job(
    datasetImportJobArn = dataset_interactions_import_job_arn)['datasetImportJob']

print('Name: ' + description['jobName'])
print('ARN: ' + description['datasetImportJobArn'])
print('Status: ' + description['status'])

max_time = time.time() + 3*60*60 # 3 hours
while time.time() < max_time:
    describe_dataset_import_job_response = personalize.describe_dataset_import_job(
        datasetImportJobArn = dataset_interactions_import_job_arn
    )
    status = describe_dataset_import_job_response["datasetImportJob"]['status']
    print("Interactions DatasetImportJob: {}".format(status))

    if status == "ACTIVE" or status == "CREATE FAILED":
        break

    time.sleep(60)
```

3단계: 솔루션 생성

데이터를 가져온 후 다음과 같이 솔루션과 솔루션 버전을 생성합니다. 솔루션에는 모델을 학습하기 위한 구성이 포함되어 있으며 솔루션 버전은 학습된 모델입니다.

1. 다음 코드를 사용하여 새 솔루션을 생성합니다. 이전의 매개변수, 솔루션 이름, 사용자 개인화-v2 레시피의 ARN () 을 매개변수로 전달합니다. dataset_group_arn

arn:aws:personalize:::recipe/aws-user-personalization-v2 새 솔루션의 ARN을 저장해 두었다가 나중에 사용할 수 있습니다.

```
create_solution_response = personalize.create_solution(
    name='solution name',
    recipeArn= 'arn:aws:personalize:::recipe/aws-user-personalization-v2',
    datasetGroupArn = 'dataset group arn'
)
solution_arn = create_solution_response['solutionArn']
print('solution_arn: ', solution_arn)
```

- 다음 코드를 사용하여 솔루션 버전을 생성합니다. 이전 단계의 solution_arn을 파라미터로 전달합니다. 다음 코드는 솔루션 버전을 만듭니다. 학습 중에 코드는 [DescribeSolutionVersion](#) 작업을 사용하여 솔루션 버전의 상태를 검색합니다. 학습이 완료되면 메서드는 새 솔루션 버전의 ARN을 반환합니다. 나중에 사용하기 위해 이 ARN을 저장합니다.

```
import time
import json

create_solution_version_response = personalize.create_solution_version(
    solutionArn = solution_arn
)

solution_version_arn = create_solution_version_response['solutionVersionArn']
print(json.dumps(create_solution_version_response, indent=2))

max_time = time.time() + 3*60*60 # 3 hours
while time.time() < max_time:
    describe_solution_version_response = personalize.describe_solution_version(
        solutionVersionArn = solution_version_arn
    )
    status = describe_solution_version_response["solutionVersion"]["status"]
    print("SolutionVersion: {}".format(status))

    if status == "ACTIVE" or status == "CREATE FAILED":
        break

    time.sleep(60)
```

4단계: 캠페인 생성

솔루션 버전을 생성한 후 Personalize 캠페인을 통해 배포합니다. 다음 코드를 사용하여 솔루션 버전을 배포하는 캠페인을 생성합니다. 파라미터로 캠페인 이름, `solution_version_arn`을 전달합니다. 메서드는 새 캠페인의 리소스 이름(ARN)을 반환합니다. 나중에 사용하기 위해 이 ARN을 저장합니다.

```
response = personalize.create_campaign(
    name = 'campaign name',
    solutionVersionArn = 'solution version arn'
)

arn = response['campaignArn']

description = personalize.describe_campaign(campaignArn = arn)['campaign']
print('Name: ' + description['name'])
print('ARN: ' + description['campaignArn'])
print('Status: ' + description['status'])
```

5단계: 추천 받기

캠페인을 생성한 후 이 캠페인을 사용하여 추천을 받을 수 있습니다. 다음 코드는 캠페인에서 추천을 받고 각 추천 항목의 ID를 인쇄하는 방법을 보여줍니다. 이전 단계에서 생성한 캠페인의 ARN을 전달합니다. 사용자 ID의 경우 학습 데이터에서 가져온 사용자 ID(예123:)를 전달합니다.

```
response = personalizeRt.get_recommendations(
    campaignArn = 'Campaign ARN',
    userId = '123',
    numResults = 10
)

print("Recommended items")
for item in response['itemList']:
    print (item['itemId'])
```

주피터(IPython) 노트북으로 Personalize API 사용 시작하기

주피터 노트북을 사용하여 Personalize를 사용하려면 [Personalize 샘플](#) 리포지토리의 [getting_start](#) 폴더에 있는 일련의 노트북을 복제하거나 다운로드합니다. 노트북에서는 Personalize를 사용하여 학습 데이터를 가져오고, 솔루션을 생성하고, 캠페인을 생성하고, 추천을 받는 과정을 안내합니다.

Note

노트북을 시작하기 전에 [README.md](#)의 단계에 따라 환경을 구축해야 합니다

시작하기(Java 2.x용 SDK)

이 자습서에서는 AWS SDK for Java 2.x에서 Personalize 워크플로우를 처음부터 끝까지 완료하는 방법을 보여줍니다.

시작하기 연습을 마쳤을 때 불필요한 요금이 발생하지 않도록 하려면 [리소스 정리](#)의 단계에 따라 자습서에서 생성한 리소스를 삭제합니다.

더 많은 예제는 [Personalize 프로젝트 완료](#) 단원을 참조하세요.

주제

- [사전 조건](#)
- [Personalize 프로젝트 완료](#)

사전 조건

이 자습서를 완료하기 위한 필수 조건 단계는 다음과 같습니다.

- [시작하기 전제 조건](#)를 완료하여 필요한 권한을 설정하고 학습 데이터를 생성합니다. [시작하기\(콘솔\)](#) 또는 [시작하기\(AWS CLI\)](#) 연습에 사용된 것과 동일한 소스 데이터를 사용할 수 있습니다. 고유한 소스 데이터를 사용하는 경우 필수 조건 단계와 같이 데이터의 형식이 지정되었는지 확인합니다.
- 개발자 안내서의 설정 프로시저에 지정된 대로 Java 2.x용 SDK 환경 AWS 및 자격 증명을 [설정합니다](#). AWS SDK for Java 2.x

튜토리얼

다음에 이어지는 단계에서는 Personalize 패키지를 사용하도록 프로젝트를 설정하고 Personalize Java 2.x용 SDK 클라이언트를 생성합니다. 그런 다음 데이터를 가져와서 캠페인이 포함된 솔루션 버전을 생성 및 배포하고, 추천을 받습니다.

1단계: Personalize 패키지를 사용하도록 프로젝트 설정

필수 조건을 완료한 후, Personalize 종속 항목을 pom.xml 파일에 추가하고 Personalize 패키지를 가져옵니다.

1. pom.xml 파일에 다음 종속 항목을 추가합니다. 최신 버전 번호가 예제 코드와 다를 수 있습니다.

```
<dependency>
  <groupId>software.amazon.awssdk</groupId>
  <artifactId>personalize</artifactId>
  <version>2.16.83</version>
</dependency>
<dependency>
  <groupId>software.amazon.awssdk</groupId>
  <artifactId>personalizeruntime</artifactId>
  <version>2.16.83</version>
</dependency>
<dependency>
  <groupId>software.amazon.awssdk</groupId>
  <artifactId>personalizeevents</artifactId>
  <version>2.16.83</version>
</dependency>
```

2. 프로젝트에 다음 가져오기 명령문을 추가합니다.

```
// import client packages
import software.amazon.awssdk.services.personalize.PersonalizeClient;
import software.amazon.awssdk.services.personalizeruntime.PersonalizeRuntimeClient;
// Amazon Personalize exception package
import software.amazon.awssdk.services.personalize.model.PersonalizeException;
// schema packages
import software.amazon.awssdk.services.personalize.model.CreateSchemaRequest;
// dataset group packages
import software.amazon.awssdk.services.personalize.model.CreateDatasetGroupRequest;
import software.amazon.awssdk.services.personalize.model.DescribeDatasetGroupRequest;
// dataset packages
import software.amazon.awssdk.services.personalize.model.CreateDatasetRequest;
// dataset import job packages
import
  software.amazon.awssdk.services.personalize.model.CreateDatasetImportJobRequest;
import software.amazon.awssdk.services.personalize.model.DataSource;
import software.amazon.awssdk.services.personalize.model.DatasetImportJob;
import
  software.amazon.awssdk.services.personalize.model.DescribeDatasetImportJobRequest;
// solution packages
import software.amazon.awssdk.services.personalize.model.CreateSolutionRequest;
import software.amazon.awssdk.services.personalize.model.CreateSolutionResponse;
// solution version packages
```

```
import software.amazon.awssdk.services.personalize.model.DescribeSolutionRequest;
import
    software.amazon.awssdk.services.personalize.model.CreateSolutionVersionRequest;
import
    software.amazon.awssdk.services.personalize.model.CreateSolutionVersionResponse;
import
    software.amazon.awssdk.services.personalize.model.DescribeSolutionVersionRequest;
// campaign packages
import software.amazon.awssdk.services.personalize.model.CreateCampaignRequest;
import software.amazon.awssdk.services.personalize.model.CreateCampaignResponse;
// get recommendations packages
import
    software.amazon.awssdk.services.personalizeruntime.model.GetRecommendationsRequest;
import
    software.amazon.awssdk.services.personalizeruntime.model.GetRecommendationsResponse;
import software.amazon.awssdk.services.personalizeruntime.model.PredictedItem;
// Java time utility package
import java.time.Instant;
```

2단계: Personalize 클라이언트 생성

Personalize 종속 항목을 pom.xml 파일에 추가하고 필요한 패키지를 가져온 후 다음과 같은 Personalize 클라이언트를 생성합니다.

```
PersonalizeClient personalizeClient = PersonalizeClient.builder()
    .region(region)
    .build();

PersonalizeRuntimeClient personalizeRuntimeClient = PersonalizeRuntimeClient.builder()
    .region(region)
    .build();
```

3단계: 데이터 가져오기

Personalize 클라이언트를 초기화한 후에는 [시작하기 전제 조건](#) 완료 시 생성한 과거 데이터를 가져옵니다. 과거 데이터를 Personalize로 가져오려면 다음과 같이 합니다.

1. 다음 Avro 스키마를 작업 디렉토리에 JSON 파일로 저장합니다. 이 스키마는 [시작하기 전제 조건](#)을 완료할 때 만든 CSV 파일의 열과 일치합니다.

```
{
```

```

"type": "record",
"name": "Interactions",
"namespace": "com.amazonaws.personalize.schema",
"fields": [
  {
    "name": "USER_ID",
    "type": "string"
  },
  {
    "name": "ITEM_ID",
    "type": "string"
  },
  {
    "name": "TIMESTAMP",
    "type": "long"
  }
],
"version": "1.0"
}

```

2. 다음 `createSchema` 메서드를 사용하여 Personalize에서 스키마를 생성합니다. Personalize 서비스 클라이언트, 스키마 이름, 이전 단계에서 생성한 스키마 JSON 파일의 파일 경로를 파라미터로 전달합니다. 메서드가 새 스키마의 리소스 이름(ARN)을 반환합니다. 나중에 사용하기 위해 이 ARN을 저장합니다.

```

public static String createSchema(PersonalizeClient personalizeClient, String
schemaName, String filePath) {

    String schema = null;
    try {
        schema = new String(Files.readAllBytes(Paths.get(filePath)));
    } catch (IOException e) {
        System.out.println(e.getMessage());
    }

    try {
        CreateSchemaRequest createSchemaRequest = CreateSchemaRequest.builder()
            .name(schemaName)
            .schema(schema)
            .build();

        String schemaArn =
personalizeClient.createSchema(createSchemaRequest).schemaArn();

```

```

        System.out.println("Schema arn: " + schemaArn);

        return schemaArn;

    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}

```

3. 데이터세트 그룹을 생성합니다. 다음 `createDatasetGroup` 메서드를 사용하여 데이터세트 그룹을 생성합니다. Personalize 서비스 클라이언트 및 데이터세트 그룹 이름을 파라미터로 전달합니다. 메서드가 새 데이터세트 그룹의 ARN을 반환합니다. 나중에 사용하기 위해 이 ARN을 저장합니다.

```

public static String createDatasetGroup(PersonalizeClient personalizeClient,
String datasetGroupName) {

    try {
        CreateDatasetGroupRequest createDatasetGroupRequest =
CreateDatasetGroupRequest.builder()
            .name(datasetGroupName)
            .build();

        return
personalizeClient.createDatasetGroup(createDatasetGroupRequest).datasetGroupArn();
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}

```

4. 항목 상호 작용 데이터 세트를 생성합니다. 다음 `createDataset` 메서드를 사용하여 항목 상호 작용 데이터 세트를 생성합니다. Personalize 서비스 클라이언트, 데이터세트 이름, 스키마의 ARN, 데이터세트 그룹의 ARN, 데이터세트 유형에 대한 Interactions을 파라미터로 전달합니다. 메서드가 새 데이터세트의 ARN을 반환합니다. 나중에 사용하기 위해 이 ARN을 저장합니다.

```

public static String createDataset(PersonalizeClient personalizeClient,
String datasetName,
String datasetGroupArn,
String datasetType,
String schemaArn) {

```

```

try {
    CreateDatasetRequest request = CreateDatasetRequest.builder()
        .name(datasetName)
        .datasetGroupArn(datasetGroupArn)
        .datasetType(datasetType)
        .schemaArn(schemaArn)
        .build();

    String datasetArn = personalizeClient.createDataset(request)
        .datasetArn();
    System.out.println("Dataset " + datasetName + " created.");
    return datasetArn;

} catch (PersonalizeException e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
return "";
}

```

5. 데이터세트 가져오기 작업을 사용하여 데이터를 가져옵니다. 다음

`createPersonalizeDatasetImportJob` 메서드를 사용하여 데이터세트 가져오기 작업을 생성합니다.

Amazon Personalize 서비스 클라이언트, 작업 이름, 항목 상호 작용 데이터 세트의 ARN, 교육 데이터를 저장한 Amazon S3 버킷 경로 (`s3://bucket name/folder name/ratings.csv`), 서비스 역할의 ARN (이 역할을 일부로 생성함) 을 파라미터로 전달합니다. [시작하기 전제 조건](#) 메서드가 데이터세트 가져오기 작업의 ARN을 반환합니다. 나중에 사용할 수 있도록 저장할 수도 있습니다.

```

public static String createPersonalizeDatasetImportJob(PersonalizeClient
personalizeClient,
    String jobName,
    String datasetArn,
    String s3BucketPath,
    String roleArn) {

    long waitInMilliseconds = 60 * 1000;
    String status;
    String datasetImportJobArn;

    try {
        DataSource importDataSource = DataSource.builder()
            .dataLocation(s3BucketPath)

```

```
        .build());

        CreateDatasetImportJobRequest createDatasetImportJobRequest =
CreateDatasetImportJobRequest.builder()
        .datasetArn(datasetArn)
        .dataSource(importDataSource)
        .jobName(jobName)
        .roleArn(roleArn)
        .build();

        datasetImportJobArn =
personalizeClient.createDatasetImportJob(createDatasetImportJobRequest)
        .datasetImportJobArn();

        DescribeDatasetImportJobRequest describeDatasetImportJobRequest =
DescribeDatasetImportJobRequest.builder()
        .datasetImportJobArn(datasetImportJobArn)
        .build();

        long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

        while (Instant.now().getEpochSecond() < maxTime) {

            DatasetImportJob datasetImportJob = personalizeClient
                .describeDatasetImportJob(describeDatasetImportJobRequest)
                .datasetImportJob();

            status = datasetImportJob.status();
            System.out.println("Dataset import job status: " + status);

            if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
                break;
            }
            try {
                Thread.sleep(waitInMilliseconds);
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
        return datasetImportJobArn;

    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}
```

```
}

```

4단계: 솔루션 생성

데이터를 가져온 후 다음과 같이 솔루션과 솔루션 버전을 생성합니다. 솔루션에는 모델을 학습하기 위한 구성이 포함되어 있으며 솔루션 버전은 학습된 모델입니다.

1. 다음 `createPersonalizeSolution` 메서드를 사용하여 새 솔루션을 생성합니다. Amazon Personalize 서비스 클라이언트, 데이터세트 그룹 Amazon 리소스 이름 (ARN), 솔루션 이름, 사용자 개인화-v2 레시피용 ARN () 을 파라미터로 전달합니다. `arn:aws:personalize:::recipe/aws-user-personalization-v2` 메서드가 새 솔루션의 ARN을 반환합니다. 나중에 사용하기 위해 이 ARN을 저장합니다.

```
public static String createPersonalizeSolution(PersonalizeClient
personalizeClient,
        String datasetGroupArn,
        String solutionName,
        String recipeArn) {

    try {
        CreateSolutionRequest solutionRequest = CreateSolutionRequest.builder()
            .name(solutionName)
            .datasetGroupArn(datasetGroupArn)
            .recipeArn(recipeArn)
            .build();

        CreateSolutionResponse solutionResponse =
personalizeClient.createSolution(solutionRequest);
        return solutionResponse.solutionArn();

    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}
```

2. 다음 `createPersonalizeSolutionVersion` 메서드를 사용하여 솔루션 버전을 생성합니다. 이 전 단계의 솔루션 ARN을 파라미터로 전달합니다. 다음 코드는 먼저 솔루션이 준비되었는지 확인한 다음 솔루션 버전을 생성합니다. 학습 중에 코드는 [DescribeSolutionVersion](#) 작업을 사용하여 솔루션

버전의 상태를 검색합니다. 학습이 완료되면 메서드는 새 솔루션 버전의 ARN을 반환합니다. 나중에 사용하기 위해 이 ARN을 저장합니다.

```
public static String createPersonalizeSolutionVersion(PersonalizeClient
personalizeClient, String solutionArn) {
    long maxTime = 0;
    long waitInMilliseconds = 30 * 1000; // 30 seconds
    String solutionStatus = "";
    String solutionVersionStatus = "";
    String solutionVersionArn = "";

    try {
        DescribeSolutionRequest describeSolutionRequest =
DescribeSolutionRequest.builder()
            .solutionArn(solutionArn)
            .build();

        maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

        // Wait until solution is active.
        while (Instant.now().getEpochSecond() < maxTime) {

            solutionStatus =
personalizeClient.describeSolution(describeSolutionRequest).solution().status();
            System.out.println("Solution status: " + solutionStatus);

            if (solutionStatus.equals("ACTIVE") || solutionStatus.equals("CREATE
FAILED")) {
                break;
            }
            try {
                Thread.sleep(waitInMilliseconds);
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }

        if (solutionStatus.equals("ACTIVE")) {

            CreateSolutionVersionRequest createSolutionVersionRequest =
CreateSolutionVersionRequest.builder()
                .solutionArn(solutionArn)
                .build();
```



```
        CreateSolutionVersionResponse createSolutionVersionResponse =
personalizeClient
            .createSolutionVersion(createSolutionVersionRequest);
        solutionVersionArn =
createSolutionVersionResponse.solutionVersionArn();

        System.out.println("Solution version ARN: " + solutionVersionArn);

        DescribeSolutionVersionRequest describeSolutionVersionRequest =
DescribeSolutionVersionRequest.builder()
            .solutionVersionArn(solutionVersionArn)
            .build();

        while (Instant.now().getEpochSecond() < maxTime) {

            solutionVersionStatus =
personalizeClient.describeSolutionVersion(describeSolutionVersionRequest)
                .solutionVersion().status();
            System.out.println("Solution version status: " +
solutionVersionStatus);

            if (solutionVersionStatus.equals("ACTIVE") ||
solutionVersionStatus.equals("CREATE FAILED")) {
                break;
            }
            try {
                Thread.sleep(waitInMilliseconds);
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
        return solutionVersionArn;
    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}
```

자세한 내용은 [솔루션 및 솔루션 버전 생성](#) 단원을 참조하세요. 솔루션 버전 생성 시 계속하기 전에 성능을 평가할 수 있습니다. 자세한 내용은 [지표를 사용하여 솔루션 버전 평가](#) 단원을 참조하세요.

5단계: 캠페인 생성

솔루션 버전을 학습시키고 평가한 후 Personalize 캠페인을 사용하여 배포할 수 있습니다. 솔루션 버전을 배포하려면 다음 `createPersonalCampaign` 메서드를 사용합니다. Personalize 서비스 클라이언트, 이전 단계에서 생성한 솔루션 버전의 리소스 이름(ARN), 캠페인 이름을 파라미터로 전달합니다. 메서드가 새 캠페인의 ARN을 반환합니다. 나중에 사용하기 위해 이 ARN을 저장합니다.

```
public static String createPersonalCampaign(PersonalizeClient personalizeClient, String
    solutionVersionArn, String name) {

    try {
        CreateCampaignRequest createCampaignRequest = CreateCampaignRequest.builder()
            .minProvisionedTPS(1)
            .solutionVersionArn(solutionVersionArn)
            .name(name)
            .build();

        CreateCampaignResponse campaignResponse =
            personalizeClient.createCampaign(createCampaignRequest);
        System.out.println("The campaign ARN is "+campaignResponse.campaignArn());
        return campaignResponse.campaignArn();

    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

Personalize 캠페인에 대한 자세한 내용은 [캠페인 생성](#) 단원을 참조하세요.

6단계: 추천 받기

캠페인을 생성한 후 이 캠페인을 사용하여 추천을 받습니다. 사용자에게 추천을 받으려면 다음 `getRecs` 메서드를 사용합니다. Personalize 런타임 클라이언트, 이전 단계에서 생성한 캠페인의 리소스 이름(ARN), 가져온 과거 데이터의 사용자 ID(예:123)를 파라미터로 전달합니다. 메서드가 추천 항목 목록을 화면에 인쇄합니다.

```
public static void getRecs(PersonalizeRuntimeClient personalizeRuntimeClient,
    String campaignArn, String userId) {
```

```
try {
    GetRecommendationsRequest recommendationsRequest =
    GetRecommendationsRequest.builder()
        .campaignArn(campaignArn)
        .numResults(20)
        .userId(userId)
        .build();

    GetRecommendationsResponse recommendationsResponse =
personalizeRuntimeClient
        .getRecommendations(recommendationsRequest);
    List<PredictedItem> items = recommendationsResponse.itemList();
    for (PredictedItem item : items) {
        System.out.println("Item Id is : " + item.itemId());
        System.out.println("Item score is : " + item.score());
    }

} catch (AwsServiceException e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
}
```

Personalize 프로젝트 완료

[Java 2.x용 SDK를 사용하여 Amazon Personalize 워크플로를 완료하는 방법을 보여주는 all-in-one 프로젝트에 대해서는 Amazon-Personalize-Java-앱을 참조하십시오.](#) GitHub 이 프로젝트에는 레시피가 서로 다른 여러 솔루션 버전을 교육하고 작업과 함께 이벤트를 기록하는 작업이 포함됩니다. PutEvents

추가 예제는 AWS SDK 예제 저장소의 [personalize](#) 폴더에 있는 코드를 참조하십시오.

리소스 정리

불필요한 요금이 발생하지 않도록 하려면 시작하기 연습을 마쳤을 때에 생성한 리소스를 삭제합니다. Amazon Personalize 콘솔을 사용하여 리소스를 삭제하려면 리소스의 세부 정보 페이지에서 삭제를 선택합니다. Amazon Personalize API로 리소스를 삭제하려면 SDK 또는 () 와 함께 Delete API를 사용합니다. AWS Command Line Interface AWS CLI

Amazon Personalize 콘솔 AWS Command Line Interface AWS CLI() AWS 또는 SDK를 사용하여 데이터 세트를 삭제하는 방법에 대한 자세한 단계는 을 참조하십시오. [데이터세트를 삭제하여 해당 데이터를 모두 삭제합니다.](#) 이 단계의 패턴을 다른 Amazon Personalize 리소스에 적용할 수 있습니다.

일부 리소스는 다른 리소스보다 먼저 삭제해야 합니다. 다음 섹션에서는 리소스 삭제 지침을 제공하고 도메인 기반 및 사용자 지정 리소스를 삭제하는 순서를 나열합니다.

주제

- [리소스 정리 가이드라인](#)
- [도메인 기반 리소스 정리](#)
- [사용자 지정 리소스 정리](#)

리소스 정리 가이드라인

다음은 도메인 기반 리소스와 사용자 지정 리소스 모두에 적용됩니다.

- 상태가 CREATE PENDING 또는 IN PROGRESS인 리소스는 삭제할 수 없습니다. 리소스 상태가 ACTIVE 또는 CREATE FAILED여야 합니다. API (예: Describe API 작업) 를 사용하여 상태를 확인합니다. [DescribeCampaign](#)
- Amazon S3에 업로드한 교육 데이터를 삭제하려면 S3 [버킷에서 객체를 삭제하려면 어떻게 해야 합니까?](#) 를 참조하십시오. ratings.csv .
- 데이터세트를 삭제하기 전에 데이터세트에 대한 모든 데이터세트 가져오기 작업이 완료되었는지 확인하세요.
- 데이터세트 가져오기 작업이 완료된 후에는 요금이 부과되지 않으며 삭제할 수도 없습니다.
- Amazon Personalize 콘솔에서는 스키마를 삭제할 수 없으며 스키마 저장 비용도 청구되지 않습니다. 스키마를 삭제하려면 [DeleteSchema](#) API 작업을 사용하십시오.

도메인 기반 리소스 정리

도메인 데이터세트 그룹을 생성한 경우 다음 순서대로 리소스를 삭제하세요.

1. 추천자 — API로 추천자를 삭제하려면 API 작업을 사용하세요. [DeleteRecommender](#) 콘솔에서 추천자를 삭제하려면 추천자 페이지에서 추천자를 선택한 다음 오른쪽 상단에서 삭제를 선택합니다.
2. 데이터세트 — API에서 항목 상호작용 데이터세트를 삭제하려면 API 작업을 사용하세요. [DeleteDataset](#) 콘솔에서 데이터세트를 삭제하려면 데이터세트 페이지에서 데이터세트를 선택하여 세부정보 페이지를 확인하세요. 그런 다음 오른쪽 상단에서 삭제를 선택합니다.

Note

데이터세트를 삭제하려면 연결된 데이터세트 가져오기 작업이 만들기 보류 중 또는 진행 중일 수 없습니다. 또한 어떤 관련 추천인도 생성 보류 중 또는 진행 중 상태일 수 없습니다.

3. 데이터세트 그룹 - API가 포함된 데이터세트 그룹을 삭제하려면 API 작업을 사용하세요.

[DeleteDatasetGroup](#) 콘솔에서 데이터세트 그룹을 삭제하려면 데이터세트 그룹 페이지에서 데이터세트 그룹을 선택하고 오른쪽 상단에서 삭제를 선택합니다.

사용자 지정 리소스 정리

커스텀 데이터세트 그룹을 만든 경우 이 튜토리얼에서 만든 커스텀 리소스를 다음 순서대로 삭제하세요.

1. 캠페인 — API로 캠페인을 삭제하려면 [DeleteCampaign](#) API 작업을 사용하세요. 콘솔에서 캠페인을 삭제하려면 캠페인 페이지에서 캠페인을 선택한 다음 오른쪽 상단에서 삭제를 선택합니다.
2. 솔루션 — API로 솔루션을 삭제하려면 [DeleteSolution](#) API 작업을 사용하십시오. 콘솔에서 솔루션을 삭제하려면 솔루션 페이지에서 솔루션을 선택하여 해당 세부 정보 페이지를 확인하십시오. 그런 다음 오른쪽 상단에서 삭제를 선택합니다.

Note

솔루션을 삭제하면 관련 솔루션 버전이 모두 삭제됩니다. 어떤 솔루션 버전도 생성 보류 중 또는 진행 중 상태일 수 없습니다.

3. 데이터세트 - API가 포함된 항목 상호작용 데이터세트를 삭제하려면 API 작업을 사용하세요.

[DeleteDataset](#) 콘솔에서 데이터세트를 삭제하려면 데이터세트 페이지에서 데이터세트를 선택하여 세부정보 페이지를 확인하세요. 그런 다음 오른쪽 상단에서 삭제를 선택합니다.

Note

데이터세트를 삭제하려면 상태가 생성 보류 중 또는 진행 중일 DatasetImportJob 수 있는 관련자는 없어야 합니다. 또한 어떤 관련자도 생성 보류 중 또는 진행 중 상태일 SolutionVersion 수 없습니다.

4. 데이터세트 그룹 - API가 포함된 데이터세트 그룹을 삭제하려면 API 작업을 사용하세요.

[DeleteDatasetGroup](#) 콘솔에서 데이터세트 그룹을 삭제하려면 데이터세트 그룹 페이지에서 데이터 세트 그룹을 선택하고 오른쪽 상단에서 삭제를 선택합니다.

데이터세트 및 스키마

Personalize 데이터세트는 데이터를 위한 컨테이너입니다. 다음과 같은 5가지 유형의 데이터 세트가 있습니다.

- **항목 상호 작용** – 이 데이터 세트는 사용자와 항목 간 상호 작용의 이력 및 실시간 데이터를 저장합니다. Personalize에서 상호작용은 기록한 다음 교육 데이터로 가져오는 이벤트입니다. 도메인 데이터 세트 그룹과 사용자 지정 데이터 세트 그룹 모두에 대해 최소한의 항목 상호 작용 데이터 세트를 생성해야 합니다.
- **사용자** – 이 데이터세트는 사용자에 대한 메타데이터를 저장합니다. 여기에는 연령, 성별, 로열티 멤버십 또는 항목 제목과 같은 정보가 포함될 수 있습니다.
- **항목** – 이 데이터세트는 항목에 대한 메타데이터를 저장합니다. 여기에는 가격, SKU 유형 또는 재고 여부와 같은 정보가 포함될 수 있습니다.
- **작업** – 이 데이터 세트는 작업에 대한 메타데이터를 저장합니다. 작업은 고객에게 추천하고 싶은 참여 활동입니다. 작업에는 모바일 앱 설치, 멤버십 프로필 작성, 로열티 프로그램 가입, 프로모션 이메일 구독 등이 포함될 수 있습니다. 차선택-작업 레시피의 경우 작업 데이터 세트가 필요합니다. 다른 사용자 지정 레시피나 도메인 사용 사례에서는 작업 데이터를 사용하지 않습니다.
- **작업 상호 작용** – 이 데이터 세트는 사용자와 작업 간 상호 작용의 이력 및 실시간 데이터를 저장합니다. 차선택-작업 레시피는 이 데이터와 작업 데이터 세트의 데이터를 통해 사용자에게 작업을 추천합니다. 다른 사용자 지정 레시피나 도메인 사용 사례에서는 작업 상호 작용 데이터를 사용하지 않습니다.

각 데이터세트 그룹에는 한 가지 유형의 데이터세트만 있을 수 있습니다. 도메인 데이터세트 그룹에서는 액션 및 액션 인터랙션 데이터세트를 비롯한 차선택 액션 리소스를 만들 수 없습니다. Personalize는 사용자가 데이터세트를 삭제할 때까지 데이터세트에 데이터를 저장합니다. 모든 사용 사례(도메인 데이터세트 그룹) 및 레시피(사용자 지정 데이터세트 그룹)에 대해 상호작용 데이터에는 다음이 포함되어야 합니다.

- 카탈로그의 항목과 상호 작용하는 사용자의 1,000개 이상의 항목 상호 작용 기록. 이러한 상호작용은 대량 가져오기, 스트리밍된 이벤트 또는 둘 다에서 발생할 수 있습니다.
- 각각 2번 이상의 항목 상호 작용을 갖는 25개 이상의 고유 사용자 ID.

품질 추천의 경우, 각 2번 이상의 항목 상호 작용을 갖는 1,000명 이상의 사용자로부터 50,000건 이상의 항목 상호 작용을 갖는 것이 좋습니다.

데이터세트를 생성하기 전에 해당 데이터세트에 대한 스키마를 정의합니다. 스키마는 Personalize에 데이터 구조를 알려주고 Personalize가 데이터를 파싱할 수 있도록 해 줍니다. 스키마에는 값이 데이터 세트 유형과 일치해야 하는 이름 키가 있습니다. 스키마를 생성하고 나면 스키마를 변경할 수 없습니다.

도메인 데이터세트 그룹의 경우, 각 데이터세트 유형에는 필수 필드와 예약된 키워드가 포함된 기본 스키마가 있습니다. 데이터세트를 생성할 때마다 기존 도메인 스키마를 사용하거나 기존 기본 스키마를 수정하여 새 도메인 스키마를 만들 수 있습니다. 기본 스키마를 도메인에 가져올 데이터에 대한 설명서로 사용하세요. 스키마를 정의하고 데이터세트를 생성한 후에는 스키마를 변경할 수 없습니다.

데이터를 대량으로 가져오는 경우, 데이터를 CSV(쉼표로 구분된 값) 형식으로 저장해야 합니다. CSV 파일의 첫 번째 행에는 스키마와 일치해야 하는 열 헤더가 포함되어야만 합니다. Personalize의 대량 데이터 형식 지정 방법에 대한 자세한 내용은 [데이터 형식 지침](#) 단원을 참조하세요.

주제

- [데이터세트](#)
- [스키마](#)
- [데이터 형식 지침](#)

데이터세트

다음 주제는 Amazon Personalize 데이터 세트에 대한 상세 정보를 제공합니다. 각 데이터세트 유형에는 서로 다른 데이터 요구 사항이 있습니다. 두 도메인 데이터세트 그룹 및 사용자 지정 데이터세트 그룹에 대해 상호작용 데이터에는 다음이 포함되어야 합니다.

- 카탈로그의 항목과 상호 작용하는 사용자의 1,000개 이상의 항목 상호 작용 기록. 이러한 상호작용은 대량 가져오기, 스트리밍된 이벤트 또는 둘 다에서 발생할 수 있습니다.
- 각각 2번 이상의 항목 상호 작용을 갖는 25개 이상의 고유 사용자 ID.

품질 추천의 경우, 각 2번 이상의 항목 상호 작용을 갖는 1,000명 이상의 사용자로부터 50,000건 이상의 항목 상호 작용을 갖는 것이 좋습니다.

도메인 데이터세트 그룹을 생성하는 경우 각 데이터세트에는 도메인에 따라 추가 요구사항이 있습니다. 어떤 유형의 데이터가 필요한지 잘 모르겠으면, 도메인 데이터세트 그룹을 만들고 도메인의 기본 스키마를 설명서로 사용하는 것이 좋습니다. 데이터세트 및 스키마에 대한 자세한 내용은 [스키마](#) 단원을 참조하세요.

주제

- [항목 상호 작용 데이터 세트](#)
- [사용자 데이터 세트](#)
- [항목 데이터 세트](#)
- [작업 데이터 세트](#)
- [작업 상호 작용 데이터 세트](#)

항목 상호 작용 데이터 세트

항목 상호 작용은 카탈로그의 항목과 사용자 간의 긍정적인 상호 작용 이벤트입니다. 영화를 보거나, 목록을 보거나, 신발을 구매하는 사용자를 예로 들 수 있습니다. 항목과 사용자의 상호 작용에 대한 데이터를 항목 상호 작용 데이터 세트로 가져옵니다. 클릭, 시청 또는 좋아요 등과 같은 여러 이벤트 유형을 기록할 수 있습니다.

예를 들어, 사용자가 특정 항목을 클릭한 다음 해당 항목에 좋아요를 누르면 Personalize에서 이러한 이벤트를 교육 데이터로 사용할 수 있게 만들 수 있습니다. 각 이벤트에 대해 사용자 ID, 항목 ID, 타임스탬프(Unix 타임 에포크 형식) 및 이벤트 유형(클릭 및 좋아요)을 기록합니다. 그런 다음 두 항목 상호 작용 이벤트를 항목 상호 작용 데이터 세트에 추가합니다.

모든 사용 사례(도메인 데이터 세트 그룹) 및 레시피(사용자 지정 리소스)에 대해 항목 상호 작용 데이터에는 다음이 포함되어야 합니다.

- 카탈로그의 항목과 상호 작용하는 사용자의 1,000개 이상의 항목 상호 작용 기록. 이러한 상호 작용은 대량 가져오기, 스트리밍된 이벤트 또는 둘 다에서 발생할 수 있습니다.
- 각각 2번 이상의 항목 상호 작용을 갖는 25개 이상의 고유 사용자 ID.

품질 추천의 경우, 각 2번 이상의 항목 상호 작용을 갖는 1,000명 이상의 사용자로부터 50,000건 이상의 항목 상호 작용을 갖는 것이 좋습니다.

추천 또는 사용자 지정 솔루션을 만들려면 최소한 항목 상호 작용 데이터 세트를 만들어야 합니다. 이 섹션에서는 Amazon Personalize로 가져올 수 있는 다음과 같은 유형의 항목 상호 작용 데이터에 대한 정보를 제공합니다.

주제

- [이벤트 유형 및 이벤트 값 데이터](#)
- [컨텍스트 메타데이터](#)

• [노출 데이터](#)

이벤트 유형 및 이벤트 값 데이터

항목 상호작용 데이터세트는 각 상호작용에 대한 이벤트 유형 및 이벤트 값 데이터를 저장할 수 있습니다. 사용자 지정 리소스만 이벤트 값 데이터를 사용합니다.

이벤트 유형 데이터

Amazon Personalize는 클릭 또는 구매 데이터와 같은 이벤트 유형 데이터를 사용하여 사용자 의도와 관심을 식별합니다. 도메인 추천자를 생성하는 경우 모든 사용 사례에는 이벤트 유형 데이터가 필요합니다. 일부 사용 사례에는 특정 이벤트 유형이 필요합니다. 추가 이벤트 유형을 자유롭게 사용할 수 있습니다. 자세한 내용은 [사용 사례 선택](#) 단원을 참조하세요.

사용자 지정 리소스를 만드는 경우 교육에 사용할 이벤트를 이벤트 유형별로 선택할 수 있습니다. 데이터세트의 EVENT_TYPE 열에 여러 이벤트 유형이 있고 사용자 지정 솔루션을 구성할 때 이벤트 유형을 제공하지 않는 경우, Amazon Personalize는 유형에 관계없이 모든 항목 상호 작용 데이터를 동일한 가중치로 학습에 사용합니다. 자세한 정보는 [교육에 사용할 항목 상호 작용 데이터 선택](#)을 참조하세요.

포지티브 및 네거티브 이벤트 유형

Amazon Personalize는 모든 상호 작용이 긍정적인 것으로 가정합니다. 싫어요와 같은 부정적인 이벤트 유형과의 상호 작용이 반드시 사용자의 향후 추천에 해당 항목이 표시되지 않는 것은 아닙니다.

부정적인 사건과 사용자의 무관심한 부분이 권장 사항에 영향을 미치도록 하는 방법은 다음과 같습니다.

- 모든 도메인 사용 사례 및 [사용자-개인 맞춤](#) 레시피에 대해 Amazon Personalize는 노출 데이터를 사용할 수 있습니다. 노출 데이터에 항목이 나타나고 사용자가 선택하지 않으면 해당 항목이 추천에 나타날 가능성이 줄어듭니다. 자세한 정보는 [노출 데이터](#)을 참조하세요.
- 사용자 지정 리소스를 사용하고 긍정적 및 부정적 이벤트 유형을 가져오는 경우 긍정적인 이벤트 유형만 학습시킨 다음 사용자가 부정적으로 상호 작용한 항목을 필터링할 수 있습니다. 자세한 내용은 [교육에 사용할 항목 상호 작용 데이터 선택](#) 및 [추천 및 사용자 세그먼트 필터링](#) 섹션을 참조하세요.

이벤트 값 데이터 (사용자 지정 리소스)

이벤트 값 데이터는 사용자가 본 영화의 비율 또는 10점 만점의 평점일 수 있습니다. 사용자 지정 솔루션을 만들고 이벤트 유형 데이터와 함께 이벤트 값 데이터를 가져오는 경우 유형 및 값을 기반으로 교

육에 사용할 레코드를 선택할 수 있습니다. 도메인 추천자를 사용하면 Amazon Personalize는 이벤트 값 데이터를 사용하지 않으므로 교육 전에 이벤트를 필터링할 수 없습니다.

유형 및 값을 기준으로 레코드를 선택하려면 각 이벤트의 이벤트 유형과 이벤트 값을 기록하십시오. 각 이벤트에 대해 선택하는 값은 제외하려는 데이터와 기록하려는 이벤트 유형에 따라 달라집니다. 예를 들어, 시청 이벤트 유형별로 사용자가 시청한 동영상의 비율과 같은 사용자 활동을 일치시킬 수 있습니다.

솔루션을 구성할 때 특정 값을 임계값으로 설정하여 교육에서 레코드를 제외합니다. 예를 들어, 시청의 EVENT_TYPE이 포함된 이벤트의 EVENT_VALUE 데이터가 사용자가 시청한 동영상의 비율이고, 이벤트 값 임계값을 0.5로, 이벤트 유형을 시청으로 설정하면, Personalize는 EVENT_VALUE가 0.5 이상인 시청 상호작용 이벤트만 사용하여 모델을 교육합니다.

자세한 정보는 [교육에 사용할 항목 상호 작용 데이터 선택](#) 섹션을 참조하십시오.

컨텍스트 메타데이터

Personalize는 특정 레시피와 추천 사용 사례를 통해 사용자에게 가장 관련성이 높은 항목을 나타내는 기본 패턴을 식별할 때 컨텍스트 메타데이터를 사용할 수 있습니다. 컨텍스트 메타데이터는 이벤트 발생 시 사용자 환경에서 수집하는 상호작용 데이터(예: 위치 또는 디바이스 유형)입니다.

컨텍스트 메타데이터를 포함하면 기존 사용자에게 보다 개인화된 경험을 제공할 수 있습니다. 예를 들어, 고객이 휴대폰으로 카탈로그에 액세스할 때 컴퓨터와 다르게 쇼핑하는 경우, 사용자 기기에 대한 컨텍스트 메타데이터를 포함하세요. 그러면 고객의 검색 방식에 따라 관련성이 더 높은 추천이 제공될 수 있습니다.

또한 컨텍스트 메타데이터는 신규 사용자 또는 신원 미확인 사용자의 콜드 스타트 단계를 줄이는 데에도 도움이 됩니다. 콜드 스타트 단계는 권장 엔진이 해당 사용자에게 대한 이력 정보의 부족으로 인해 관련성이 낮은 추천을 제공하는 기간을 말합니다.

도메인 데이터세트 그룹의 경우 다음과 같은 추천자 사용 사례에서 컨텍스트 메타데이터를 사용할 수 있습니다.

- [추천 제품](#)(ECOMMERCE 도메인)
- [가장 적합한 추천 제품](#)(VIDEO_ON_DEMAND 도메인)

사용자 지정 리소스의 경우 컨텍스트 메타데이터를 사용하는 레시피에는 다음이 포함됩니다.

- [사용자 개인화 - v2](#) 및 [사용자-개인 맞춤](#)
- [맞춤형 랭킹-v2](#) 및 [개인 맞춤형-순위](#)

컨텍스트 정보에 대한 자세한 내용은 다음 AWS Machine Learning 블로그 게시물을 참조하십시오. 컨텍스트 정보를 활용하여 [Amazon Personalize 권장 사항의 관련성 높이기](#).

노출 데이터

노출은 사용자가 특정 항목과 상호작용(예: 클릭 또는 시청)했을 때 볼 수 있었던 항목의 목록입니다. 개인화 또는 [사용자-개인 맞춤](#) 레시피를 제공하는 도메인 사용 사례를 사용하는 경우 Amazon Personalize는 노출 데이터를 사용하여 탐색을 안내할 수 있습니다.

탐색을 사용하면 추천에 새 항목 또는 작업, 상호 작용이 거의 없는 항목 또는 작업, 이전 행동을 기준으로 사용자와 관련성이 낮은 항목 또는 작업 등 일반적으로 사용자에게 추천될 가능성이 낮은 일부 항목 또는 작업이 포함됩니다. 노출 데이터에서 항목이 더 자주 나타날수록 Personalize에서 탐색에 해당 항목을 포함할 가능성이 낮아집니다.

추천자 또는 솔루션을 생성할 때 Amazon Personalize는 항상 노출 데이터를 교육에서 제외합니다. Amazon Personalize는 노출 데이터를 사용하여 모델을 학습시키지 않기 때문입니다. 대신 권장 사항을 받을 때 이를 사용하여 사용자의 탐색을 안내합니다.

노출 값은 최대 1000자(세로 막대 문자 포함)를 포함할 수 있습니다. 도메인 데이터세트 그룹의 경우 다음과 같은 추천자 사용 사례에서 노출 데이터를 사용할 수 있습니다.

- [추천 제품](#)(ECOMMERCE 도메인)
- [가장 적합한 추천 제품](#)(VIDEO_ON_DEMAND 도메인)

탐색에 대한 자세한 내용은 [탐색](#) 단원을 참조하세요. Personalize는 다음과 같은 두 가지 유형의 노출: [암시적 노출](#) 및 [명시적 노출](#)를 모델링할 수 있습니다.

암시적 노출

암시적 노출은 사용자에게 보여주는 Personalize에서 검색한 추천입니다. ([GetRecommendations](#) 및 [GetPersonalizedRanking](#) 작업에서 반환한) RecommendationId를 향후 [PutEvents](#) 요청에 대한 입력으로 포함하여 추천 워크플로우에 통합할 수 있습니다. Personalize는 추천 데이터를 기반으로 암시적 노출을 도출합니다.

예를 들면, 동영상 스트리밍을 위한 추천을 제공하는 애플리케이션이 있을 수 있습니다. 암시적 노출을 사용하는 추천 워크플로우는 다음과 같을 수 있습니다.

1. Personalize [the section called "GetRecommendations"](#) API 작업을 사용하여 사용자 중 한 명에게 동영상 추천을 요청합니다.

2. Personalize는 모델(솔루션 버전)을 사용하여 사용자를 위한 추천을 생성하고 API 응답에서 `recommendationId`와 함께 이를 반환합니다.
3. 애플리케이션에서 사용자에게 동영상 추천을 보여줍니다.
4. 사용자가 동영상과 상호작용하는 경우(예: 클릭), [PutEvents](#) API 직접 호출에 선택 사항을 기록하고 `recommendationId`를 파라미터로 포함합니다. 코드 샘플에 대해서는 [노출 데이터 기록](#) 단원을 참조하세요.
5. Personalize는 `recommendationId`를 사용하여 이전 동영상 권고 사항에서 노출 데이터를 추출한 다음 노출 데이터를 사용하여 탐색을 안내합니다. 이 경우, 향후 추천에는 상호작용 데이터 또는 관련성이 적은 새 동영상이 포함됩니다.

암시적 노출 데이터를 사용한 이벤트 기록에 대한 자세한 내용은 [노출 데이터 기록](#) 단원을 참조하세요.

명시적 노출

명시적 노출은 수동으로 기록하여 Personalize에 전송하는 노출입니다. 명시적 노출을 사용하여 Personalize의 결과를 조작할 수 있습니다. 품목 순서는 영향을 받지 않습니다.

예를 들면, 신발을 위한 추천을 제공하는 쇼핑 애플리케이션이 있을 수 있습니다. 현재 재고가 있는 신발만 추천하는 경우, 명시적인 노출을 사용하여 해당 품목을 지정할 수 있습니다. 명시적 노출을 사용하는 추천 워크플로우는 다음과 같을 수 있습니다.

1. Personalize [the section called "GetRecommendations"](#) API를 사용하여 사용자 중 한 명에게 추천을 요청합니다.
2. Personalize는 모델(솔루션 버전)을 사용하여 사용자를 위한 추천을 생성하고 API 응답에서 이를 반환합니다.
3. 재고가 있는 권장 신발만 사용자에게 보여줍니다.
4. 실시간 증분 데이터 가져오기의 경우, 사용자가 신발 한 켤레와 상호작용(예: 클릭)할 때 [PutEvents](#) API 직접 호출에 선택 항목을 기록하고 `impression` 파라미터에 재고가 있는 권장 항목을 나열합니다. 코드 샘플에 대해서는 [노출 데이터 기록](#) 단원을 참조하세요.

이력 항목 상호 작용 데이터에서 노출을 가져오기 위해 csv 파일에 명시적 노출을 나열하고 각 항목을 '|' 문자로 구분할 수 있습니다. 세로 막대 문자는 1000자 한도에 포함됩니다. 관련 예제는 [명시적 노출의 형식 지정](#) 단원을 참조하세요.

5. Personalize는 노출 데이터를 사용하여 탐색을 안내하며, 향후 추천에는 상호작용 데이터 또는 관련성이 더 적은 새로운 신발이 포함됩니다.

사용자 데이터 세트

Personalize로 가져올 수 있는 사용자 데이터에는 성별 또는 로열티 멤버십과 같은 사용자에 대한 수치적 및 범주형 메타데이터가 포함됩니다. 사용자에 대한 메타데이터를 Personalize 사용자 데이터세트로 가져옵니다. 최대 메타데이터 열 수는 25개입니다.

이 주제는 다음 유형의 사용자 데이터에 대한 정보를 제공합니다.

주제

- [범주형 메타데이터](#)

범주형 메타데이터

일부 레시피와 VIDEO_ON_DEMAND 및 ECOMMERCE 도메인을 모두 사용하여 Personalize는 사용자에게 가장 관련성이 높은 항목을 나타내는 기본 패턴을 식별할 때 사용자의 성별 또는 멤버십 상태와 같은 범주형 메타데이터를 사용합니다. 사용 사례를 기반으로 자체 값 범위를 직접 정의합니다. 범주형 메타데이터는 어떤 언어로도 사용할 수 있습니다.

모든 레시피와 도메인으로 범주형 메타데이터를 가져와서 사용자의 속성에 따라 추천을 필터링하는데 사용할 수 있습니다. 추천 필터링에 대한 자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 단원을 참조하세요.

범주형 값은 최대 1000자까지 포함할 수 있습니다. 범주형 값이 1000자를 초과하는 사용자가 있는 경우, 데이터세트 가져오기 작업이 실패합니다.

사용자 지정 데이터세트 그룹 및 사용자 지정 솔루션의 경우, 범주형 메타데이터를 사용하는 레시피에는 다음이 포함됩니다.

- [사용자 개인화 - v2](#) 및 [사용자-개인 맞춤](#)
- [맞춤형 랭킹-v2](#) 및 [개인 맞춤형-순위](#)
- [유사-항목](#)

항목 데이터 세트

Personalize로 가져올 수 있는 항목 데이터에는 생성 타임스탬프, 가격, 장르, 설명 및 재고 여부와 같은 수치 및 범주형 메타데이터가 포함됩니다. 항목에 대한 메타데이터를 Personalize 항목 데이터세트로 가져옵니다.

Amazon Personalize는 교육 시 항목 제목 또는 작성자 데이터와 같은 비범주형 문자열 항목 데이터를 사용하지 않습니다. 하지만 일부 Amazon Personalize 기능은 이 데이터를 사용하여 권장 사항을 개선합니다. 자세한 정보는 [비범주형 문자열 데이터](#) 섹션을 참조하십시오.

최대 메타데이터 열 수는 100개입니다. Amazon Personalize가 교육 중에 고려하는 최대 항목 수는 사용 사례 또는 레시피에 따라 다릅니다. 교육 중에 고려한 항목만 권장 사항에 표시될 수 있습니다.

- 사용자 개인화-v2 또는 개인화 순위-v2의 경우 훈련 중에 모델이 고려하는 최대 항목 수는 5백만 개입니다. 이러한 항목은 항목 및 항목 상호 작용 데이터세트 모두에서 가져온 것입니다.
- 사용자 맞춤화-v2 및 개인화 순위-v2를 제외한 모든 도메인 사용 사례 및 사용자 지정 레시피의 경우, 학습 및 추천 생성 과정에서 모델이 고려하는 최대 항목 수는 75만 개입니다.

[레시피 선택](#) 레시피 요구 사항에 대한 자세한 내용은 을 참조하십시오.

이 주제는 다음 유형의 항목 데이터 데이터에 대한 정보를 제공합니다.

주제

- [생성 타임스탬프 데이터](#)
- [범주형 메타데이터](#)
- [비정형 텍스트 메타데이터](#)
- [비범주형 문자열 데이터](#)

생성 타임스탬프 데이터

Personalize는 생성 타임스탬프 데이터(Unix epoch 시간 형식, 초 단위)를 사용하여 항목의 수명을 계산하고 그에 따라 추천을 조정합니다.

하나 이상의 항목에 대한 생성 타임스탬프 데이터가 누락된 경우, Personalize는 상호작용 데이터(있는 경우)에서 이 정보를 유추하고 항목에서 가장 오래된 상호작용 데이터의 타임스탬프를 항목 생성 타임스탬프로 사용합니다. 항목에 상호작용 데이터가 없는 경우, 생성 타임스탬프는 교육 세트에 있는 최신 상호작용의 타임스탬프로 설정되며 Personalize는 이를 새 항목으로 간주합니다.

범주형 메타데이터

특정 레시피와 도메인을 통해 Personalize는 사용자에게 가장 관련성이 높은 항목을 나타내는 기본 패턴을 식별할 때 항목의 장르 또는 색상과 같은 범주형 메타데이터를 사용합니다. 사용 사례를 기반으로 자체 값 범위를 직접 정의합니다. 범주형 메타데이터는 어떤 언어로든 사용할 수 있습니다.

모든 레시피와 도메인을 통해 범주형 데이터를 가져와서 항목의 속성에 따라 추천을 필터링하는 데 사용할 수 있습니다. 추천 필터링에 대한 자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 단원을 참조하세요.

범주형 값은 최대 1000자까지 입력할 수 있습니다. 범주형 값이 1000자를 초과하는 항목이 있는 경우, 데이터세트 가져오기 작업이 실패합니다.

도메인 데이터세트 그룹의 경우, VIDEO_ON_DEMAND 도메인과 ECOMMERCE 도메인 모두 범주형 메타데이터를 사용합니다. 사용자 지정 데이터세트 그룹 및 사용자 지정 솔루션의 경우, 범주형 메타데이터를 사용하는 레시피에는 다음이 포함됩니다.

- [사용자 개인화 - v2](#) 및 [사용자-개인 맞춤](#)
- [맞춤형 랭킹-v2](#) 및 [개인 맞춤형-순위](#)
- [유사-항목](#)
- [항목-친화도](#)
- [항목-속성-친화도](#)

비정형 텍스트 메타데이터

특정 레시피와 도메인을 통해 Personalize는 제품 설명, 제품 리뷰 또는 영화 시놉시스와 같은 비정형 텍스트 메타데이터에서 중요한 정보를 추출할 수 있습니다. Personalize는 특히 새 항목이거나 상호작용 데이터가 적은 항목의 경우 구조화되지 않은 텍스트를 사용하여 사용자와 관련된 항목을 식별합니다. 아이템 데이터세트에 구조화되지 않은 텍스트 데이터를 포함하여 카탈로그의 새 항목에 대한 클릭률과 전환률을 높이세요.

비정형 데이터를 사용하려면 유형 string이 있는 필드를 항목 스키마에 추가하고 필드 textual속성을 true로 설정합니다. 최대 1개의 텍스트 필드를 추가할 수 있습니다. 그런 다음 대량 CSV 파일 및 개별 항목 가져오기에 텍스트 데이터를 포함합니다.

대량 CSV 파일의 경우 텍스트를 큰따옴표로 감싸고 새 줄 문자를 모두 제거합니다. \ 문자를 사용하여 데이터에 있는 큰따옴표 또는 \ 문자를 이스케이프처리합니다. 비정형 텍스트 데이터용 필드가 있는 항목 스키마의 예제는 [항목 데이터세트 스키마 예제\(사용자 지정\)](#) 단원을 참조하세요. Personalize는 텍스트 필드를 글자 수 제한으로 잘라냅니다. 텍스트에서 가장 관련성이 높은 정보가 필드 시작 부분에 있는지 확인하세요. Personalize로 데이터를 가져오는 방법에 대한 자세한 내용은 [단계 2: 데이터 준비 및 가져오기](#) 단원을 참조하세요.

구조화되지 않은 텍스트 값은 중국어와 일본어를 제외한 모든 언어에서 최대 20,000자를 포함할 수 있습니다. 중국어와 일본어의 경우, 최대 7,000자까지 입력할 수 있습니다. Personalize는 문자 제한을 초과하는 값을 문자 제한까지 잘라냅니다.

텍스트는 다음 언어로 제공될 수 있습니다.

- 중국어 간체
- 중국어 번체
- 영어
- 프랑스어
- 독일어
- 일본어
- 포르투갈어
- 스페인어

구조화되지 않은 텍스트 항목을 여러 언어로 제출할 수 있지만, 각 항목의 텍스트는 한 가지 언어로만 작성해야 합니다.

도메인 데이터세트 그룹의 경우, VIDEO_ON_DEMAND 도메인과 ECOMMERCE 도메인 모두 텍스트 메타데이터를 사용합니다. 사용자 지정 데이터세트 그룹 및 사용자 지정 솔루션의 경우, 텍스트 메타데이터를 사용하는 레시피에 다음이 포함됩니다.

- [사용자 개인화 -v2](#) 및 [사용자-개인 맞춤](#)
- [맞춤형 랭킹-v2](#) 및 [개인 맞춤형-순위](#)
- [유사-항목](#)
- [항목-친화도](#)
- [항목-속성-친화도](#)

비범주형 문자열 데이터

항목 ID를 제외하고 Amazon Personalize는 교육 시 항목 제목 또는 작성자 데이터와 같은 비범주형 문자열 데이터를 사용하지 않습니다. 하지만 Amazon Personalize는 이를 다음과 같은 기능과 함께 사용할 수 있습니다.

- Amazon Personalize는 추천에 비범주형 문자열 값을 비롯한 항목 메타데이터를 포함할 수 있습니다. 메타데이터를 통해 사용자 인터페이스에서 추천을 강화할 수 있습니다(예: 영화 추천 캐러셀에 감독 이름 추가). 자세한 정보는 [권장 사항의 항목 메타데이터](#)를 참조하세요.
- [유사-항목](#)를 사용하면 테마가 포함된 배치 추천을 생성할 수 있습니다. 테마가 포함된 배치 추천을 생성할 때는 배치 추론 작업에서 항목 이름 열을 지정해야 합니다. 자세한 정보는 [콘텐츠 생성기에서 테마가 있는 배치 추천](#)을 참조하세요.
- 필터를 생성하여 비범주형 문자열 데이터를 기반으로 항목을 추천에 포함하거나 추천에서 제거할 수 있습니다. 필터에 대한 자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 단원을 참조하세요.

작업 데이터 세트

작업은 사용자에게 추천하고 싶은 참여 또는 수익 창출 활동입니다. 작업에는 모바일 앱 설치, 멤버십 프로필 작성, 로열티 프로그램 가입, 프로모션 이메일 구독 등이 포함될 수 있습니다. 작업에 대한 데이터를 Amazon Personalize 작업 데이터 세트로 가져옵니다. 작업 데이터의 예로는 작업 ID, 작업의 예상 값, 작업의 만료 타임스탬프 등이 있습니다.

모델 교육 중에 Amazon Personalize는 최대 1,000개의 작업을 고려합니다. 작업을 1,000개가 넘게 가져올 경우 Amazon Personalize는 새 작업(상호 작용 없이 최근에 추가한 작업)과 최근 상호 작용 데이터가 있는 기존 작업에 우선순위를 두고 교육에 포함할 작업을 결정합니다.

Note

도메인 데이터세트 그룹에는 액션 및 액션 인터랙션 데이터세트를 비롯한 차선책 액션 리소스를 생성할 수 없습니다.

최대 열 수는 10개입니다. 이 주제는 다음 유형의 작업 데이터에 대한 정보를 제공합니다.

주제

- [작업 만료 타임스탬프 데이터](#)
- [반복 빈도 데이터](#)
- [값 데이터](#)
- [생성 타임스탬프 데이터](#)
- [범주형 메타데이터](#)

작업 만료 타임스탬프 데이터

작업 만료 타임스탬프는 작업이 더 이상 유효하지 않은 날짜를 지정합니다. 작업 만료 타임스탬프 데이터를 Unix epoch 시간 형식(초)으로 제공합니다. 작업이 만료된 경우 Amazon Personalize는 해당 작업을 추천에 포함하지 않습니다.

추천을 특정 시간 프레임에만 표시하도록 제한하려면 작업에 대해 작업 만료 타임스탬프를 지정하세요. 예를 들어, 특정 달까지 멤버십 모집을 실행하는 애플리케이션이 있을 수 있습니다. 등록 작업의 만료 타임스탬프를 해당 월말로 설정할 수 있습니다. Amazon Personalize는 해당 날짜에 다르면 이 작업 추천을 자동으로 중단합니다.

새 작업에 대한 만료 타임스탬프를 과거 시간으로 설정하거나 작업 타임스탬프를 과거 시간으로 업데이트하면 추천에서 작업이 제거되는 데 최대 2시간이 걸릴 수 있습니다.

반복 빈도 데이터

반복 빈도 데이터는 작업 상호 작용 데이터 세트의 사용자 기록을 기반으로 사용자가 상호 작용한 후 Amazon Personalize가 특정 작업을 추천할 때까지 기다려야 하는 기간을 지정합니다. 작업의 반복 빈도를 일 단위로 지정합니다(최대 30일).

예를 들어, 각 사용자가 계정과 프로필을 생성하는 전자 상거래 애플리케이션이 있을 수 있습니다. complete profile 작업이 있는데 사용자가 상호 작용한 후 일주일을 기다렸다가 다시 추천하려는 경우 작업의 REPEAT_FREQUENCY를 7일로 지정합니다. 7일 후, Amazon Personalize는 추천에 해당 작업을 고려하기 시작합니다.

작업의 반복 빈도를 제공하지 않으면 Amazon Personalize는 추천에 해당 작업이 나타나는 횟수에 제한을 두지 않습니다.

값 데이터

값 데이터는 각 작업의 비즈니스 가치 또는 중요도입니다. 작업의 value는 1~10이 될 수 있으며, 여기서 10이 데이터 세트에서 가장 중요한 작업입니다.

예를 들어, 기본 구독에 등록하기 위한 작업과 프리미엄 서비스에 등록하기 위한 작업이라는 2개의 작업이 있을 수 있습니다. 기본 서비스의 경우 5 값을 지정하고, 프리미엄의 경우 값을 10으로 지정할 수 있습니다.

Amazon Personalize는 사용자에게 추천할 최선의 작업을 결정할 때 값 데이터를 하나의 입력으로 사용합니다. 예를 들어, 사용자가 한 작업 또는 다른 작업을 수행할 확률이 동일한 경우 Amazon Personalize는 추천에서 가장 높은 값을 지닌 작업을 보다 가치 있게 평가합니다.

생성 타임스탬프 데이터

Amazon Personalize는 생성 타임스탬프 데이터(Unix epoch 시간 형식, 초 단위)를 사용하여 작업의 수명을 계산하고 그에 따라 추천을 조정합니다.

생성 타임스탬프 데이터가 없는 경우 Amazon Personalize는 모든 작업 상호 작용 데이터에서 이 정보를 유추합니다. 작업에서 가장 오래된 상호 작용 데이터의 타임스탬프를 작업 생성 타임스탬프로 사용합니다. 작업에 상호 작용 데이터가 없는 경우, 생성 타임스탬프는 교육 세트에 있는 최신 상호 작용의 타임스탬프로 설정되며 Amazon Personalize는 이를 새 작업으로 간주합니다.

범주형 메타데이터

Amazon Personalize는 사용자에게 가장 적합한 작업을 나타내는 기본 패턴을 식별할 때 계절성, 작업 독점성 등 작업에 대한 범주형 메타데이터를 사용합니다. 사용 사례를 기반으로 자체 값 범위를 직접 정의합니다. 범주형 메타데이터는 어떤 언어로든 사용할 수 있습니다.

범주형 데이터를 가져와서 작업 속성에 따라 추천을 필터링하는 데 사용할 수 있습니다. 추천 필터링에 대한 자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 단원을 참조하세요.

범주형 값은 최대 1000자까지 입력할 수 있습니다. 범주형 값이 1,000자를 초과하는 작업이 있는 경우 데이터 세트 가져오기 작업이 실패합니다.

작업 상호 작용 데이터 세트

작업 상호 작용은 [작업 데이터 세트](#)에 있는 사용자와 작업을 포함하는 상호 작용입니다. 작업 상호 작용을 Amazon Personalize 작업 상호 작용 데이터 세트로 가져옵니다. 각 작업 상호 작용은 userID, actionID, 타임스탬프, 이벤트 유형 및 상호 작용에 대한 추가 데이터(예: 범주별 메타데이터)로 구성됩니다.

예를 들어, 작업 데이터 세트에 등록 작업이 있고 사용자가 이 작업을 수행하는 경우 사용자 ID, 작업 ID, 타임스탬프를 기록하고 이벤트 유형에는 TAKEN을 기록합니다. 데이터 세트 가져오기 작업을 통해 작업 상호 작용 이벤트를 대량으로 가져오거나 [PutActionInteractions](#) API 작업을 통해 실시간으로 스트리밍할 수 있습니다. 데이터 가져오기에 대한 자세한 내용은 [단계 2: 데이터 준비 및 가져오기](#) 섹션을 참조하세요.

Note

도메인 데이터세트 그룹에는 액션 및 액션 인터랙션 데이터세트를 비롯한 차선책 액션 리소스를 생성할 수 없습니다.

PERSONALIZED_ACTIONS 사용자 지정 레시피를 사용하는 경우 Amazon Personalize는 작업 상호 작용 데이터 세트의 모든 데이터를 입력으로 이용하여 사용자가 수행할 가능성이 가장 높은 작업을 예측합니다. 작업 상호 작용 데이터에 대한 최소 요구 사항은 없습니다. 품질 작업 추천을 위해서는 데이터를 가져오는 것이 좋습니다. 작업 상호 작용 데이터가 없는 경우 빈 작업 상호 작용 데이터 세트를 만들고 [PutActionInteractions](#) API 작업을 사용하여 고객과 작업의 상호 작용을 기록할 수 있습니다.

행동 상호 작용 데이터를 가져올 때까지 Amazon Personalize는 개인화가 없는 사용자의 행동을 권장하며 성향 점수는 0.0입니다. 다음 조건을 충족하면 해당 작업에 점수가 부여됩니다.

- TAKEN 이벤트 유형과의 작업 상호 작용이 50회 이상 있어야 합니다.
- NOT_TAKEN 또는 VIEWED 이벤트 유형과의 동작 상호 작용이 50개 이상 있어야 합니다.

이러한 액션 인터랙션은 최신 솔루션 버전 교육 시 존재해야 하며, 액션 인터랙션 데이터셋의 최신 인터랙션 타임스탬프로부터 6주 이내에 발생해야 합니다.

다음 주제는 Amazon Personalize에서 사용할 수 있는 작업 상호 작용 데이터에 관하여 추가 정보를 제공합니다.

주제

- [이벤트 유형 데이터](#)

이벤트 유형 데이터

Amazon Personalize는 이벤트 유형 데이터의 패턴을 활용하여 사용자가 수행할 가능성이 가장 높은 작업을 식별할 수 있습니다. 예를 들어, 고객이 이메일 구독 작업(NOT_TAKEN 이벤트 유형으로 표시됨)을 자주 무시하는 경우 Amazon Personalize는 이러한 유형의 작업을 더 적게 포함하도록 추천을 조정할 수 있습니다.

작업 상호 작용 이벤트에는 다음 이벤트 유형만 사용할 수 있습니다. Amazon Personalize는 이러한 이벤트를 통해 사용자에게 대해 파악하고 다음에 추천할 작업을 추정합니다.

- Taken – 사용자가 추천 작업을 실행할 때 Taken 이벤트가 기록됩니다.
- Not Taken – 사용자가 작업을 본 후 작업을 실행하지 않기로 의도적으로 선택한 경우 Not Taken 이벤트가 기록됩니다. 작업을 보여줄 때 아니요를 선택하는 경우를 예로 들 수 있습니다. Not Taken 이벤트는 고객이 해당 작업에 관심이 없음을 의미할 수 있습니다.
- Viewed – 작업을 수행할지 여부를 결정하기 전에 사용자에게 작업을 보여주면 Viewed 이벤트가 기록됩니다. Amazon Personalize는 View 이벤트를 통해 사용자의 관심도를 파악합니다. 예를 들어, 사용자가 작업을 보고 실행하지 않는 경우 이 사용자는 앞으로 해당 작업에 관심이 없을 수 있습니다.

스키마

스키마는 Personalize에 데이터 구조를 알려주고 Personalize가 데이터를 파싱할 수 있도록 해 줍니다. 스키마에는 값이 데이터세트 유형과 일치해야 하는 이름 키가 있습니다. 스키마를 생성하고 나면 스키마를 변경할 수 없습니다.

도메인 데이터세트 그룹의 경우, 각 데이터세트 유형에는 필수 필드와 예약된 키워드가 포함된 기본 스키마가 있습니다. 데이터세트를 생성할 때마다 기존 도메인 스키마를 사용하거나 기존 기본 스키마를 수정하여 새 도메인 스키마를 만들 수 있습니다. 기본 스키마를 도메인에 가져올 데이터에 대한 설명서로 사용하세요. 스키마를 정의하고 데이터세트를 생성한 후에는 스키마를 변경할 수 없습니다.

주제

- [스키마 서식 지적 요구 사항](#)
- [도메인 데이터세트 및 스키마](#)
- [사용자 지정 데이터세트 및 스키마](#)
- [SDK for Python\(Boto3\)을 이용한 스키마 만들기](#)

스키마 서식 지적 요구 사항

도메인 데이터세트 그룹 또는 사용자 지정 데이터세트 그룹에 대한 스키마를 생성할 때는 다음 지침을 따라야만 합니다.

- [Avro 형식](#)으로 스키마를 정의해야 합니다. 지원되는 Avro 데이터 유형에 대한 자세한 내용은 [스키마 데이터 유형](#)을 참조하세요.
- 스키마 필드는 순서대로 나타날 수 있지만, CSV 파일의 해당 열 헤더의 순서와 일치해야 합니다.
- 스키마는 중첩 구조가 없는 플랫폼 JSON 파일이어야 합니다. 예를 들어, 특정 필드는 여러 하위 필드의 부모 필드가 될 수 없습니다.
- Personalize 스키마는 배열 및 맵과 같은 복잡한 유형을 지원하지 않습니다.
- 스키마 필드는 고유한 영숫자 이름을 가져야 합니다. 예를 들어, GENRES_FIELD_1 필드와 GENRESFIELD1 필드를 모두 추가할 수는 없습니다.
- 필수 필드를 필수 데이터 유형으로 정의해야 합니다. 예약된 범주형 문자열 필드는 categorical 속성을 true로 설정해야만 하며, 예약된 문자열 필드는 범주형일 수 없습니다. 키워드는 데이터에 포함될 수 없습니다.

- 고유 메타데이터 유형 string 필드를 추가하고 교육 시 Amazon Personalize에서 사용하도록 하려면 categorical 속성 또는 textual 속성(항목 스키마는 텍스트 속성이 있는 필드만 지원)을 포함해야 합니다.
- Amazon Personalize는 테마 생성, 추천의 메타데이터 반환, 추천 필터링 시 항목 이름 열과 같은 비범주형 문자열 열을 사용할 수 있습니다. 자세한 정보는 [비범주형 문자열 데이터를 참조](#)하세요.
- Personalize는 추천을 교육하거나 필터링할 때 boolean유형 데이터를 사용하지 않습니다. Personalize가 교육 또는 필터링 시 부울 데이터를 사용하도록 하려면, 문자열 유형의 필드를 사용하고 데이터에 값 "True" 및 "False"를 사용합니다. 또는 int 또는 long 유형과 값 0 및 1를 사용할 수 있습니다.
- 텍스트 필드는 해당 유형 string이어야 하며 textual속성이 true로 설정되어 있어야 합니다. 비정형 텍스트 데이터에 대한 자세한 내용은 [비정형 텍스트 메타데이터](#) 단원을 참조하세요.

도메인 데이터세트 그룹 데이터세트에는 도메인과 데이터세트 유형에 따른 추가 요구 사항이 있습니다. 사용자 지정 데이터세트 그룹 데이터세트에는 유형에 따라 추가 요구 사항이 있습니다.

스키마 데이터 유형

Personalize 스키마는 필드에 대해 다음과 같은 Avro 유형을 지원합니다.

- float
- double
- int
- long
- 문자열
- boolean
- null

일부 필수 및 예약 필드는 null 데이터를 지원합니다. 필드에 null유형을 추가하면 불완전한 데이터 (예: 값이 비어 있는 메타데이터)를 사용하여 추천을 생성할 수 있습니다. null 데이터를 지원하는 필드에 대한 자세한 내용은 [도메인 데이터세트 및 스키마](#) 또는 [사용자 지정 데이터세트 및 스키마](#)을 참조하세요. 다음 예제에서는 GENDER 필드에 null을 추가하는 방법을 보여 줍니다.

```
{
  "name": "GENDER",
  "type": [
    "null",
```

```

    "string"
  ],
  "categorical": true
}

```

도메인 데이터세트 및 스키마

도메인 데이터세트 그룹을 생성할 때 선택한 도메인에 따라 데이터세트와 스키마 요구 사항이 결정됩니다. 각 도메인에는 각 데이터세트 유형에 대한 기본 스키마가 있습니다.

데이터세트를 생성할 때 기본 스키마를 사용하거나 기본 스키마를 기반으로 새 데이터세트를 생성할 수 있습니다. 기본 스키마를 어떤 데이터를 수집하고 각 데이터세트 유형으로 가져올 것인지에 대한 지침으로 사용합니다. 다음 항목에서는 각 도메인의 데이터세트 및 스키마 요구 사항을 설명합니다.

Personalize로 가져올 수 있는 데이터 유형에 대한 자세한 내용은 [Personalize가 사용할 수 있는 데이터 유형](#) 단원을 참조하세요.

형식 지정 요구 사항 및 사용 가능한 필드 데이터 유형과 같은 일반적인 Personalize 스키마 요구 사항에 대한 자세한 내용은 [스키마](#) 단원을 참조하세요. 이러한 요구 사항은 도메인과 상관없이 모든 스키마에 적용됩니다.

주제

- [VIDEO_ON_DEMAND 데이터세트 및 스키마](#)
- [ECOMMERCE 데이터세트 및 스키마](#)

VIDEO_ON_DEMAND 데이터세트 및 스키마

VIDEO_ON_DEMAND 도메인에 대한 도메인 데이터세트 그룹을 생성하면 각 데이터세트 유형에는 VIDEO_ON_DEMAND 고유의 필수 및 권장 필드 세트가 포함된 기본 스키마가 있습니다. 기본 스키마를 사용하거나 기본 스키마를 기반으로 새 데이터세트를 생성할 수 있습니다. 가져오는 데이터는 형식 및 유형이 스키마와 일치해야 합니다. 아래 단원에 나열된 기본 도메인 스키마를 지침으로 사용하여 VIDEO_ON_DEMAND 기반 추천자를 생성하기 위해 어떤 데이터를 가져올지 결정합니다.

추가 필드를 자유롭게 추가할 수 있습니다. 필드가 필수 또는 예약으로 나열되어 있지 않고 데이터 유형이 [스키마 데이터 유형](#)에 나열된 경우, 필드 이름과 데이터 사용자가 선택할 수 있습니다.

형식 지정 요구 사항 및 사용 가능한 필드 데이터 유형과 같은 일반적인 Personalize 스키마 요구 사항에 대한 자세한 내용은 [스키마](#) 단원을 참조하세요. 이러한 요구 사항은 도메인과 상관없이 모든 스키마에 적용됩니다.

다음 항목에서는 VIDEO_ON_DEMAND 도메인에 대한 각 데이터세트의 필수 및 권장 필드에 대한 정보를 제공합니다. 각 데이터세트 단원에는 JSON 형식의 기본 VIDEO_ON_DEMAND 스키마가 포함되어 있습니다.

주제

- [VIDEO_ON_DEMAND 도메인 데이터세트 및 스키마 요구 사항](#)
- [항목 상호 작용 데이터 세트 요구 사항\(VIDEO_ON_DEMAND 도메인\)](#)
- [상호작용 데이터세트 요구 사항\(VIDEO_ON_DEMAND 도메인\)](#)
- [항목 데이터세트 요구 사항\(VIDEO_ON_DEMAND 도메인\)](#)

VIDEO_ON_DEMAND 도메인 데이터세트 및 스키마 요구 사항

각 데이터세트에는 다음 필수 필드와 예약어가 있습니다. 예약어는 메타데이터가 아닌 선택형 필드입니다. 사용할 때 필드를 필수 데이터 유형으로 정의해야 하므로 이러한 필드는 예약된 것으로 간주됩니다. 예약된 범주형 문자열 필드는 `categorical`을 `true`로 설정되어 있어야 하지만 예약된 문자열 필드는 범주형일 수 없습니다. 키워드는 데이터에 포함될 수 없습니다.

데이터세트 유형	필수 필드	예약어
항목 상호 작용(기본 스키마)	USER_ID(string) ITEM_ID(string) TIMESTAMP(long) EVENT_TYPE(string 및 사용 사례 , Watch 및 Click 이벤트 유형에 따라 달라짐)	EVENT_VALUE(float, null) IMPRESSION(string, null) RECOMMENDATION_ID(string, null) EVENT_ATTRIBUTION_SOURCE(string, null)
사용자(기본 스키마)	USER_ID(string) 메타데이터 필드 1개(범주형 string 또는 숫자형)	SUBSCRIPTION_MODEL(범주형 string, null)
항목(기본 스키마)	ITEM_ID(string)	PRICE(float, null) DURATION(float, null)

데이터세트 유형	필수 필드	예약어
	CREATION_TIMESTAMP (long)	GENRE_L2(범주형 string, null)
	GENRES(범주형 string)	GENRE_L3(범주형 string, null)
		AVERAGE_RATING(float, null)
		PRODUCT_DESCRIPTION(텍스트형 string, null)
		CONTENT_OWNER(범주형 string, null)
		CONTENT_CLASSIFICATION(범주형 string, null)

항목 상호 작용 데이터 세트 요구 사항(VIDEO_ON_DEMAND 도메인)

항목 상호 작용 데이터 세트는 사용자와 항목 간 상호 작용의 이력 및 실시간 데이터를 VIDEO_ON_DEMAND 카탈로그에 저장합니다. 상호작용 데이터세트에 저장할 수 있는 데이터 유형에 대한 자세한 내용은 [항목 상호 작용 데이터 세트](#) 단원을 참조하세요.

모든 사용 사례에 대한 항목 상호 작용 데이터 세트가 있어야 하며, 스키마에는 다음 필드가 있어야 합니다.

- USER_ID(string)
- ITEM_ID string
- TIMESTAMP(long)
- EVENT_TYPE(string 및 [사용 사례](#), Watch 및 Click 이벤트 유형에 따라 달라짐)

다음과 같은 예약어를 스키마에 포함할 수도 있습니다.

- EVENT_VALUE(float, null)
- IMPRESSION(string, null)

- RECOMMENDATION_ID(string, null)

사용 사례와 데이터에 따라 추가 필드를 마음대로 추가할 수 있습니다. 필드가 필수 또는 예약어로 나열되어 있지 않고 데이터 유형이 [스키마 데이터 유형](#)에 나열된 경우, 필드 이름과 데이터 사용자가 선택할 수 있습니다. VIDEO_ON_DEMAND 도메인의 항목 상호 작용 데이터 세트에 대한 기본 스키마의 예제는 [기본 상호작용 스키마\(VIDEO_ON_DEMAND 도메인\)](#) 섹션을 참조하세요.

시청한 동영상의 비율과 같은 이벤트에 대한 값 데이터가 있는 경우 예약어 EVENT_VALUE를 원하는 대로 추가할 수 있습니다. 명시적 노출 및 암시적 노출 데이터를 포함하려는 경우 예약어 IMPRESSION을 추가할 수도 있습니다. 노출 데이터 기록에 대한 자세한 내용은 [노출 데이터](#) 단원을 참조하세요.

항목 상호 작용 데이터 세트에 추가할 수 있는 선택형 메타데이터 필드의 최대 개수는 항목 상호 작용 데이터에 있는 개별 이벤트 유형의 총 개수와 합산하여 10개입니다. 이 수치에 포함되는 메타데이터 필드는 스키마에 추가한 사용자 지정 메타데이터 필드와 함께 EVENT_TYPE, EVENT_VALUE 필드입니다. 예약된 필드(예: IMPRESSION)를 제외한 메타데이터 필드의 최대 수는 5개입니다. 범주형 값은 최대 1000자까지 포함할 수 있습니다. 범주형 값이 1000자를 초과하는 상호작용이 있는 경우, 데이터세트 가져오기 작업이 실패합니다.

VIDEO_ON_DEMAND 도메인의 항목 상호 작용 데이터 세트에 대한 자세한 최소 요구 사항 및 최대 데이터 한도 정보는 [Service quotas](#) 섹션을 참조하세요. 형식 지정 요구 사항 및 사용 가능한 필드 데이터 유형과 같은 일반적인 Personalize 스키마 요구 사항에 대한 자세한 내용은 [스키마](#) 단원을 참조하세요. 이러한 요구 사항은 도메인과 상관없이 모든 스키마에 적용됩니다.

기본 상호작용 스키마(VIDEO_ON_DEMAND 도메인)

다음은 항목 상호 작용 데이터 세트의 기본 VIDEO_ON_DEMAND 도메인 스키마입니다.

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "ITEM_ID",
```

```

        "type": "string"
    },
    {
        "name": "EVENT_TYPE",
        "type": "string"
    },
    {
        "name": "TIMESTAMP",
        "type": "long"
    }
],
"version": "1.0"
}

```

상호작용 데이터세트 요구 사항(VIDEO_ON_DEMAND 도메인)

사용자 데이터세트는 사용자에게 대한 메타데이터를 저장합니다. 여기에는 각 항목에 대한 연령, 성별, 로열티 멤버십과 같은 정보가 포함될 수 있습니다. Personalize로 가져올 수 있는 사용자 데이터 유형에 대한 자세한 내용은 [사용자 데이터 세트](#) 단원을 참조하세요. 형식 지정 요구 사항 및 사용 가능한 필드 데이터 유형과 같은 일반적인 Personalize 스키마 요구 사항에 대한 자세한 내용은 [스키마](#) 단원을 참조하세요. 이러한 요구 사항은 도메인과 상관없이 모든 스키마에 적용됩니다.

사용자 데이터세트는 모든 VIDEO_ON_DEMAND 사용 사례에 대해 선택 사항입니다. 사용자 데이터가 있는 경우, 가장 관련성이 높은 추천을 받을 수 있도록 한 개의 사용자 데이터를 생성하는 것이 좋습니다. 항목 데이터세트를 만드는 경우, 스키마에 다음 필드가 포함되어야 합니다.

- USER_ID
- 메타데이터 필드 1개(범주형 string 또는 숫자형)

사용 사례와 데이터에 따라 추가 필드를 마음대로 추가할 수 있습니다. 필드가 필수 또는 예약으로 나열되어 있지 않고 데이터 유형이 [스키마 데이터 유형](#)에 나열된 경우, 필드 이름과 데이터 사용자가 선택할 수 있습니다. VIDEO_ON_DEMAND 도메인의 사용자 상호작용 데이터세트에 대한 기본 스키마의 예제는 [기본 사용자 스키마\(VIDEO_ON_DEMAND 도메인\)](#) 단원을 참조하세요.

SUBSCRIPTION_MODEL 필드는 기본 스키마에 포함됩니다. 이 필드는 선택형 예약어이며 범주형이 true로 설정된 상태에서 유형 string이어야 합니다. 데이터에 각 사용자에게 대한 구독 모델 정보가 있는 경우, 최상의 추천을 받으려면 이 필드를 스키마에 유지하는 것이 좋습니다. 가져오는 데이터는 스키마와 일치해야 합니다.

범주형 데이터 사용

범주형 데이터를 사용하려면 스키마에서 유형 string 필드를 추가하고 필드의 범주형 속성을 true로 설정합니다. 그런 다음 대량 CSV 파일 및 개별 항목 가져오기에 범주형 데이터를 포함합니다. 여러 범주를 갖는 개인 사용자의 경우, 세로 막대 '|'를 사용하여 각 값을 구분합니다. 예를 들어, SUBSCRIPTION_MODEL 필드의 경우 사용자에 대한 데이터는 학생|월별|할인이 될 수 있습니다.

범주형 값은 최대 1000자까지 포함할 수 있습니다. 범주형 값이 1000자를 초과하는 사용자가 있는 경우, 데이터세트 가져오기 작업이 실패합니다.

기본 사용자 스키마(VIDEO_ON_DEMAND 도메인)

다음은 사용자 데이터세트의 기본 VIDEO_ON_DEMAND 도메인 스키마입니다.

```
{
  "type": "record",
  "name": "Users",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "SUBSCRIPTION_MODEL",
      "type": "string",
      "categorical": true
    }
  ],
  "version": "1.0"
}
```

항목 데이터세트 요구 사항(VIDEO_ON_DEMAND 도메인)

항목 데이터세트는 카탈로그의 항목에 대한 메타데이터를 저장합니다. 여기에는 각 항목의 가격, 장르, 재고 여부 등의 정보가 포함될 수 있습니다. Personalize로 가져올 수 있는 항목 데이터 유형에 대한 자세한 내용은 [항목 데이터 세트](#) 단원을 참조하세요. 형식 지정 요구 사항 및 사용 가능한 필드 데이터 유형과 같은 일반적인 Personalize 스키마 요구 사항에 대한 자세한 내용은 [스키마](#) 단원을 참조하세요. 이러한 요구 사항은 도메인과 상관없이 모든 스키마에 적용됩니다.

일부 사용 사례에는 항목 데이터세트가 필요합니다([VIDEO_ON_DEMAND 사용 사례](#) 참조). 선택 사항인 경우에도 가장 관련성이 높은 추천을 받으려면 하나를 새로 만드는 것이 좋습니다. 항목 데이터세트를 만드는 경우, 스키마에 다음 필드가 포함되어야 합니다.

- ITEM_ID
- GENRES(범주형 string)
- CREATION_TIMESTAMP(Unix 에포크 타임 형식으로)

다음과 같은 예약어를 스키마에 포함할 수도 있습니다. 각 예약어에는 필수 데이터 유형과 null 데이터 지원 여부가 나열됩니다. null 유형을 추가하는 것은 선택 사항입니다.

- PRICE(float)
- DURATION(float)
- GENRE_L2(범주형 string, null)
- GENRE_L3(범주형 string, null)
- AVERAGE_RATING(float, null)
- PRODUCT_DESCRIPTION(텍스트형 string, null)
- CONTENT_OWNER(범주형 string, null): 동영상을 소유한 회사입니다. 예를 들어, 값은 HBO, 파라마운트, NBC일 수 있습니다.
- CONTENT_CLASSIFICATION(범주형 string, null): 콘텐츠의 등급입니다. 예를 들어, 값은 G, PG, PG-13, R, NC-17 및 등급보류일 수 있습니다.

최상의 추천을 얻으려면 이러한 필드를 데이터의 수만큼 스키마에 보관하는 것이 좋습니다. 가져오는 데이터는 스키마와 일치해야 합니다. 최대 메타데이터 열 수는 100개입니다. 사용 사례와 데이터에 따라 추가 필드를 마음대로 추가할 수 있습니다. 필드가 필수 또는 예약어로 나열되어 있지 않고 데이터 유형이 [스키마 데이터 유형](#)에 나열된 경우, 필드 이름과 데이터 사용자가 선택할 수 있습니다.

다중 레벨 범주가 여러 개 있는 항목에는 예약어 GENRE_L2 및 GENRE_L3을 사용합니다. 자세한 내용은 [범주형 데이터 사용](#) 단원을 참조하세요. 텍스트 및 범주형 메타데이터에 대한 자세한 내용은 [항목 데이터 세트](#)를 참조하세요. ECOMMERCE 도메인의 항목 데이터세트에 대한 기본 스키마의 예제는 [기본 항목 스키마\(VIDEO_ON_DEMAND 도메인\)](#) 단원을 참조하세요.

범주형 데이터 사용

범주형 데이터를 사용하려면 스키마에서 유형 string 필드를 추가하고 필드의 범주형 속성을 true로 설정합니다. 그런 다음 대량 CSV 파일 및 개별 항목 가져오기에 범주형 데이터를 포함합니다. 범주형

값은 최대 1000자까지 포함할 수 있습니다. 범주형 값이 1000자를 초과하는 항목이 있는 경우, 데이터 세트 가져오기 작업이 실패합니다.

여러 범주를 갖는 항목의 경우, 세로 막대 '|'로 각 값을 구분합니다. 예를 들어, GENRES 필드의 경우 항목에 대한 데이터가 Action|Crime|Biopic이 될 수 있습니다. 여러 수준의 범주형 데이터가 있고 일부 항목에 계층의 각 수준에 대해 여러 범주가 있는 경우, 각 수준에 필드를 추가하고 각 필드 이름 뒤에 GENRES, GENRE_L2, GENRE_L3 등의 수준 지표를 추가합니다. 이렇게 하면 항목이 여러 멀티 레벨 범주에 속하더라도 하위 범주를 기준으로 추천을 필터링할 수 있습니다. 예를 들어, 동영상에는 각 범주 수준에 대해 다음과 같은 데이터가 있을 수 있습니다.

- GENRES: 액션|어드벤처
- GENRE_L2: 범죄|서부영화
- GENRE_L3: 일대기 영화

이 예제에서는 동영상이 액션 > 범죄 > 일대기 영화 계층 및 모험 > 서부 영화 > 일대기 영화 계층 구조로 되어 있습니다. L3까지만 사용하는 것이 좋지만, 필요한 경우 더 많은 레벨을 사용할 수 있습니다. 필터 생성 및 사용에 대한 자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 단원을 참조하세요.

기본 항목 스키마(VIDEO_ON_DEMAND 도메인)

다음은 VIDEO_ON_DEMAND 도메인의 항목 데이터세트에 대한 기본 스키마입니다.

```
{
  "type": "record",
  "name": "Items",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "GENRES",
      "type": [
        "string"
      ],
      "categorical": true
    },
    {
      "name": "CREATION_TIMESTAMP",
```

```

    "type": "long"
  }
],
"version": "1.0"
}

```

ECOMMERCE 데이터세트 및 스키마

ECOMMERCE 도메인에 대한 도메인 데이터세트 그룹을 생성하면 각 데이터세트 유형에는 ECOMMERCE 고유의 필수 및 권장 필드 세트가 포함된 기본 스키마가 있습니다. 기본 스키마를 사용하거나 기본 스키마를 기반으로 새 데이터세트를 생성할 수 있습니다. 가져오는 데이터는 형식 및 유형이 스키마와 일치해야 합니다. 아래 단원에 나열된 기본 도메인 스키마를 지침으로 사용하여 ECOMMERCE 기반 추천자를 생성하기 위해 어떤 데이터를 가져올지 결정합니다.

추가 필드를 자유롭게 추가할 수 있습니다. 필드가 필수 또는 예약으로 나열되어 있지 않고 데이터 유형이 [스키마 데이터 유형](#)에 나열된 경우, 필드 이름과 데이터 사용자가 선택할 수 있습니다.

형식 지정 요구 사항 및 사용 가능한 필드 데이터 유형과 같은 일반적인 Personalize 스키마 요구 사항에 대한 자세한 내용은 [스키마](#) 단원을 참조하세요. 이러한 요구 사항은 도메인과 상관없이 모든 스키마에 적용됩니다.

다음 항목에서는 ECOMMERCE 도메인에 대한 각 데이터세트의 필수 및 권장 필드에 대한 정보를 제공합니다. 각 데이터세트 단원에는 JSON 형식의 기본 ECOMMERCE 스키마가 포함되어 있습니다.

주제

- [ECOMMERCE 도메인 데이터세트 및 스키마 요구 사항](#)
- [항목 상호 작용 데이터 세트 요구 사항\(ECOMMERCE 도메인\)](#)
- [상호작용 데이터세트 요구사항\(ECOMMERCE 도메인\)](#)
- [항목 데이터세트 요구사항\(ECOMMERCE 도메인\)](#)

ECOMMERCE 도메인 데이터세트 및 스키마 요구 사항

각 데이터세트에는 다음 필수 필드와 예약어가 있습니다. 예약어는 메타데이터가 아닌 선택형 필드입니다. 사용할 때 필드를 필수 데이터 유형으로 정의해야 하므로 이러한 필드는 예약된 것으로 간주됩니다. 예약된 범주형 문자열 필드는 `categorical`을 `true`로 설정되어 있어야 하지만 예약된 문자열 필드는 범주형일 수 없습니다. 키워드는 데이터에 포함될 수 없습니다.

데이터세트 유형	필수 필드	예약어
항목 상호 작용(기본 스키마)	USER_ID(string) ITEM_ID(string) TIMESTAMP(long) EVENT_TYPE(string 및 사용 사례 , Purchase 및 View 이벤트 유형에 따라 달라짐)	EVENT_VALUE(float, null) IMPRESSION(string, null) RECOMMENDATION_ID(string, null) EVENT_ATTRIBUTION_SOURCE(string, null)
사용자(기본 스키마)	USER_ID(string) 메타데이터 필드 1개(범주형 string 또는 숫자형)	
항목(기본 스키마)	ITEM_ID(string) PRICE(float) CATEGORY_L1(범주형 string)	CATEGORY_L2(범주형, string) null CATEGORY_L3(범주형 string, null) PRODUCT_DESCRIPTION(텍스트형 string, null) CREATION_TIMESTAMP(long) AGE_GROUP(범주별 string, null) ADULT(범주형 string, null) GENDER(범주형 string, null)

항목 상호 작용 데이터 세트 요구 사항(ECOMMERCE 도메인)

항목 상호 작용 데이터 세트는 사용자와 항목 간 상호 작용의 이력 및 실시간 데이터를 ECOMMERCE 카탈로그에 저장합니다. 상호작용 데이터세트에 저장할 수 있는 데이터 유형에 대한 자세한 내용은 [항목 상호 작용 데이터 세트](#) 단원을 참조하세요. 형식 지정 요구 사항 및 사용 가능한 필드 데이터 유형과 같은 일반적인 Personalize 스키마 요구 사항에 대한 자세한 내용은 [스키마](#) 단원을 참조하세요. 이러한 요구 사항은 도메인과 상관없이 모든 스키마에 적용됩니다.

최소한 항목 상호 작용 데이터 세트를 생성해야 하며, 스키마에는 다음 필드가 있어야 합니다.

- USER_ID(string)
- ITEM_ID(string)
- TIMESTAMP(long)
- EVENT_TYPE(string 및 [사용 사례](#), Purchase 및 View 이벤트 유형에 따라 달라짐)

다음과 같은 예약어를 스키마에 포함할 수도 있습니다.

- EVENT_VALUE(float, null)
- IMPRESSION(string, null)
- RECOMMENDATION_ID(string, null)

가져오는 데이터는 스키마와 일치해야 합니다. 사용 사례와 데이터에 따라 추가 필드를 마음대로 추가할 수 있습니다. 필드가 필수 또는 예약어로 나열되어 있지 않고 데이터 유형이 [스키마 데이터 유형](#)에 나열된 경우, 필드 이름과 데이터 사용자가 선택할 수 있습니다. ECOMMERCE 도메인의 항목 상호 작용 데이터 세트에 대한 기본 스키마의 예제는 [기본 상호작용 스키마\(ECOMMERCE 도메인\)](#) 섹션을 참조하세요.

이벤트에 대한 값 데이터가 있는 경우 예약어 EVENT_VALUE를 원하는 대로 추가할 수 있습니다. 명시적 노출 및 암시적 노출 데이터를 포함하려는 경우 예약어 IMPRESSION을 추가할 수도 있습니다. 노출 데이터 기록에 대한 자세한 내용은 [노출 데이터](#) 단원을 참조하세요.

항목 상호 작용 데이터 세트에 추가할 수 있는 선택형 메타데이터 필드의 최대 개수는 항목 상호 작용 데이터에 있는 개별 이벤트 유형의 총 개수와 합산하여 10개입니다. 이 수치에 포함되는 메타데이터 필드는 스키마에 추가한 사용자 지정 메타데이터 필드와 함께 EVENT_TYPE, EVENT_VALUE 필드입니다. 예약된 필드(예: IMPRESSION)를 제외한 메타데이터 필드의 최대 수는 5개입니다. 범주형 값은 최대 1000자까지 포함할 수 있습니다. 범주형 값이 1000자를 초과하는 상호작용이 있는 경우, 데이터세트 가져오기 작업이 실패합니다.

ECOMMERCE 도메인의 항목 상호 작용 데이터 세트에 대한 자세한 최소 요구 사항 및 최대 데이터 한도 정보는 [Service quotas](#) 섹션을 참조하세요.

기본 상호작용 스키마(ECOMMERCE 도메인)

다음은 항목 상호 작용 데이터 세트의 기본 ECOMMERCE 도메인 스키마입니다.

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "EVENT_TYPE",
      "type": "string"
    },
    {
      "name": "TIMESTAMP",
      "type": "long"
    }
  ],
  "version": "1.0"
}
```

상호작용 데이터세트 요구사항(ECOMMERCE 도메인)

사용자 데이터세트는 사용자에 대한 메타데이터를 저장합니다. 여기에는 각 사용자의 연령, 성별, 로열티 멤버십과 같은 정보가 포함될 수 있습니다. Personalize로 가져올 수 있는 사용자 데이터 유형에 대한 더 자세한 내용은 [사용자 데이터 세트](#) 단원을 참조하세요. 형식 지정 요구 사항 및 사용 가능한 필드 데이터 유형과 같은 일반적인 Personalize 스키마 요구 사항에 대한 자세한 내용은 [스키마](#) 단원을 참조하세요. 이러한 요구 사항은 도메인과 상관없이 모든 스키마에 적용됩니다.

사용자 데이터세트는 모든 ECOMMERCE 사용 사례에 대해 선택 사항입니다. 사용자 데이터가 있는 경우, 가장 관련성이 높은 추천을 받을 수 있도록 한 개의 사용자 데이터를 생성하는 것이 좋습니다.. 항목 데이터세트를 만드는 경우, 스키마에 다음 필드가 포함되어야 합니다.

- USER_ID
- 메타데이터 필드 1개(범주형 string 또는 숫자형)

가져오는 데이터는 스키마와 일치해야 합니다. 사용 사례와 데이터에 따라 추가 필드를 마음대로 추가할 수 있습니다. 필드가 필수 또는 예약으로 나열되어 있지 않고 데이터 유형이 [스키마 데이터 유형](#)에 나열된 경우, 필드 이름과 데이터 사용자가 선택할 수 있습니다. ECOMMERCE 도메인의 사용자 데이터 세트에 대한 기본 스키마의 예제는 [기본 사용자 스키마\(ECOMMERCE 도메인\)](#) 단원을 참조하세요.

사용자 데이터세트에 대한 최소 요구 사항 및 최대 데이터 한도에 대한 자세한 내용은 [Service quotas](#) 단원을 참조하세요.

범주형 데이터 사용

범주형 데이터를 사용하려면 스키마에서 유형 string 필드를 추가하고 필드의 범주형 속성을 true로 설정합니다. 그런 다음 대량 CSV 파일 및 개별 항목 가져오기에 범주형 데이터를 포함합니다. 여러 범주를 갖는 개인 사용자의 경우, 세로 막대 '|'를 사용하여 각 값을 구분합니다. 예를 들어, SUBSCRIPTION_MODEL 필드의 경우 사용자에 대한 데이터는 학생|월별|할인이 될 수 있습니다.

범주형 값은 최대 1000자까지 포함할 수 있습니다. 범주형 값이 1000자를 초과하는 사용자가 있는 경우, 데이터세트 가져오기 작업이 실패합니다.

기본 사용자 스키마(ECOMMERCE 도메인)

다음은 CATEGORY 필드가 필수 메타데이터 필드인 사용자 데이터세트의 기본 ECOMMERCE 도메인 스키마입니다.

```
{
  "type": "record",
  "name": "Users",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    }
  ],
}
```

```

    {
      "name": "MEMBERSHIP_STATUS",
      "type": "string",
      "categorical": true
    }
  ],
  "version": "1.0"
}

```

항목 데이터셋 요구사항(ECOMMERCE 도메인)

항목 데이터셋은 ECOMMERCE 항목에 대한 메타데이터를 저장합니다. 여기에는 각 항목에 대한 가격, 범주, 제품 설명 등의 정보가 포함될 수 있습니다. Personalize로 가져올 수 있는 항목 데이터 유형에 대한 더 자세한 내용은 [항목 데이터 세트](#) 단원을 참조하세요. 형식 지정 요구 사항 및 사용 가능한 필드 데이터 유형과 같은 일반적인 Personalize 스키마 요구 사항에 대한 자세한 내용은 [스키마](#) 단원을 참조하세요. 이러한 요구 사항은 도메인과 상관없이 모든 스키마에 적용됩니다.

항목 데이터셋은 모든 ECOMMERCE 사용 사례에 대해 선택 사항입니다. 항목 데이터가 있는 경우, 가장 관련성이 높은 추천을 받을 수 있도록 한 개의 사용자 데이터를 생성하는 것이 좋습니다.. 항목 데이터셋을 만드는 경우, 스키마에 다음 필드가 포함되어야 합니다.

- ITEM_ID
- PRICE(float)
- CATEGORY_L1(범주형 string)

다음과 같은 예약어를 스키마에 포함할 수도 있습니다. 범주형 필드의 경우, 사용 사례를 기반으로 자체 값 범위를 직접 정의합니다.

- CATEGORY_L2(범주형string,null)
- CATEGORY_L3(범주형string,null)
- PRODUCT_DESCRIPTION(텍스트형 string, null)
- CREATION_TIMESTAMP(float)
- AGE_GROUP(범주형string,null): 항목이 대상 연령대입니다. 값은 신생아, 유아, 어린이, 성인일 수 있습니다.
- ADULT(범주형 string, null): 항목이 알코올과 같이 성인으로만 제한되는지 여부. 값은 예 또는 아니오일 수 있습니다.
- GENDER(범주형string,null): 항목이 성별입니다. 값은 남성, 여성, 남녀공용일 수 있습니다.

최상의 추천을 얻으려면 이러한 필드를 데이터의 수만큼 스키마에 보관하는 것이 좋습니다. 가져오는 데이터는 스키마와 일치해야 합니다. 가져오는 데이터는 스키마와 일치해야 합니다. 최대 메타데이터 열 수는 100개입니다. 사용 사례와 데이터에 따라 추가 필드를 마음대로 추가할 수 있습니다. 필드가 필수 또는 예약으로 나열되어 있지 않고 데이터 유형이 [스키마 데이터 유형](#)에 나열된 경우, 필드 이름과 데이터 사용자가 선택할 수 있습니다.

다중 레벨 범주가 여러 개 있는 항목에는 예약어 CATEGORY_L2 및 CATEGORY_L3을 사용합니다. 자세한 내용은 [범주형 데이터 사용](#) 단원을 참조하세요. 텍스트 및 범주형 메타데이터에 대한 자세한 내용은 [비정형 텍스트 메타데이터](#)를 참조하세요. ECOMMERCE 도메인의 항목 데이터세트에 대한 기본 스키마의 예제는 [기본 항목 스키마\(ECOMMERCE 도메인\)](#) 단원을 참조하세요.

범주형 데이터 사용

범주형 데이터를 사용하려면 스키마에서 유형 string 필드를 추가하고 필드의 범주형 속성을 true로 설정합니다. 그런 다음 대량 CSV 파일 및 개별 항목 가져오기에 범주형 데이터를 포함합니다. 사용 사례를 기반으로 자체 값 범위를 직접 정의할 수 있습니다. 범주형 값은 최대 1000자까지 포함할 수 있습니다. 범주형 값이 1000자를 초과하는 항목이 있는 경우, 데이터세트 가져오기 작업이 실패합니다.

여러 범주를 갖는 항목의 경우, 세로 막대 '|'로 각 값을 구분합니다. 예를 들어, CATEGORY_L1 필드의 경우 항목에 대한 데이터가 Electronics|Productivity|Mouse이 될 수 있습니다. 여러 수준의 범주형 데이터가 있고 일부 항목에 계층의 각 수준에 대해 여러 범주가 있는 경우, 각 수준에 대한 필드를 추가하고 각 필드 이름 뒤에 수준 표시기(CATEGORY_L1, CATEGORY_L2, CATEGORY_L3)를 추가하세요. 이렇게 하면 항목이 여러 멀티 레벨 범주에 속하더라도 하위 범주를 기준으로 추천을 필터링할 수 있습니다. 예를 들어, 항목에는 각 범주 수준에 대해 다음과 같은 데이터가 있을 수 있습니다.

- CATEGORY_L1: 전자제품|생산성
- CATEGORY_L2: 생산성|컴퓨터
- CATEGORY_L3: 마우스

이 예제에서 항목은 전자 제품 > 생산성 > 마우스 계층 구조 및 생산성 > 컴퓨터 > 마우스 계층 구조입니다. L3까지만 사용하는 것이 좋지만, 필요한 경우 더 많은 레벨을 사용할 수 있습니다. 필터 생성 및 사용에 대한 자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 단원을 참조하세요.

기본 항목 스키마(ECOMMERCE 도메인)

다음은 필수 필드만 포함된 ECOMMERCE 도메인의 항목 데이터세트에 대한 기본 스키마입니다.

```
{
```

```
"type": "record",
"name": "Items",
"namespace": "com.amazonaws.personalize.schema",
"fields": [
  {
    "name": "ITEM_ID",
    "type": "string"
  },
  {
    "name": "PRICE",
    "type": "float"
  },
  {
    "name": "CATEGORY_L1",
    "type": [
      "string"
    ],
    "categorical": true
  }
],
"version": "1.0"
}
```

사용자 지정 데이터세트 및 스키마

사용자 지정 데이터세트 그룹을 만들 때는 처음부터 자체 스키마를 생성합니다. 사용자 지정 데이터세트 그룹 데이터세트와 스키마는 필수 필드 수가 더 적고 유연성이 높습니다. 다음 주제에서는 사용자 지정 데이터 세트 그룹의 데이터 세트에 대한 스키마와 데이터 요구 사항을 설명합니다. 각 데이터세트 단위에는 데이터세트 유형을 위한 필수 데이터가 나열되고 스키마의 JSON 예제를 제공합니다.

Personalize로 가져올 수 있는 데이터 유형에 대한 자세한 내용은 [데이터세트](#) 단원을 참조하세요. 형식 지정 요구 사항 및 사용 가능한 필드 데이터 유형과 같은 일반적인 Personalize 스키마 요구 사항에 대한 자세한 내용은 [스키마](#) 단원을 참조하세요. 이러한 요구 사항은 모든 Personalize 스키마에 적용됩니다.

주제

- [사용자 지정 데이터세트 및 스키마 요구 사항](#)
- [항목 상호 작용 데이터 세트 스키마 요구 사항\(사용자 지정\)](#)
- [사용자 데이터 세트 스키마 요구 사항\(사용자 지정\)](#)
- [항목 데이터 세트 스키마 요구 사항\(사용자 지정\)](#)

- [작업 데이터 세트 스키마 요구 사항\(사용자 지정\)](#)
- [작업 상호 작용 데이터 세트 스키마 요구 사항\(사용자 지정\)](#)

사용자 지정 데이터세트 및 스키마 요구 사항

사용자 지정 데이터세트 그룹의 데이터세트를 만들 때 각 데이터세트 유형에는 다음과 같은 필수 필드와 필수 데이터 유형이 포함된 예약어가 있습니다.

데이터세트 유형	필수 필드	예약어
항목 상호 작용(스키마 예제)	USER_ID(string) ITEM_ID(string) TIMESTAMP(long)	EVENT_TYPE(string) EVENT_VALUE(float, null) IMPRESSION(string, null) RECOMMENDATION_ID(string, null) EVENT_ATTRIBUTION_SOURCE(string, null)
사용자(스키마 예제)	USER_ID(string) 메타데이터 필드 1개(범주형 string 또는 숫자형)	
항목(스키마 예제)	ITEM_ID(string) 메타데이터 필드 1개(범주형 또는 텍스트형 string 필드 또는 숫자형 필드)	CREATION_TIMESTAMP(long)
작업(스키마 예제)	ACTION_ID(string) 메타데이터 필드 1개(범주형 string 또는 숫자형)	CREATION_TIMESTAMP(long) VALUE(long, null) TYPE(string, null)

데이터세트 유형	필수 필드	예약어
		EXPIRATION_TIMESTAMP(long, null) REPEAT_FREQUENCY(long, null)
작업 상호 작용(스키마 예제)	USER_ID(string) ACTION_ID(string) EVENT_TYPE(string) TIMESTAMP(long)	IMPRESSION(string, null) RECOMMENDATION_ID(string, null)

메타데이터 필드

메타데이터에는 필수가 아니거나 예약어를 사용하지 않는 문자열 또는 문자열이 아닌 필드가 포함됩니다. 메타데이터 스키마에는 다음과 같은 제한이 있습니다.

- 사용자, 항목 및 작업 스키마에는 하나 이상의 메타데이터 필드가 필요합니다.
- 사용자 스키마의 경우 최대 25개의 메타데이터 필드, 항목 스키마의 경우 100개의 메타데이터 필드, 작업 스키마의 경우 10개의 메타데이터 필드를 추가할 수 있습니다.
- 유형 `string`의 자체 메타데이터 필드를 추가하는 경우, `categorical`속성이나 `textual`속성을 포함해야만 합니다(항목 스키마는 텍스트 속성이 있는 필드만 지원함). 그렇지 않으면 Personalize는 모델을 학습할 때 필드를 사용하지 않습니다.

예약어

예약어는 메타데이터가 아닌 선택형 필드입니다. 이러한 필드는 사용할 때 필수 데이터 유형으로 필드를 정의해야 하고 예약어는 데이터의 값으로 사용할 수 없으므로 예약된 것으로 간주됩니다. 예약된 범주형 문자열 필드는 `categorical`을 `true`로 설정되어 있어야 하지만 예약된 문자열 필드는 범주형일 수 없습니다. 다음은 예약어입니다.

- `EVENT_TYPE`: 하나 이상의 이벤트 유형(예: 클릭 및 다운로드 모두)이 있는 항목 상호 작용 데이터 세트의 경우, `EVENT_TYPE` 필드를 사용합니다. `EVENT_TYPE` 필드를 `string`으로 정의해야 하며 범주형으로 설정할 수 없습니다.

- **EVENT_VALUE**: 이벤트에 대한 값 데이터(예: 사용자가 시청한 동영상의 비율)가 포함된 항목 상호 작용 데이터 세트의 경우, 유형 float 및 선택형 null이 포함된 EVENT_VALUE 필드를 사용합니다.
- **CREATION_TIMESTAMP**: 각 항목의 생성일에 대한 타임스탬프가 있는 항목 또는 작업 데이터 세트의 경우, 유형 long이 있는 CREATION_TIMESTAMP 필드를 사용합니다. Personalize는 CREATION_TIMESTAMP 데이터를 사용하여 항목의 수명을 계산하고 그에 따라 추천을 조정합니다. [생성 타임스탬프 데이터](#) 단원을 참조하세요.
- **IMPRESSION**: 명시적 노출 데이터가 포함된 항목 상호 작용 데이터 세트의 경우, 유형 String이 있고 선택형 유형 null이 있는 IMPRESSION 필드를 사용합니다. 노출은 사용자가 특정 항목과 상호 작용(예: 클릭 또는 시청)했을 때 볼 수 있었던 항목의 목록입니다. 자세한 내용은 [노출 데이터](#) 단원을 참조하세요.
- **RECOMIDATION_ID**: 이전 추천을 암시적 노출 데이터로 사용하는 항목 상호 작용 데이터 세트의 경우, 유형 String이 있고 선택형 유형 null이 있는 RECOMMENDATION_ID 필드를 필요에 따라 사용합니다.

추천을 생성할 때 Personalize에 암시적 노출을 사용하기 위해 RECOMMENDATION_ID 필드를 추가할 필요는 없습니다. 필드가 없어도 [PutEvents](#) 작업에서 recommendationId를 전달할 수 있습니다. 자세한 내용은 [노출 데이터](#) 단원을 참조하세요.

- **VALUE**: 작업 데이터 세트의 경우, 일부 또는 모든 작업에 데이터에 대한 값이 있다면 스키마에 VALUE 필드를 추가합니다. 유형에는 long을 사용하고 필요에 따라 null 유형을 사용합니다. 작업 및 작업 값에 대한 자세한 내용은 [값 데이터](#) 섹션을 참조하세요.
- **ACTION_EXPIRATION_TIMESTAMP**: 작업 데이터 세트의 경우, 일부 또는 모든 작업에 만료 타임스탬프가 있다면 스키마에 ACTION_EXPIRATION_TIMESTAMP 필드를 추가합니다. 유형에는 long을 사용하고 필요에 따라 null 유형을 사용합니다. 만료 타임스탬프에 대한 자세한 내용은 [작업 만료 타임스탬프 데이터](#) 섹션을 참조하세요.
- **REPEAT_FREQUENCY**: 작업 데이터 세트의 경우, 일부 또는 모든 작업에 반복 빈도 데이터가 있다면 스키마에 REPEAT_FREQUENCY 필드를 추가합니다. 유형에는 long을 사용하고 필요에 따라 null 유형을 사용합니다. 반복 빈도 데이터에 대한 자세한 내용은 [반복 빈도 데이터](#) 섹션을 참조하세요.

항목 상호 작용 데이터 세트 스키마 요구 사항(사용자 지정)

항목 상호 작용 데이터 세트는 사용자와 항목 간 상호 작용의 이력 및 실시간 데이터를 카탈로그에 저장합니다. Personalize에서 사용할 수 있는 상호 작용 데이터 유형에 대한 자세한 내용은 [항목 상호 작용 데이터 세트](#) 단원을 참조하세요.

각 상호작용에 제공하는 데이터는 스키마와 일치해야 합니다. 스키마에 따라 상호작용 메타데이터에는 비어 있음/null 값이 포함될 수 있습니다. 최소한 각 상호작용에 대해 다음을 제공해야 합니다.

- 사용자 ID
- 항목 ID
- 타임스탬프(Unix 에포크 타임 형식으로)

사용 사례와 데이터에 따라 추가 필드를 마음대로 추가할 수 있습니다. 필드가 필수 또는 예약으로 나열되어 있지 않고 데이터 유형이 [스키마 데이터 유형](#)에 나열된 경우, 필드 이름과 데이터 사용자가 선택할 수 있습니다.

항목 상호 작용 데이터 세트에 추가할 수 있는 선택형 메타데이터 필드의 최대 개수는 항목 상호 작용 데이터에 있는 개별 이벤트 유형의 총 개수와 합산하여 10개입니다. 이 수치에 포함되는 메타데이터 필드는 스키마에 추가한 사용자 지정 메타데이터 필드와 함께 EVENT_TYPE, EVENT_VALUE 필드입니다. 예약된 필드(예: IMPRESSION 를 제외한 메타데이터 필드의 최대 수는 5개입니다. 범주형 값은 최대 1000자까지 포함할 수 있습니다. 범주형 값이 1000자를 초과하는 상호작용이 있는 경우, 데이터세트 가져오기 작업이 실패합니다.

항목 상호 작용 데이터 세트에 대한 자세한 최소 요구 사항 및 최대 데이터 한도 정보는 [Service quotas](#) 섹션을 참조하세요.

상호작용 스키마 예제(사용자 지정)

다음 예제는 항목 상호 작용 데이터 세트의 스키마를 보여줍니다. USER_ID, ITEM_ID 및 TIMESTAMP 필드는 필수입니다. EVENT_TYPE, EVENT_VALUE 및 IMPRESSION 필드는 Personalize 에서 인식한 선택 가능한 예약어입니다. EVENT_TYPE 은 문자열이어야 하며 범주형일 수 없습니다. LOCATION 및 DEVICE 는 선택 가능한 컨텍스트 메타데이터 필드입니다. 스키마 요구 사항에 대한 자세한 내용은 [사용자 지정 데이터세트 및 스키마 요구 사항](#) 단원을 참조하세요.

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
```

```
    "name": "ITEM_ID",
    "type": "string"
  },
  {
    "name": "EVENT_TYPE",
    "type": "string"
  },
  {
    "name": "EVENT_VALUE",
    "type": [
      "float",
      "null"
    ]
  },
  {
    "name": "LOCATION",
    "type": "string",
    "categorical": true
  },
  {
    "name": "DEVICE",
    "type": [
      "string",
      "null"
    ],
    "categorical": true
  },
  {
    "name": "TIMESTAMP",
    "type": "long"
  },
  {
    "name": "IMPRESSION",
    "type": "string"
  }
],
"version": "1.0"
}
```

이 스키마의 경우, CSV 파일의 이력 데이터의 처음 몇 줄은 다음과 같을 수 있습니다. 참고로 EVENT_VALUE의 일부 값은 null입니다.

```
USER_ID, ITEM_ID, EVENT_TYPE, EVENT_VALUE, LOCATION, DEVICE, TIMESTAMP, IMPRESSION
```

```

35,73,click,,Ohio,Tablet,1586731606,73|70|17|95|96|92|55|45|16|97|56|54|33|94|36|10|5|
43|19|13|51|90|65|59|38
54,35,watch,0.75,Indiana,Cellphone,1586735164,35|82|78|57|20|63|1|90|76|75|49|71|26|24|
25|6|37|85|40|98|32|13|11|54|48
9,33,click,,Oregon,Cellphone,1586735158,68|33|62|6|15|57|45|24|78|89|90|40|26|91|66|31|
47|17|99|29|27|41|77|75|14
23,10,watch,0.25,California,Tablet,1586735697,92|89|36|10|39|77|4|27|79|18|83|16|28|68|
78|40|50|3|99|7|87|49|12|57|53
27,11,watch,0.55,Indiana,Tablet,1586735763,11|7|39|95|71|1|6|40|41|28|99|53|68|76|0|65|
69|36|22|42|34|67|24|20|66
...
...

```

사용자 데이터 세트 스키마 요구 사항(사용자 지정)

사용자 데이터세트는 사용자에게 대한 메타데이터를 저장합니다. 여기에는 각 항목에 대한 연령, 성별, 로열티 멤버십과 같은 정보가 포함될 수 있습니다. Personalize로 가져올 수 있는 사용자 데이터 유형에 대한 자세한 내용은 [사용자 데이터 세트](#) 단원을 참조하세요.

각 사용자에게 제공하는 데이터는 스키마와 일치해야 합니다. 최소한 각 사용자의 사용자 ID(최대 길이 256자)를 제공해야 합니다. 스키마에 따라 사용자 메타데이터에는 비어 있음/null 값이 포함될 수 있습니다. 사용자 스키마에는 최소 하나 이상의 메타데이터 필드가 있어야 하지만 null유형을 추가할 경우 사용자에게 대해 이 값이 null이 될 수 있습니다. 사용 사례와 데이터에 따라 추가 필드를 마음대로 추가할 수 있습니다. 필드가 필수 또는 예약으로 나열되어 있지 않고 데이터 유형이 [스키마 데이터 유형](#)에 나열된 경우, 필드 이름과 데이터 사용자가 선택할 수 있습니다.

범주형 데이터를 사용하려면 스키마에서 유형 string필드를 추가하고 필드의 범주형 속성을 true로 설정합니다. 그런 다음 대량 CSV 파일 및 개별 항목 가져오기에 범주형 데이터를 포함합니다. 여러 범주를 갖는 개인 사용자의 경우, 세로 막대 '|'를 사용하여 각 값을 구분합니다. 예를 들어, SUBSCRIPTION_MODEL 필드의 경우 사용자에게 대한 데이터는 학생|월별|할인이 될 수 있습니다.

범주형 값은 최대 1000자까지 포함할 수 있습니다. 범주형 값이 1000자를 초과하는 사용자가 있는 경우, 데이터세트 가져오기 작업이 실패합니다.

사용자 데이터세트에 대한 최소 요구 사항 및 최대 데이터 한도에 대한 자세한 내용은 [Service quotas](#) 단원을 참조하세요.

사용자 스키마 예제(사용자 지정)

다음 예제에서는 사용자 스키마를 조직하는 방법을 보여줍니다. USER_ID 필드는 필수 필드이고 AGE 및 GENDER 필드는 메타데이터입니다. 하나 이상의 메타데이터 필드가 필요하며 최대 25개의 메타

데이터 필드를 추가할 수 있습니다. 스키마 요구 사항에 대한 자세한 내용은 [사용자 지정 데이터세트 및 스키마 요구 사항](#) 단원을 참조하세요.

```
{
  "type": "record",
  "name": "Users",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "AGE",
      "type": "int"
    },
    {
      "name": "GENDER",
      "type": "string",
      "categorical": true
    }
  ],
  "version": "1.0"
}
```

이 스키마의 경우, CSV 파일의 이력 데이터의 처음 몇 줄은 다음과 같을 수 있습니다.

```
USER_ID,AGE,GENDER
5,34,Male
6,56,Female
8,65,Male
...
...
```

항목 데이터 세트 스키마 요구 사항(사용자 지정)

항목 데이터세트는 카탈로그의 항목에 대한 메타데이터를 저장합니다. 여기에는 각 항목의 가격, 장르, 재고 여부 등의 정보가 포함될 수 있습니다. Personalize로 가져올 수 있는 항목 데이터 유형에 대한 자세한 내용은 [항목 데이터 세트](#) 단원을 참조하세요.

각 항목에 제공하는 데이터는 항목 데이터세트 스키마와 일치해야 합니다. 최소한 각 항목의 항목 ID(최대 길이 256자)를 제공해야 합니다. 스키마에 따라 항목 메타데이터에는 비어 있음/null 값이 포함

될 수 있습니다. 스키마에는 최소 하나 이상의 메타데이터 필드가 있어야 하지만 null 유형을 추가하면 항목에 대해 이 값이 null이 될 수 있습니다. 사용 사례와 데이터에 따라 추가 필드를 마음대로 추가할 수 있습니다. 필드가 필수 또는 예약으로 나열되어 있지 않고 데이터 유형이 [스키마 데이터 유형](#)에 나열된 경우, 필드 이름과 데이터 사용자가 선택할 수 있습니다.

범주형 데이터를 사용하려면 스키마에서 유형 string 필드를 추가하고 필드의 범주형 속성을 true로 설정합니다. 그런 다음 대량 CSV 파일 및 개별 항목 가져오기에 범주형 데이터를 포함합니다. 범주형 값은 최대 1000자까지 포함할 수 있습니다. 범주형 값이 1000자를 초과하는 항목이 있는 경우, 데이터 세트 가져오기 작업이 실패합니다.

여러 범주를 갖는 항목의 경우, 세로 막대 '|'로 각 값을 구분합니다. 예를 들어, GENRES 필드의 경우 항목에 대한 데이터가 Action|Crime|Biopic이 될 수 있습니다. 여러 수준의 범주형 데이터가 있고 일부 항목에 계층의 각 수준에 대해 여러 범주가 있는 경우, 각 수준에 필드를 추가하고 각 필드 이름 뒤에 GENRES, GENRE_L2, GENRE_L3 등의 수준 지표를 추가합니다. 이렇게 하면 항목이 여러 멀티 레벨 범주에 속하더라도 하위 범주를 기준으로 추천을 필터링할 수 있습니다(필터 생성 및 사용에 대한 자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 단원을 참조하세요). 예를 들어, 동영상에는 각 범주 수준에 대해 다음과 같은 데이터가 있을 수 있습니다.

- GENRES: 액션|어드벤처
- GENRE_L2: 범죄|서부영화
- GENRE_L3: 일대기 영화

이 예제에서는 동영상이 액션 > 범죄 > 일대기 영화 계층 및 모험 > 서부 영화 > 일대기 영화 계층 구조로 되어 있습니다. L3까지만 사용하는 것이 좋지만, 필요한 경우 더 많은 레벨을 사용할 수 있습니다.

모델 교육 중에 Personalize는 최대 750,000개의 항목을 고려합니다. 750,000개가 넘는 항목을 가져오는 경우, Personalize는 새 항목(상호작용 없이 최근 추가한 항목)과 최근 상호작용 데이터가 있는 기존 항목을 포함시키는 것에 중점을 두고 교육에 포함할 항목을 결정합니다.

항목 데이터세트에 대한 최소 요구 사항 및 최대 데이터 한도에 대한 자세한 내용은 [Service quotas](#) 단원을 참조하세요.

항목 데이터세트 스키마 예제(사용자 지정)

다음 예제에서는 항목 스키마를 조직하는 방법을 보여줍니다. ITEM_ID 필드는 필수 사항입니다. GENRE 필드는 범주형 메타데이터이고 DESCRIPTION 필드는 텍스트 메타데이터입니다. 최소 한 개 이상의 메타데이터 필드가 필요합니다. 최대 100개의 메타데이터 필드를 추가할 수 있습니다. CREATION_TIMESTAMP 필드는 예약어입니다. 스키마 요구 사항에 대한 자세한 내용은 [사용자 지정 데이터세트 및 스키마 요구 사항](#) 단원을 참조하세요.

```
{
  "type": "record",
  "name": "Items",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "GENRES",
      "type": [
        "null",
        "string"
      ],
      "categorical": true
    },
    {
      "name": "CREATION_TIMESTAMP",
      "type": "long"
    },
    {
      "name": "DESCRIPTION",
      "type": [
        "null",
        "string"
      ],
      "textual": true
    }
  ],
  "version": "1.0"
}
```

이 스키마의 경우, CSV 파일의 이력 데이터의 처음 몇 줄은 다음과 같을 수 있습니다.

```
ITEM_ID,GENRES,CREATION_TIMESTAMP,DESCRIPTION
1,Adventure|Animation|Children|Comedy|Fantasy,1570003267,"This is an animated movie
that features action, comedy, and fantasy. Audience is children. This movie was
released in 2004."
2,Adventure|Children|Fantasy,1571730101,"This is an adventure movie with elements of
fantasy. Audience is children. This movie was release in 2010."
3,Comedy|Romance,1560515629,"This is a romantic comedy. The movie was released in 1999.
Audience is young women."
```



```
4,Comedy|Drama|Romance,1581670067,"This movie includes elements of both comedy and
drama as well as romance. This movie was released in 2020."
```

```
...
...
```

작업 데이터 세트 스키마 요구 사항(사용자 지정)

작업은 고객에게 추천하고 싶은 참여 활동입니다. 작업에는 모바일 앱 설치, 멤버십 프로필 작성, 로열티 프로그램 가입, 프로모션 이메일 구독 등이 포함될 수 있습니다. 작업 데이터 세트는 사용자 행동에 대한 데이터를 저장합니다. Amazon Personalize로 가져올 수 있는 작업 데이터 유형에 대한 내용은 [작업 데이터 세트](#) 섹션을 참조하세요.

각 작업에 제공하는 데이터는 작업 데이터 세트 스키마와 일치해야 합니다. 스키마에 따라 작업 메타데이터에는 비어 있음/null 값이 포함될 수 있습니다.

최소한 각 항목에 대한 작업 ID를 제공해야 합니다 (최대 길이 256자). 스키마에는 최소 하나 이상의 메타데이터 필드가 있어야 하지만 null 유형을 추가하면 작업에 대해 이 값이 null이 될 수 있습니다. 사용 사례와 데이터에 따라 별도의 필드를 추가할 수 있습니다. 필드가 필수 또는 예약된 상태로 나열되고 데이터 유형이 [스키마 데이터 유형](#)에 나열되지 않는 한, 필드 이름과 데이터 유형을 선택할 수 있습니다.

범주형 필드를 추가하려면 유형의 string 필드를 추가하고 스키마에서 필드의 범주형 속성을 로 설정합니다. true 그런 다음 대량 CSV 파일 및 개별 작업 가져오기에 범주형 데이터를 포함합니다. 범주형 값은 최대 1000자까지 포함할 수 있습니다. 범주형 값이 1,000자를 초과하는 작업이 있는 경우 데이터 세트 가져오기 작업이 실패합니다.

작업 데이터 세트에 대한 자세한 최소 요구 사항 및 최대 데이터 한도 정보는 [Service quotas](#) 섹션을 참조하세요.

작업 데이터 세트 스키마 예제(사용자 지정)

다음 예제에서는 작업 스키마를 구성하는 방법을 보여줍니다. ACTION_ID 필드는 필수 사항입니다. MEMBERSHIP_LEVEL 필드는 범주형 문자열 필드입니다. VALUE, CREATION_TIMESTAMP, REPEAT_FREQUENCY 필드는 필수 유형이 포함된 예약된 키워드입니다. 최대 10개의 열을 추가할 수 있습니다. 스키마 요구 사항에 대한 자세한 내용은 [사용자 지정 데이터세트 및 스키마 요구 사항](#) 단원을 참조하세요.

```
{
  "type": "record",
  "name": "Actions",
```

```
"namespace": "com.amazonaws.personalize.schema",
"fields": [
  {
    "name": "ACTION_ID",
    "type": "string"
  },
  {
    "name": "VALUE",
    "type": [
      "null",
      "long"
    ]
  },
  {
    "name": "MEMBERSHIP_LEVEL",
    "type": [
      "null",
      "string"
    ],
    "categorical": true
  },
  {
    "name": "CREATION_TIMESTAMP",
    "type": "long"
  },
  {
    "name": "REPEAT_FREQUENCY",
    "type": [
      "long",
      "null"
    ]
  }
],
"version": "1.0"
}
```

이 스키마의 경우, CSV 파일의 이력 데이터의 처음 몇 줄은 다음과 같을 수 있습니다.

```
ACTION_ID,VALUE,MEMBERSHIP_LEVEL,CREATION_TIMESTAMP,REPEAT_FREQUENCY
1,10,Deluxe|Premium,1510003267,7
2,5,Basic,1580003267,7
```

```
3,5,Preview,1590003267,3
4,10,Deluxe|Platinum,1560003267,4
...
...
```

작업 상호 작용 데이터 세트 스키마 요구 사항(사용자 지정)

작업 상호 작용 데이터 세트는 사용자와 작업 간 상호 작용의 이력 및 실시간 데이터를 작업 데이터 세트에 저장합니다. Amazon Personalize에서 사용할 수 있는 데이터 유형에 대한 내용은 [작업 상호 작용 데이터 세트](#) 섹션을 참조하세요.

각 상호작용에 제공하는 데이터는 스키마와 일치해야 합니다. 스키마에 따라 상호작용 메타데이터에는 비어 있음/null 값이 포함될 수 있습니다. 스키마에는 최소한 다음이 포함되어야 합니다.

- USER_ID
- ACTION_ID
- TIMESTAMP
- EVENT_TYPE

사용 사례와 데이터에 따라 별도의 필드를 추가할 수 있습니다. 필드가 필수 또는 예약된 상태로 나열되고 데이터 유형이 [스키마 데이터 유형](#)에 나열되지 않는 한, 필드 이름과 데이터 유형을 선택할 수 있습니다.

작업 상호 작용 데이터 세트에 대한 자세한 최소 요구 사항 및 최대 데이터 한도 정보는 [Service quotas](#) 섹션을 참조하세요.

작업 상호 작용 데이터 세트 스키마 예제(사용자 지정)

다음 예제에서는 필수 필드만 있는 작업 상호 작용 데이터 세트에 대한 스키마를 보여줍니다. 일반적인 스키마 형식 요구 사항에 대한 자세한 내용은 [스키마 서식 지정 요구 사항](#) 섹션을 참조하세요.

```
{
  "type": "record",
  "name": "ActionInteractions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    }
  ]
}
```

```

    },
    {
      "name": "ACTION_ID",
      "type": "string"
    },
    {
      "name": "EVENT_TYPE",
      "type": "string"
    },
    {
      "name": "TIMESTAMP",
      "type": "long"
    }
  ],
  "version": "1.0"
}

```

이 스키마의 경우, CSV 파일의 이력 데이터의 처음 몇 줄은 다음과 같을 수 있습니다. 참고로 IMPRESSION의 일부 값은 null입니다.

```

USER_ID,ACTION_ID,EVENT_TYPE,TIMESTAMP
35,73,Viewed,1586731606
54,35,Not taken,1586731609
9,33,Viewed,1586735158
23,10,Taken,1586735697
27,11,Taken,1586735763
...
...

```

SDK for Python(Boto3)을 이용한 스키마 만들기

1. 사용하려는 Avro 형식 스키마를 정의합니다.
2. 기본 Python 폴더의 JSON 파일에 스키마를 저장합니다.
3. 다음 코드를 사용하여 스키마를 생성합니다.

```

import boto3

personalize = boto3.client('personalize')

with open('schema.json') as f:
    createSchemaResponse = personalize.create_schema(

```

```

        name = 'YourSchema'
        schema = f.read()
    )

    schema_arn = createSchemaResponse['schemaArn']

    print('Schema ARN:' + schema_arn )

```

4. Personalize는 새 스키마의 ARN을 반환합니다. 나중에 사용하기 위해 이 ARN을 저장합니다.

Personalize는 스키마를 관리하기 위한 작업을 제공합니다. 예를 들어, [ListSchemasAPI](#)를 사용하면 사용 가능한 스키마 목록을 가져올 수 있습니다.

스키마를 생성한 후 이 스키마를 그와 일치하는 데이터세트와 함께 사용하십시오. 자세한 내용은 [데이터 형식 지침](#) 단원을 참조하세요.

데이터 형식 지침

Personalize 데이터세트로 데이터를 가져올 때 레코드를 대량으로 가져오거나 개별적으로 가져오거나 두 가지 방법으로 모두 가져오도록 선택할 수 있습니다.

- 대량 가져오기에는 많은 수의 이력 레코드를 한 번에 가져오는 작업이 포함됩니다. Data Wrangler 및 여러 데이터 원본을 사용하여 대량 SageMaker 데이터를 준비하고 가져올 수 있습니다. 또는 직접 대량 데이터를 준비하여 S3의 CSV 파일에서 Personalize로 직접 가져올 수도 있습니다.
- 개별 가져오기의 경우 Personalize 콘솔 및 API 작업을 통해 개별 레코드를 가져옵니다. 또는 라이브 이벤트의 상호작용 데이터를 실시간으로 스트리밍할 수 있습니다. 가져오기 옵션에 대한 자세한 내용은 [개별 레코드 가져오기](#) 단원을 참조하세요.

대량 데이터를 가져오기 전에 형식이 올바른지 확인합니다. 다음 단원은 대량 데이터의 형식을 지정하는 데 도움이 될 수 있습니다. 데이터 형식을 지정하는 방법을 잘 모르겠으면 Amazon 데이터 랭글러 (SageMaker Data Wrangler) 를 사용하여 데이터를 준비할 수 있습니다. 자세한 정보는 [Amazon SageMaker 데이터 랭글러를 사용하여 데이터 준비 및 가져오기](#)을 참조하세요.

주제

- [대량 데이터 형식 지침 및 요구 사항](#)
- [상호작용 데이터 예제](#)
- [명시적 노출의 형식 지정](#)
- [범주형 데이터의 형식 지정](#)

대량 데이터 형식 지침 및 요구 사항

다음 지침 및 요구 사항은 대량 데이터의 형식을 올바르게 지정하는 데 도움이 될 수 있습니다.

- 입력 데이터는 CSV(쉼표로 구분된 값) 파일에 있어야만 합니다.
- CSV 파일의 첫 번째 행에는 열 헤더가 포함되어야만 합니다. 헤더를 따옴표(")로 묶지 않습니다.
- 데이터세트 유형의 필수 필드가 있는지 확인하고, 해당 이름이 Personalize 요구 사항과 일치하는지 확인합니다. 예를 들어, 항목 데이터에는 각 항목의 ID가 포함된 ITEM_IDENTIFICATION_NUMBER로 불리는 열이 있을 수 있습니다. 이 열을 ITEM_ID 필드로 사용하려면 열 이름을 ITEM_ID로 바꾸세요. Data Wrangler를 사용하여 데이터 형식을 지정하는 경우, Personalize Data Wrangler 변환용 맵 열을 사용하여 열의 이름이 올바르게 지정되었는지 확인할 수 있습니다.

필수 필드에 대한 자세한 내용은 [스키마](#) 단원을 참조하세요. 데이터를 준비하기 위하여 Data Wrangler를 사용하는 것에 대한 자세한 내용은 [Amazon SageMaker 데이터 랭글러를 사용하여 데이터 준비 및 가져오기](#) 단원을 참조하세요.

- CSV 파일의 열 헤더 이름은 스키마에 매핑되어야만 합니다.
- CSV 파일의 각 레코드는 하나의 줄에 있어야만 합니다.
- 각 열의 데이터 유형은 스키마에 매핑되어야만 합니다. Data Wrangler를 사용하여 데이터의 형식을 지정한 경우, Data Wrangler 변환 [값을 유형으로 파싱](#)을 사용하여 데이터 유형을 변환할 수 있습니다.
- TIMESTAMP 및 CREATION_TIMESTAMP 데이터는 UNIX epoch 시간 형식이어야 합니다. 자세한 정보는 [타임스탬프 데이터](#)을 참조하세요.
- 항목 ID, 사용자 ID 및 작업 ID 데이터에 문자나 특수 " 문자를 포함하지 마십시오.
- 데이터에 ASCII로 인코딩되지 않은 문자가 포함된 경우, CSV 파일은 UTF-8 형식으로 인코딩되어야만 합니다.
- [비정형 텍스트 메타데이터](#)에 설명된 대로 모든 텍스트 데이터의 형식을 지정해야만 합니다.
- [명시적 노출의 형식 지정](#) 및 [범주형 데이터의 형식 지정](#)에서 설명된 대로 노출 수 데이터와 범주형 데이터에 형식을 지정해야만 합니다.

상호작용 데이터 예제

다음과 같은 상호작용 데이터는 영화 티켓을 판매하는 웹사이트에서의 과거 사용자 활동을 나타냅니다. 이 데이터를 사용하여 사용자의 상호작용 데이터를 기반으로 영화 추천을 제공하는 모델을 학습시킬 수 있습니다.

```
USER_ID,ITEM_ID,EVENT_TYPE,EVENT_VALUE,TIMESTAMP
196,242,click,15,881250949
186,302,click,13,891717742
22,377,click,10,878887116
244,51,click,20,880606923
166,346,click,10,886397596
298,474,click,40,884182806
115,265,click,20,881171488
253,465,click,50,891628467
305,451,click,30,886324817
```

다음은 관련 상호작용 스키마입니다.

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    { "name": "EVENT_TYPE",
      "type": "string"
    },
    {
      "name": "EVENT_VALUE",
      "type": "float"
    },
    {
      "name": "TIMESTAMP",
      "type": "long"
    }
  ],
  "version": "1.0"
}
```

Personalize에는 USER_ID, ITEM_ID 및 TIMESTAMP 필드가 필요합니다. USER_ID는 애플리케이션의 사용자 식별자입니다. ITEM_ID는 영화의 식별자입니다. EVENT_TYPE 및 EVENT_VALUE은 사용자 활동의 식별자입니다. 이 샘플 데이터에서 click은 영화 구입 이벤트를, 15는 영화의 구입 가격을 나타낼 수 있습니다. TIMESTAMP는 영화 구입이 발생한 Unix epoch 시간을 나타냅니다.

타임스탬프 데이터

TIMESTAMP(항목 상호 작용 데이터 세트용) 또는 CREATION_TIMESTAMP(항목 데이터 세트용) 데이터와 같은 타임스탬프 데이터는 Unix epoch 시간 형식(초)이어야 합니다. 예를 들어, 2020년 7월 31일 날짜의 에포크 타임스탬프(초)는 1596238243입니다. 날짜를 Unix epoch 타임스탬프로 변환하려면 [Epoch 변환기 - Unix 타임스탬프 변환기](#)를 사용합니다.

명시적 노출의 형식 지정

[사용자-개인 맞춤](#) 레시피를 사용하면 노출 데이터를 기록하고 업로드할 수 있습니다. 노출은 사용자가 특정 항목과 상호작용(예: 클릭 또는 시청)할 때 사용자가 볼 수 있었던 항목의 목록입니다. 대량 데이터 가져오기에서 노출 데이터를 업로드하려면 각 항목 ID를 수동으로 기록합니다. 과거 상호작용 데이터의 일부로 값을 세로 막대('|') 문자로 구분해야 합니다. 노출 데이터의 경우 세로 막대 문자도 1000자 한도에 포함됩니다. 노출 데이터 준비에 대한 자세한 내용은 [노출 데이터](#) 단원을 참조하세요.

다음은 IMPRESSION 열에 명시적 노출이 포함된 항목 상호 작용 데이터 세트에서 발췌한 것입니다.

EVENT_TYPE	IMPRESSION	ITEM_ID	TIMESTAMP	USER_ID
click	73 70 17 95 96	73	1586731606	USER_1
click	35 82 78 57 20 63 1 90 76 75 49 71 26 24 25 6	35	1586735164	USER_2
...

애플리케이션이 사용자 USER_1 항목 73, 70, 17, 95 및 96을 보여 주었고 사용자가 최종적으로 항목 73을 선택했습니다. 이 데이터를 기반으로 새 솔루션 버전을 만들면 사용자 USER_1에게 70, 17, 95 및 96 항목이 추천되는 빈도가 줄어듭니다.

범주형 데이터의 형식 지정

범주형 문자열 데이터를 사용할 때 단일 항목에 여러 범주를 포함시키려면 세로 막대 '|' 문자를 사용하여 값을 구분합니다. 예를 들어, 범주가 두 개 있는 항목의 경우 데이터 행은 다음과 비슷할 것입니다.

```
ITEM_ID,GENRE  
item_123,horror|comedy
```

데이터의 형식을 지정한 후 Personalize로 가져올 수 있도록 S3 버킷에 업로드합니다. 자세한 내용은 [S3 버킷에 업로드](#) 단원을 참조하세요.

도메인 사용 사례 및 사용자 지정 레시피

Amazon Personalize는 학습 모델을 위한 다양한 도메인 사용 사례와 사용자 지정 레시피를 제공합니다.

- 도메인 데이터 세트 그룹에서 추천자를 생성할 때 사용 사례를 지정합니다. Amazon Personalize는 사용 사례에 가장 적합한 구성으로 추천자를 지원하는 모델을 학습시킵니다.
- 사용자 지정 데이터 세트 그룹 또는 도메인 데이터 세트 그룹에서 사용자 지정 솔루션을 생성할 때는 레시피를 지정하고 학습 파라미터를 구성합니다. 솔루션용 솔루션 버전을 생성하면 Amazon Personalize는 레시피 및 학습 구성을 기반으로 솔루션 버전을 지원하는 모델을 학습시킵니다.

주제

- [사용 사례 및 레시피 기능](#)
- [사용 사례 선택](#)
- [레시피 선택](#)

사용 사례 및 레시피 기능

일부 사용 사례 및 레시피의 경우 Personalize는 다음 기능을 사용하여 관련성이 더 높은 추천을 생성하고 항목 검색 및 참여를 개선합니다.

주제

- [실시간 개인 맞춤](#)
- [탐색](#)
- [자동 업데이트](#)

실시간 개인 맞춤

일부 사용 사례 및 레시피의 경우 Personalize는 실시간 개인 맞춤을 사용하여 사용자의 변화하는 관심에 따라 추천을 업데이트하고 조정합니다. 마지막 전체 교육에서 제공된 항목 또는 작업과의 상호 작용을 기록하면 사용자를 위한 추천을 업데이트합니다. 이러한 상호 작용을 이벤트 추적기 및 [PutEvents](#) 작업을 사용하여 기록하거나, 작업과의 상호 작용의 PutActionInteractions 경우 작업을 사용하여 기록합니다.

이벤트 기록에 대한 자세한 내용은 [이벤트 기록](#) 단원을 참조하세요. 새로운 데이터가 실시간 개인 맞춤을 포함하여 실시간 추천에 영향을 미치는 방식에 대한 자세한 내용은 [새 데이터가 실시간 추천에 영향을 미치는 방식](#) 단원을 참조하세요.

실시간 개인 맞춤을 지원하는 사용 사례 및 레시피는 다음과 같습니다.

- [추천 제품\(ECOMMERCE 사용 사례\)](#)
- [가장 적합한 추천 제품\(VIDEO_ON_DEMAND 사용 사례\)](#)
- [사용자 맞춤화-v2 레시피](#)
- [사용자-개인 맞춤 레시피](#)
- [맞춤형 랭킹-v2 레시피](#)
- [개인 맞춤형-순위 레시피](#)
- [차선책-작업 레시피](#)

탐색

일부 도메인 사용 사례 및 사용자 지정 레시피의 경우 Amazon Personalize는 항목을 추천할 때 탐색을 사용합니다. 탐색을 사용하면 추천에 새 항목 또는 작업, 상호 작용이 거의 없는 항목 또는 작업, 이전 행동을 기준으로 사용자와 관련성이 낮은 항목 또는 작업 등 일반적으로 사용자에게 추천될 가능성이 낮은 일부 항목 또는 작업이 포함됩니다. 이렇게 하면 카탈로그가 빠르게 변경되거나 뉴스 기사나 홍보와 같은 새 항목이 새로 만들어졌기 때문에 사용자에게 더 관련성이 높은 항목을 발견하고 참여도를 높일 수 있습니다.

탐색 구성

사용자 개인화-v2 레시피를 사용하는 경우 Amazon Personalize는 탐색 구성을 대신 처리하며 탐색을 통해 포함된 항목은 권장 사항 응답에 포함됩니다. Exploration Reason 새 항목이 권장 사항에 포함되도록 프로모션 필터를 사용하여 생성 타임스탬프를 기반으로 새 항목을 홍보할 수 있습니다. 프로모션에 대한 자세한 내용은 [참조의 항목 홍보](#)를 참조하십시오.

탐색을 사용하는 다른 모든 사용 사례 또는 레시피의 경우, 추천 또는 사용자 지정 캠페인을 만들 때 또는 일괄 추천 작업(사용자 지정 리소스)을 만들 때 다음 필드를 사용하여 탐색을 구성할 수 있습니다.

- 관련성이 낮은 항목 탐색에 주목(탐색 가중치) — 탐색 범위를 설정합니다. 0과 1 사이에서 소수 값을 지정합니다. 기본값은 0.3입니다. 값이 1에 가까울수록 탐색이 더 많아집니다. 더 많이 탐색할수록 항목 상호 작용 데이터가 적거나 이전 행동을 기반으로 한 관련성이 낮은 항목이 더 많이 추천에 포함됩니다. 0이면 탐색이 수행되지 않으며 최신 데이터(관련성)를 기반으로 추천이 제공됩니다.

- 탐색 항목 기간 제한 – 항목 상호 작용 데이터 세트의 모든 항목에서 최근 상호 작용이 발생한 이후 최대 항목 사용 기간(일)을 지정합니다. 이렇게 하면 항목 기간에 따른 항목 탐색 범위가 정의됩니다. Amazon Personalize는 생성 타임스탬프 또는 생성 타임스탬프 데이터가 누락된 경우 항목 상호 작용 데이터를 기반으로 항목 기간을 결정합니다. Personalize가 항목 기간을 결정하는 방법에 대한 자세한 내용은 [생성 타임스탬프 데이터](#) 단원을 참조하세요.

Personalize가 탐색 중에 고려하는 항목을 늘리려면 더 큰 값을 입력합니다. 최소값은 1일이고, 기본값은 30일입니다. 지정한 항목 보관 기간 한도보다 오래된 항목이 추천에 포함될 수 있습니다. 이는 이러한 항목이 사용자와 관련이 있고 탐색에서 해당 항목을 식별할 수 없었기 때문입니다.

탐색을 사용하는 사용 사례 및 레시피

탐색을 사용하는 각 사용 사례 또는 레시피에 대한 자세한 내용은 다음을 참조하세요.

- [추천 제품\(ECOMMERCE 사용 사례\)](#)
- [가장 적합한 추천 제품\(VIDEO_ON_DEMAND 사용 사례\)](#)
- [사용자 맞춤화-v2 레시피](#)
- [사용자-개인 맞춤 레시피](#)
- [차선책-작업 레시피](#)

자동 업데이트

일부 사용 사례 및 사용자 지정 레시피의 경우 Amazon Personalize는 추천자 또는 솔루션 버전을 자동으로 업데이트하여 추천을 위한 새 항목이나 조치를 고려합니다. 자동 업데이트에는 비용이 들지 않습니다. 자동 업데이트가 포함된 사용 사례 및 레시피 목록은 [참조하십시오. 도메인 사용 사례 및 자동 업데이트가 포함된 맞춤형 레시피](#)

자동 업데이트는 다음과 같이 작동합니다.

- Amazon Personalize가 솔루션 버전 또는 추천자를 자동으로 업데이트하는 시기는 권장 사항을 받는 방법에 따라 달라집니다.
 - 실시간 추천을 위해 Amazon Personalize는 솔루션 버전 또는 추천자를 2시간마다 업데이트합니다.
 - 배치 항목 추천의 경우, 배치 추론 작업을 생성하고 솔루션에 대해 전체 학습된 최신 솔루션 버전을 지정하면 Personalize는 탐색 중에 새 항목을 고려하도록 솔루션 버전을 자동으로 업데이트합니다. 최신 솔루션 버전을 지정하지 않으면 업데이트는 발생하지 않습니다.

- Amazon Personalize는 업데이트할 때마다 를 사용하여 권장 사항에 새 항목을 포함하기 시작합니다. [탐색](#) 새 항목 또는 작업을 고려할 때 Amazon Personalize는 해당 항목에 대한 모든 메타데이터를 고려합니다. 그러나 이 데이터는 항목에 대한 상호 작용을 기록하고 완전히 재교육한 후에만 권장 사항에 더 큰 영향을 미칩니다.
- 업데이트를 적용하려면 마지막 자동 업데이트 또는 재교육 이후 새로운 동작, 항목 또는 상호 작용 데이터를 제공해야 합니다.
- Amazon Personalize는 75만 개의 아이템을 가져올 때까지 새 아이템을 고려합니다. 이는 교육 중에 고려되는 최대 품목 수입입니다.

사용자 지정 리소스에 대한 추가 지침 및 요구 사항

사용자 지정 리소스를 사용하는 경우 자동 업데이트에 대한 지침 및 요구 사항은 다음과 같습니다.

- 솔루션 버전은 캠페인에서 배포해야 합니다. 캠페인에서 업데이트된 솔루션 버전을 자동으로 사용합니다.
- 자동 업데이트는 자동 학습과 다릅니다. 자동 업데이트는 완전히 새로운 솔루션 버전을 만들지 않습니다. 그리고 모델은 최신 데이터를 통해 학습하지 않습니다. 솔루션을 유지 관리하려면 자동 훈련 빈도가 여전히 1주 이상이어야 합니다.
- 솔루션이 새 솔루션 버전을 자동으로 생성하거나 새 버전을 수동으로 생성한 후에는 캠페인에 배포했더라도 Amazon Personalize는 이전 솔루션 버전을 자동으로 업데이트하지 않습니다.
- 2시간마다 빈도가 충분하지 않은 경우 User-Personalization을 사용하여 권장 사항에 새 항목을 UPDATE 포함하도록 trainingMode 설정한 상태로 솔루션 버전을 수동으로 생성할 수 있습니다. Personalize는 전체 학습된 최신 솔루션 버전만 자동으로 업데이트한다는 점을 명심하세요. 수동으로 업데이트된 솔루션 버전은 향후에 자동으로 업데이트되지 않습니다. 솔루션에서 자동 교육을 사용하는 경우 다음 솔루션 버전에서 자동 업데이트가 다시 시작됩니다. 그렇지 않은 경우 교육 모드를 FULL 설정한 상태에서 새 솔루션을 수동으로 만들어 캠페인에 배포하세요.

도메인 사용 사례 및 자동 업데이트가 포함된 맞춤형 레시피

자동 업데이트를 사용하는 각 사용 사례 또는 레시피에 대한 자세한 내용은 다음을 참조하세요.

- [추천 제품\(ECOMMERCE 사용 사례\)](#)
- [가장 적합한 추천 제품\(VIDEO_ON_DEMAND 사용 사례\)](#)
- [사용자 맞춤화-v2 레시피](#)
- [사용자-개인 맞춤 레시피](#)

- [차선책-작업 레시피](#)

사용 사례 선택

도메인 데이터 세트 그룹에서 추천을 생성할 때 사용 사례를 지정합니다. Amazon Personalize는 사용 사례에 가장 적합한 구성으로 추천을 지원하는 모델을 교육합니다. 도메인마다 사용 사례가 다릅니다. 예를 들어, 도메인 데이터 세트 그룹에 VIDEO_ON_DEMAND를 지정하는 경우 VIDEO_ON_DEMAND 사용 사례만 사용할 수 있습니다. 사용 사례마다 권장 사항을 받기 위한 요구 사항이 다릅니다. 일부 사용 사례에는 특정 이벤트 유형이 필요합니다. 추가 이벤트 유형을 자유롭게 포함할 수 있습니다.

모든 사용 사례의 경우 상호작용 데이터는 다음을 가져야만 합니다.

- 카탈로그의 항목과 상호 작용하는 사용자의 1,000개 이상의 항목 상호 작용 기록. 이러한 상호 작용은 대량 가져오기, 스트리밍된 이벤트 또는 이 둘 다에서 발생할 수 있습니다.
- 각각 2번 이상의 항목 상호 작용을 갖는 25개 이상의 고유 사용자 ID.

품질 추천의 경우, 각 2번 이상의 항목 상호 작용을 갖는 1,000명 이상의 사용자로부터 50,000건 이상의 항목 상호 작용을 갖는 것이 좋습니다.

주제

- [VIDEO_ON_DEMAND 사용 사례](#)
- [ECOMMERCE 사용 사례](#)

VIDEO_ON_DEMAND 사용 사례

다음 단원에는 각 VIDEO_ON_DEMAND 사용 사례에 대한 요구 사항 및 Amazon 리소스 이름(ARN)이 나와 있습니다. 모든 사용 사례의 경우 상호작용 데이터는 다음을 갖고 있어야 합니다.

- 카탈로그의 항목과 상호 작용하는 사용자의 1,000개 이상의 항목 상호 작용 기록. 이러한 상호 작용은 대량 가져오기, 스트리밍된 이벤트 또는 이 둘 다에서 발생할 수 있습니다.
- 각각 2번 이상의 항목 상호 작용을 갖는 25개 이상의 고유 사용자 ID.

품질 추천의 경우, 각 2번 이상의 항목 상호 작용을 갖는 1,000명 이상의 사용자로부터 50,000건 이상의 항목 상호 작용을 갖는 것이 좋습니다.

Note

[CreateRecommender](#) API를 사용하는 경우 레시피 ARN에 대해 여기에 나열되어 있는 ARN을 제공합니다.

주제

- [X를 시청하셨기 때문입니다](#)
- [X와 유사한 제품](#)
- [가장 인기 있음](#)
- [지금 유행](#)
- [가장 적합한 추천 제품](#)

X를 시청하셨기 때문입니다

지정한 동영상을 기반으로 다른 사용자도 시청한 동영상을 추천 받을 수 있습니다. 이 사용 사례에서 Amazon Personalize는 지정한 사용자 ID 및 Watch 이벤트를 기반으로 사용자가 시청한 동영상을 자동으로 필터링합니다. 자체 필터를 적용하는 경우 사용자가 시청한 동영상이 필터링으로 걸러진 후에 필터가 적용됩니다.

Amazon Personalize는 필터링할 때 이벤트 유형별로 사용자당 최대 100개의 항목 상호 작용을 고려합니다. 이는 모든 자동 또는 사용자 지정 필터에 적용됩니다. 이 한도의 증가를 요청하려면 [Service Quotas 콘솔](#)을 사용하면 됩니다. 자세한 내용은 Service Quotas 사용 설명서의 [할당량 증가 요청](#) 단원을 참조하세요.

- 레시피 ARN: `arn:aws:personalize:::recipe/aws-vod-because-you-watched-x`
- GetRecommendations API 요구 사항:

`userId`: 필수

`itemId`: 필수

- 교육 시 사용되는 데이터 세트: 항목 상호 작용 데이터 세트 전용(필수)
- 필수 이벤트 유형: Watch 이벤트가 최소 1000개 이상.

X와 유사한 제품

지정한 동영상과 비슷한 동영상을 추천 받습니다. 이 사용 사례에서 Amazon Personalize는 지정한 사용자 ID 및 Watch 이벤트를 기반으로 사용자가 시청한 동영상을 자동으로 필터링합니다. 자체 필터를 적용하는 경우 사용자가 시청한 동영상이 필터링으로 걸러진 후에 필터가 적용됩니다.

Amazon Personalize는 필터링할 때 이벤트 유형별로 사용자당 최대 100개의 항목 상호 작용을 고려합니다. 이는 모든 자동 또는 사용자 지정 필터에 적용됩니다. 이 한도의 증가를 요청하려면 [Service Quotas 콘솔](#)을 사용하면 됩니다. 자세한 내용은 Service Quotas 사용 설명서의 [할당량 증가 요청](#) 단원을 참조하세요.

- 레시피 ARN: `arn:aws:personalize:::recipe/aws-vod-more-like-x`
- GetRecommendations API 요구 사항:

`userId`: 필수

`itemId`: 필수

- 학습 시 사용되는 데이터 세트:
 - 상호작용(필수)
 - 항목(필수)
- 필요한 이벤트 수: 모든 유형의 이벤트 최소 1000개 이상.
- 추천 이벤트 유형: Watch 및 Click 이벤트.

가장 인기 있음

가장 많은 사용자가 시청한 동영상을 추천 받습니다.

- 레시피 ARN: `arn:aws:personalize:::recipe/aws-vod-most-popular`
- GetRecommendations요구 사항:

`userId`: 필수

`itemId`: 사용되지 않음

- 교육 시 사용되는 데이터 세트: 항목 상호 작용 데이터 세트 전용(필수)
- 필수 이벤트 유형: Watch 이벤트가 최소 1000개 이상.

지금 유행

현재 유행하고 있는 동영상을 추천 받습니다. 지금 유행하는 동영상은 사용자에게 빠르게 인기를 얻고 있는 항목입니다. Amazon Personalize는 2시간마다 자동으로 상호작용 데이터를 평가하고 유행하는 항목을 식별합니다.

- 레시피 ARN: `arn:aws:personalize:::recipe/aws-vod-trending-now`
- GetRecommendations API 요구 사항:

`userId`: 사용자가 상호작용한 항목을 기준으로 `CurrentUser` 또는 항목별로 필터링하는 경우에만 필수

`itemId`: 사용되지 않음

- 교육 시 사용되는 데이터 세트: 항목 상호 작용 데이터 세트 전용(필수)
- 필요한 이벤트 수: 모든 유형의 이벤트 최소 1000개 이상.

가장 적합한 추천 제품

지정한 사용자를 위한 개인 맞춤형 콘텐츠 추천을 받습니다. 이 사용 사례에서 Amazon Personalize는 지정한 사용자 ID 및 Watch 이벤트를 기반으로 사용자가 시청한 동영상을 자동으로 필터링합니다. 자체 필터를 적용하는 경우 사용자가 시청한 동영상이 필터링으로 걸러진 후에 필터가 적용됩니다.

Amazon Personalize는 필터링할 때 이벤트 유형별로 사용자당 최대 100개의 항목 상호 작용을 고려합니다. 이는 모든 자동 또는 사용자 지정 필터에 적용됩니다. 이 한도의 증가를 요청하려면 [Service Quotas 콘솔](#)을 사용하면 됩니다. 자세한 내용은 Service Quotas 사용 설명서의 [할당량 증가 요청](#) 단원을 참조하세요.

[항목을 추천할 때 이 사용 사례에서는 항목 `real-time-personalization` 및 탐색을 사용합니다.](#) 또한 [자동 업데이트](#)를 사용하여 새 항목을 추천 대상으로 고려합니다.

- 레시피 ARN: `arn:aws:personalize:::recipe/aws-vod-top-picks`
- GetRecommendations 요구 사항:

`userId`: 필수

`itemId`: 사용되지 않음

- 학습 시 사용되는 데이터 세트:
 - 상호작용(필수)

- 항목(선택 사항)
- 사용자(선택 사항)
- 필요한 이벤트 수: 이벤트 최소 1000개 이상.
- 추천 이벤트 유형: Click 및 Watch 이벤트.
- 탐색 구성 파라미터: 추천자를 만들 때 다음을 사용하여 탐색을 구성할 수 있습니다.
 - 관련성이 낮은 항목을 탐색하는 데 집중(탐색 가중치) — 탐색 범위를 구성합니다. 0과 1 사이에서 소수 값을 지정합니다. 기본값은 0.3입니다. 값이 1에 가까울수록 더 많이 탐색합니다. 더 많이 탐색할수록 항목 상호 작용 데이터가 적거나 이전 행동을 기반으로 한 관련성이 낮은 항목이 더 많이 추천에 포함됩니다. 0이면 탐색이 수행되지 않으며 추천이 현재 데이터(관련성)를 기반으로 합니다.
 - 탐색 항목 기간 제한 – 항목 상호 작용 데이터 세트의 모든 항목에서 최근 상호 작용이 발생한 이후 최대 항목 사용 기간(일)을 지정합니다. 이렇게 하면 항목 기간에 따른 항목 탐색 범위가 정의됩니다. Amazon Personalize는 생성 타임스탬프 또는 생성 타임스탬프 데이터가 누락된 경우 항목 상호 작용 데이터를 기반으로 항목 기간을 결정합니다. Amazon Personalize가 항목 기간을 결정하는 방법에 대한 자세한 내용은 [생성 타임스탬프 데이터](#) 단원을 참조하세요.

Amazon Personalize가 탐색 중에 고려하는 항목을 늘리려면 더 큰 값을 입력합니다. 최소값은 1일이고 기본값은 30일입니다. 추천에 지정한 아이템 보관 기간 한도보다 오래된 항목이 포함될 수 있습니다. 이는 이러한 항목이 사용자와 관련이 있고 탐색에서 해당 항목을 식별할 수 없었기 때문입니다.

ECOMMERCE 사용 사례

다음에 이어지는 단원에는 각 ECOMMERCE 사용 사례에 대한 요구 사항 및 Amazon 리소스 이름 (ARN)이 나열되어 있습니다. 모든 사용 사례의 경우 상호작용 데이터는 다음을 가져야만 합니다.

- 카탈로그의 항목과 상호 작용하는 사용자의 1,000개 이상의 항목 상호 작용 기록. 이러한 상호 작용은 대량 가져오기, 스트리밍된 이벤트 또는 이 둘 다에서 발생할 수 있습니다.
- 각각 2번 이상의 항목 상호 작용을 갖는 25개 이상의 고유 사용자 ID.

품질 추천의 경우, 각 2번 이상의 항목 상호 작용을 갖는 1,000명 이상의 사용자로부터 50,000건 이상의 항목 상호 작용을 갖는 것이 좋습니다.

Note

[CreateRecommender](#) API를 사용하는 경우 레시피 ARN에 대해 여기에 나열되어 있는 ARN을 제공합니다.

주제

- [가장 많이 본 항목](#)
- [베스트 셀러](#)
- [자주 함께 구매한 제품](#)
- [X를 본 고객도 보는 제품](#)
- [추천 제품](#)

가장 많이 본 항목

고객이 항목을 본 횟수를 기반으로 인기 항목을 추천 받을 수 있습니다.

- 레시피 ARN: `arn:aws:personalize:::recipe/aws-ecomm-popular-items-by-views`

- GetRecommendations 요구 사항:

`userId`: 필수

`itemId`: 사용되지 않음

`inputList`: 해당 사항 없음

- 교육 시 사용되는 데이터 세트: 항목 상호 작용 데이터 세트 전용(필수)
- 필요한 이벤트 유형: View 이벤트가 최소 1000개 이상.

베스트 셀러

고객이 항목을 구매한 횟수를 기반으로 인기 항목을 추천 받을 수 있습니다.

- 레시피 ARN: `arn:aws:personalize:::recipe/aws-ecomm-popular-items-by-purchases`

- GetRecommendations 요구 사항:

`userId`: 필수

itemId: 사용되지 않음

inputList: 해당 사항 없음

- 교육 시 사용되는 데이터 세트: 항목 상호 작용 데이터 세트 전용(필수)
- 필수 이벤트 유형: Purchase 이벤트가 최소 1000개 이상.

자주 함께 구매한 제품

지정한 항목과 함께 고객이 자주 함께 구매하는 항목을 추천 받을 수 있습니다.

- 레시피 ARN: arn:aws:personalize:::recipe/aws-ecomm-frequently-bought-together
- GetRecommendations 요구 사항:

userId: 필터링 기준을 사용하는 경우에만 필수 CurrentUser

itemId: 필수

inputList: 해당 사항 없음

- 교육 시 사용되는 데이터 세트: 항목 상호 작용 데이터 세트 전용(필수)
- 필요한 이벤트 유형: Purchase 이벤트가 최소 1000개 이상.

X를 본 고객도 보는 제품

지정한 항목을 기반으로 고객이 조회한 항목에 대한 추천도 받을 수 있습니다. 이 사용 사례에서 Amazon Personalize는 지정한 사용자 ID 및 Purchase 이벤트를 기반으로 사용자가 구매한 항목을 자동으로 필터링합니다. 자체 필터를 적용하는 경우 사용자가 이미 구매한 항목을 필터로 걸러낸 후에 자체 필터가 적용됩니다.

Amazon Personalize는 필터링할 때 이벤트 유형별로 사용자당 최대 100개의 항목 상호 작용을 고려합니다. 이는 모든 자동 또는 사용자 지정 필터에 적용됩니다. 이 한도의 증가를 요청하려면 [Service Quotas 콘솔](#)을 사용하면 됩니다. 자세한 내용은 Service Quotas 사용 설명서의 [할당량 증가 요청](#) 단원을 참조하세요.

- 레시피 ARN: arn:aws:personalize:::recipe/aws-ecomm-customers-who-viewed-x-also-viewed
- GetRecommendations 요구 사항:

userId: 필수

itemId: 필수

inputList: 해당 사항 없음

- 교육 시 사용되는 데이터 세트: 항목 상호 작용 데이터 세트 전용(필수)
- 필요한 이벤트 유형: View 이벤트가 최소 1000개 이상.
- 추천 이벤트 유형: Purchase 이벤트.

추천 제품

지정한 사용자를 기반으로 항목에 대한 개인 맞춤형 추천을 받을 수 있습니다. 이 사용 사례에서 Amazon Personalize는 지정한 사용자 ID 및 Purchase 이벤트를 기반으로 사용자가 구매한 항목을 자동으로 필터로 걸러 냅니다. 자체 필터를 적용하는 경우 사용자가 이미 구매한 항목을 필터로 걸러낸 후에 자체 필터가 적용됩니다.

Amazon Personalize는 필터링할 때 이벤트 유형별로 사용자당 최대 100개의 항목 상호 작용을 고려합니다. 이는 모든 자동 또는 사용자 지정 필터에 적용됩니다. 이 한도의 증가를 요청하려면 [Service Quotas 콘솔](#)을 사용하면 됩니다. 자세한 내용은 Service Quotas 사용 설명서의 [할당량 증가 요청](#) 단원을 참조하세요.

항목을 추천할 때 이 사용 사례에서는 항목을 [real-time-personalization](#) 사용하고 [탐색합니다](#). 또한 [자동 업데이트](#)를 사용하여 새 항목을 추천 대상으로 고려합니다.

- 레시피 ARN: arn:aws:personalize:::recipe/aws-ecomm-recommended-for-you
- GetRecommendations 요구 사항:

userId: 필수

itemId: 사용되지 않음

inputList: 해당 사항 없음

- 학습 시 사용되는 데이터 세트:
 - 상호작용(필수)
 - 항목(선택 사항)
 - 사용자(선택 사항)
- 필요한 이벤트 수: 이벤트 최소 1000개 이상.

- 추천 이벤트 유형: View 및 Purchase 이벤트.
- 탐색 구성 파라미터: 추천자를 만들 때 다음을 사용하여 탐색을 구성할 수 있습니다.
 - 관련성이 낮은 항목을 탐색하는 데 집중(탐색 가중치) — 탐색 범위를 구성합니다. 0과 1 사이에서 소수 값을 지정합니다. 기본값은 0.3입니다. 값이 1에 가까울수록 더 많이 탐색합니다. 더 많이 탐색할수록 항목 상호 작용 데이터가 적거나 이전 행동을 기반으로 한 관련성이 낮은 항목이 더 많이 추천에 포함됩니다. 0이면 탐색이 수행되지 않으며 추천이 현재 데이터(관련성)를 기반으로 합니다.
 - 탐색 항목 기간 제한 – 항목 상호 작용 데이터 세트의 모든 항목에서 최근 상호 작용이 발생한 이후 최대 항목 사용 기간(일)을 지정합니다. 이렇게 하면 항목 기간에 따른 항목 탐색 범위가 정의됩니다. Amazon Personalize는 생성 타임스탬프 또는 생성 타임스탬프 데이터가 누락된 경우 항목 상호 작용 데이터를 기반으로 항목 기간을 결정합니다. Amazon Personalize가 항목 기간을 결정하는 방법에 대한 자세한 내용은 [생성 타임스탬프 데이터](#) 단원을 참조하세요.

Amazon Personalize가 탐색 중에 고려하는 항목을 늘리려면 더 큰 값을 입력합니다. 최소값은 1 일이고 기본값은 30일입니다. 추천에 지정한 아이템 보관 기간 한도보다 오래된 항목이 포함될 수 있습니다. 이는 이러한 항목이 사용자와 관련이 있고 탐색에서 해당 항목을 식별할 수 없었기 때문입니다.

레시피 선택

사용자 지정 솔루션을 생성할 때 레시피를 지정하고 교육 파라미터를 구성합니다. 레시피는 특정 사용 사례에 맞추어 준비된 Personalize 알고리즘입니다. Personalize는 일반적인 사용 사례를 기반으로 교육 모델에 대한 사전 정의된 레시피를 제공합니다. 솔루션용 솔루션 버전을 생성하면 Personalize는 레시피 및 학습 구성을 기반으로 솔루션 버전을 지원하는 모델을 교육합니다.

Personalize 레시피는 교육 중에 다음을 사용합니다.

- 미리 정의된 데이터 속성
- 미리 정의된 기능 변환
- 미리 정의된 알고리즘
- 알고리즘에 대한 초기 파라미터 설정

모델을 최적화하기 위해 솔루션 생성 시 이러한 파라미터 중 상당수를 재정의할 수 있습니다. 자세한 내용은 [하이퍼파라미터 및 HPO](#) 단원을 참조하세요.

주제

- [사용 사례별 Personalize 레시피 유형](#)
- [Personalize 레시피](#)
- [사용 가능한 Personalize 레시피 보기](#)
- [USER_PERSONALIZATION](#)
- [POPULAR_ITEMS](#)
- [PERSONALIZED_RANKING](#)
- [RELATED_ITEMS](#)
- [PERSONALIZED_ACTIONS](#)
- [USER_SEGMENTATION](#)

사용 사례별 Personalize 레시피 유형

레시피를 선택하려면 먼저 다음에서 사용 사례를 선택하고 해당하는 레시피 유형을 기록합니다.

- 사용자에 대한 권장 항목(USER_PERSONALIZATION 레시피)

사용자에게 개인화된 추천을 제공하려면 USER_PERSONALIZATION 레시피로 모델을 교육시키세요. 개인화된 추천은 참여도와 전환율을 높이는 데 도움이 됩니다.

- 사용자에 대한 랭킹 항목(PERSONALIZED_RANKING 레시피)

사용자에 맞게 선별된 목록 또는 검색 결과의 순서를 개인화하려면, PERSONALIZED_RANKING 레시피로 모델을 교육시키세요. PERSONALIZED_RANKING 레시피는 특정 사용자의 예상 관심 수준을 기반으로 다양한 입력 항목의 순위를 다시 매겨 개인화된 목록을 생성합니다. 개인화된 목록은 고객 경험을 개선하고 고객 충성도 및 참여도를 높입니다.

- 추세 상품 또는 인기 상품 권장(POPULAR_ITEMS 레시피)

추세 상품 또는 인기 상품을 추천하려면 POPULAR_ITEMS 레시피를 사용합니다. 고객이 다른 사용자가 상호작용을 하는 것을 높이 평가하는 경우 POPULAR_ITEMS를 사용할 수 있습니다. 일반적인 용도로는 바이럴 소셜 미디어 콘텐츠 추천, 속보 기사 또는 최근 스포츠 동영상 추천 등이 있습니다.

- 비슷한 아이템 추천(RELATED_ITEMS 레시피)

자주 함께 구매하는 아이템이나 다른 사용자도 본 영화 등 비슷한 아이템을 추천하려면 RELATED_ITEMS 레시피를 사용해야 합니다. 비슷한 아이템을 추천하면 고객이 아이템을 찾는 데 도움을 줄 수 있고 사용자 전환율을 높일 수 있습니다.

- 차선책 작업 추천(PERSONALIZED_ACTIONS 레시피)

로열티 프로그램에 가입하거나 신용카드를 신청하는 등 사용자에게 차선택 작업을 실시간으로 추천하려면 PERSONALIZED_ACTIONS 레시피를 사용해야 합니다. 차선택 작업을 추천하면 고객 충성도를 높이고, 더 많은 수익을 창출하며, 사용자 경험을 개선할 수 있습니다.

- 사용자 세그먼트 확보(사용자_세분화 레시피)

특정 속성을 가진 항목과 상호작용할 가능성이 가장 높은 사용자와 같이 항목 입력 데이터를 기반으로 사용자 세그먼트를 가져오려면 사용자_세분화 레시피를 사용해야 합니다. 사용자 세그먼트를 확보하면 사용자가 조치를 취할 가능성에 따라 다양한 사용자 세그먼트에 다양한 항목을 홍보하는 고급 마케팅 캠페인을 생성할 수 있습니다.

Personalize 레시피

Personalize는 다음과 같은 유형의 레시피를 제공합니다. 작동 차이 외에도, 다음 표에서처럼 각 유형마다 권장 요구 사항이 서로 다릅니다.

레시피 유형	레시피	API	API 요구 사항
USER_PERSONALIZATION	사용자 개인화 - v2	GetRecommendations	userId: 필수
	사용자-개인 맞춤		itemId: 사용되지 않음
	HRNN 레시피(레거시)		inputList : 해당 사항 없음
	HRNN-메타데이터 레시피(레거시)		
	HRNN-Coldstart 레시피(레거시)		
POPULAR_ITEMS	지금-유행	GetRecommendations	userId: 필요한 필터를 적용하는 경우에만 필수
	인기도-집계		itemId: 사용되지 않음

레시피 유형	레시피	API	API 요구 사항
			inputList : 해당 사항 없음
PERSONALIZED_RANKING	맞춤형 랭킹-v2 개인 맞춤형-순위	GetPersonalizedRanking	userId: 필수 itemId: 해당 사항 없음 inputList : itemId 목록
RELATED_ITEMS	유사-항목 SIMS	GetRecommendations	userId: 필요한 필터를 적용하는 경우에만 필수 itemId: 필수 inputList : 해당 사항 없음
PERSONALIZED_ACTIONS	차선택-작업	GetActionRecommendations	userId: 필수 actionId: 사용되지 않음 itemId: 사용되지 않음 inputList : 해당 사항 없음

레시피 유형	레시피	API	API 요구 사항
사용자_세분화	항목-친화도 항목-속성-친화도	CreateBatchSegmentJob	배치 워크플로우 요구 사항은 배치 세그먼트 작업 생성 단원을 참조 하세요.

사용 가능한 Personalize 레시피 보기

사용 가능한 레시피 목록을 보려면

- Personalize 콘솔에서 데이터세트 그룹을 선택합니다. 탐색 창에서 솔루션 및 레시피를 선택하고, 레시피 탭을 선택합니다.
- AWS SDK for Python (Boto3)를 사용하여 [ListRecipes](#) API를 호출합니다.
- AWS CLI를 사용하여 다음 명령을 사용합니다.

```
aws personalize list-recipes
```

Python(Boto3)에 대해 SDK를 사용하는 레시피에 대한 정보를 가져오려면 [DescribeRecipe](#) API를 호출합니다. 를 사용하여 레시피에 대한 정보를 가져오려면 다음 명령을 사용합니다. AWS CLI

```
aws personalize describe-recipe --recipe-arn recipe_arn
```

USER_PERSONALIZATION

USER_PERSONALIZATION 레시피는 상호작용, 항목 및 사용자 데이터세트를 기반으로 사용자가 상호작용할 항목을 예측합니다. 각 사용자에게 개인화된 추천을 제공하려면 USER_PERSONALIZATION 레시피를 사용하여 모델을 학습시켜야 합니다.

USER_PERSONALIZATION 레시피는 다음과 같습니다.

주제

- [사용자 맞춤화-v2 레시피](#)

- [사용자-개인 맞춤 레시피](#)
- [레거시 사용자 개인 맞춤 레시피](#)

사용자 맞춤화-v2 레시피

사용자 개인화-v2 (aws-user-personalization-v2) 레시피에서는 사용자가 선호도에 따라 상호작용할 항목을 권장합니다. 예를 들어 User-Personalization-v2를 사용하여 스트리밍 앱을 위한 개인화된 영화 추천이나 소매용 앱을 위한 개인화된 제품 추천을 생성할 수 있습니다. 다른 사용 사례로는 뉴스 사이트에 대한 실시간 추천을 생성하거나 개인화된 마케팅 캠페인을 위한 일괄 추천을 생성하는 것이 있습니다.

User-Personalization-v2는 항목 상호 작용 및 항목 데이터셋에서 최대 5백만 개의 항목을 학습할 수 있습니다. 또한 보다 짧은 지연 시간으로 관련성이 더 높은 추천을 생성합니다. [사용자-개인 맞춤](#)

User-Personalization-v2는 데이터를 기반으로 사용자에게 가장 관련성이 높은 항목을 추천하므로 상호 작용 데이터가 있는 기존 항목을 더 자주 추천합니다. 추천에 새 항목이 포함되도록 하려면 생성 타임스탬프를 기반으로 일부 항목을 포함하는 프로모션을 사용할 수 있습니다. 프로모션에 대한 자세한 내용은 [참조하십시오 추천의 항목 홍보](#).

이 레시피는 트랜스포머 기반 아키텍처를 사용하여 컨텍스트를 학습하고 데이터의 관계 및 패턴을 추적하는 모델을 학습시킵니다. 트랜스포머는 입력 시퀀스를 출력 시퀀스로 변환하거나 변경하는 일종의 신경망 아키텍처입니다. Amazon Personalize의 경우 입력 시퀀스는 데이터에 있는 사용자의 품목 상호 작용 기록입니다. 출력 시퀀스는 개인화된 권장 사항입니다. 트랜스포머에 대한 자세한 내용은 [인공 지능의 트랜스포머란 무엇입니까?](#) 를 참조하십시오. AWS 클라우드 컴퓨팅 개념 허브에서

사용자 맞춤화-v2는 다른 레시피와 다른 가격 책정 모델을 사용합니다. 요금에 대한 자세한 내용은 [Amazon Personalize](#) 요금을 참조하십시오.

주제

- [레시피 기능](#)
- [필수 및 선택형 데이터 세트](#)
- [속성 및 하이퍼파라미터](#)

레시피 기능

사용자 개인화-v2는 아이템 추천을 생성할 때 다음과 같은 Amazon Personalize 레시피 기능을 사용합니다.

- 실시간 개인화 — 실시간 개인화를 통해 Amazon Personalize는 사용자의 변화하는 관심에 따라 항목 추천을 업데이트하고 조정합니다. 자세한 정보는 [실시간 개인 맞춤](#)을 참조하세요.
- 탐색 — 탐색의 경우 권장 사항에는 상호 작용 데이터가 적거나 사용자와의 관련성이 낮은 항목이 포함됩니다. 사용자 개인화-v2를 사용하면 Amazon Personalize가 탐색 구성을 대신 처리합니다. 권장 사항에 새 항목이 포함되도록 하려면 프로모션을 사용하여 생성 타임스탬프를 기반으로 새 항목을 포함시킬 수 있습니다. 프로모션에 대한 자세한 내용은 [추천의 항목 홍보](#)를 참조하세요.
- 자동 업데이트 — Amazon Personalize는 자동 업데이트를 통해 2시간마다 최신 모델 (솔루션 버전)을 자동으로 업데이트하여 새 항목을 추천하도록 고려합니다. 자세한 정보는 [자동 업데이트](#)를 참조하세요.
- 권장 사항이 포함된 메타데이터 — User-Personalization-v2 레시피를 사용하면 메타데이터 열이 하나 이상인 항목 데이터 세트가 있는 경우 캠페인에 추천 결과와 함께 항목 메타데이터를 포함할 수 있는 옵션이 자동으로 제공됩니다. 캠페인의 메타데이터를 수동으로 활성화할 필요는 없습니다. 메타데이터를 통해 사용자 인터페이스에서 추천을 강화할 수 있습니다(예: 캐러셀에 영화 장르 추가). 자세한 정보는 [권장 사항의 항목 메타데이터](#)를 참조하세요.

필수 및 선택형 데이터 세트

사용자 맞춤 설정-v2를 사용하려면 최소 1,000개의 항목 상호 작용을 [항목 상호 작용 데이터 세트](#) 생성하고 가져와야 합니다. Amazon Personalize는 주로 아이템 상호 작용 데이터를 기반으로 권장 사항을 생성합니다. User-Personalization-v2는 항목 상호 작용 및 항목 데이터 세트에서 최대 5백만 개의 항목을 학습할 수 있습니다.

사용자 개인화-v2를 사용하면 Amazon Personalize는 다음을 포함하는 아이템 상호 작용 데이터를 사용할 수 있습니다.

- 이벤트 유형 및 이벤트 값 데이터 — Amazon Personalize는 클릭 또는 시청 이벤트 유형과 같은 이벤트 유형 데이터를 사용하여 행동 패턴을 통해 사용자 의도와 관심을 식별합니다. 또한 훈련 전에 이벤트 유형 및 이벤트 값 데이터를 사용하여 레코드를 필터링할 수 있습니다. 자세한 정보는 [이벤트 유형 및 이벤트 값 데이터](#)를 참조하세요.

Note

User-Personalization-v2를 사용하면 이벤트 유형이나 값을 기준으로 필터링하기 전에 상호 작용 데이터를 기반으로 교육 비용이 책정됩니다. 요금에 대한 자세한 내용은 [Amazon Personalize](#) 요금을 참조하십시오.

- 상황별 메타데이터 — 상황별 메타데이터는 이벤트 발생 시 사용자 환경에서 수집하는 상호 작용 데이터입니다 (예: 위치 또는 디바이스 유형). 자세한 정보는 [컨텍스트 메타데이터](#)를 참조하세요.

다음 데이터세트는 선택사항이며 권장사항을 개선할 수 있습니다.

- 사용자 데이터세트 — Amazon Personalize는 사용자 데이터세트의 데이터를 사용하여 사용자와 관심사를 더 잘 이해할 수 있습니다. 또한 사용자 데이터 세트의 데이터를 사용하여 추천을 필터링할 수 있습니다. 가져올 수 있는 사용자 데이터에 대한 자세한 내용은 [사용자 데이터 세트](#) 섹션을 참조하세요.
- 아이템 데이터세트 — Amazon Personalize는 아이템 데이터세트의 데이터를 사용하여 아이템 동작의 연관성과 패턴을 식별할 수 있습니다. 이를 통해 Amazon Personalize는 사용자와 관심사를 이해할 수 있습니다. 또한 아이템 데이터세트의 데이터를 사용하여 권장 사항을 필터링할 수 있습니다. 가져올 수 있는 항목 데이터에 대한 자세한 내용은 [항목 데이터 세트](#) 섹션을 참조하세요.

속성 및 하이퍼파라미터

사용자 맞춤설정-v2 레시피에는 다음과 같은 속성이 있습니다.

- 명칭 - aws-user-personalization-v2
- 레시피 리소스 이름(ARN) - arn:aws:personalize:::recipe/aws-user-personalization-v2
- 알고리즘 ARN - arn:aws:personalize:::algorithm/aws-user-personalization-v2

자세한 정보는 [레시피 선택](#)을 참조하세요.

다음 표에서는 사용자 개인화-v2 레시피의 하이퍼파라미터에 대해 설명합니다. 하이퍼파라미터는 모델 성능을 향상시키기 위해 조정할 수 있는 알고리즘 파라미터입니다. 알고리즘 하이퍼파라미터는 모델이 성능을 내는 방식을 제어합니다. 하이퍼파라미터에 대한 최적 값을 선택하는 과정을 하이퍼파라미터 최적화(HPO)라고 합니다. 사용자 개인화-v2를 사용할 경우 자동 교육을 켜면 Amazon Personalize는 90일마다 HPO를 자동으로 수행합니다. 자동 교육이 없으면 HPO가 발생하지 않습니다.

표에는 각 하이퍼파라미터에 대한 다음 정보가 표시됩니다.

- 범위: [하한, 상한]
- 값 형식: 정수, 연속(부동 소수점), 범주(부울, 목록, 문자열)

명칭	설명
알고리즘 하이퍼파라미터	
apply_recency_bias	<p>모델이 항목 상호 작용 데이터셋의 최신 항목 상호 작용 데이터에 더 많은 가중치를 부여해야 하는지 여부를 결정합니다. 가장 최근의 상호작용 데이터에는 상호작용 이벤트의 기본 패턴의 급격한 변화가 포함될 수 있습니다.</p> <p>최신 이벤트에 더 많은 가중치를 부여하는 모델을 학습시키려면 <code>apply_recency_bias</code> 를 <code>true</code>로 설정합니다. 지난 모든 상호작용에 동일한 가중치를 부여하는 모델을 학습시키려면 <code>apply_recency_bias</code> 를 <code>false</code>로 설정합니다.</p> <p>기본 값: <code>true</code></p> <p>범위: <code>true</code>또는 <code>false</code></p> <p>값 형식: 부울</p> <p>HPO 조정 가능: 아니요</p>

사용자-개인 맞춤 레시피

Important

[사용자 개인화-v2](#) 레시피를 사용하는 것이 좋습니다. 더 빠른 학습으로 최대 5백만 개의 항목을 고려하고 지연 시간을 줄이면서 더 관련성 높은 추천을 생성할 수 있습니다.

User-Personalization (`aws-user-personalization`) 레시피는 모든 개인화된 추천 시나리오에 최적화되어 있습니다. 사용자가 상호작용할 가능성이 가장 높은 항목을 예측합니다. 사용자 맞춤 설정을 사용하여 스트리밍 앱을 위한 개인화된 영화 추천이나 소매 앱을 위한 개인화된 제품 추천을 생성할 수 있습니다.

Amazon Personalize는 사용자 개인화를 통해 주로 항목 상호 작용 데이터 세트의 사용자 항목 상호 작용 데이터를 기반으로 권장 사항을 생성합니다. 또한 아이템 및 사용자 데이터 세트의 모든 아이템 및 사용자 메타데이터를 사용할 수 있습니다. 사용하는 데이터에 대한 자세한 내용은 [필수 및 선택형 데이터 세트](#).

주제

- [레시피 기능](#)
- [필수 및 선택형 데이터 세트](#)
- [속성 및 하이퍼파라미터](#)
- [사용자-개인 맞춤 레시피를 사용한 학습\(콘솔\)](#)
- [사용자-개인 맞춤 레시피를 사용한 학습\(Python SDK\)](#)
- [추천 받기 및 노출 기록\(Python용 SDK\(Boto3\)\)](#)
- [샘플 Jupyter Notebook](#)

레시피 기능

사용자 맞춤화는 아이템 추천을 생성할 때 다음과 같은 Amazon Personalize 레시피 기능을 사용합니다.

- **실시간 개인화** — 실시간 개인화를 통해 Amazon Personalize는 사용자의 변화하는 관심에 따라 항목 추천을 업데이트하고 조정합니다. 자세한 정보는 [실시간 개인 맞춤](#)을 참조하세요.
- **탐색** — 탐색의 경우 권장 사항에는 새 항목이나 상호 작용 데이터가 적은 항목이 포함됩니다. 이렇게 하면 카탈로그가 빠르게 변경되거나 뉴스 기사나 홍보와 같은 새 항목을 새로 만들었을 때 사용자에게 더 관련성이 높은 항목을 발견하고 참여도를 높일 수 있습니다. 탐색에 대한 자세한 내용은 [탐색단원](#)을 참조하세요.
- **자동 업데이트** — Amazon Personalize는 자동 업데이트를 통해 2시간마다 최신 모델 (솔루션 버전)을 자동으로 업데이트하여 새 항목을 추천하도록 고려합니다. 자세한 정보는 [자동 업데이트](#)을 참조하세요.

필수 및 선택형 데이터 세트

사용자 개인화를 사용하려면 최소 1,000개의 항목 상호 작용을 [항목 상호 작용 데이터 세트](#) 생성하고 가져와야 합니다. Amazon Personalize는 주로 아이템 상호 작용 데이터를 기반으로 권장 사항을 생성합니다.

Amazon Personalize는 사용자 개인화를 통해 다음을 포함하는 아이템 상호 작용 데이터를 사용할 수 있습니다.

- 이벤트 유형 및 이벤트 값 데이터 — Amazon Personalize는 클릭 또는 시청 이벤트 유형과 같은 이벤트 유형 데이터를 사용하여 행동 패턴을 통해 사용자 의도와 관심을 식별합니다. 또한 훈련 전에 이벤트 유형 및 이벤트 값 데이터를 사용하여 레코드를 필터링할 수 있습니다. 자세한 정보는 [이벤트 유형 및 이벤트 값 데이터](#)를 참조하세요.
- 상황별 메타데이터 - 상황별 메타데이터는 이벤트 발생 시 사용자 환경에서 수집하는 상호 작용 데이터입니다 (예: 위치 또는 장치 유형). 자세한 정보는 [컨텍스트 메타데이터](#)를 참조하세요.
- 노출 데이터 — 노출 횟수는 사용자가 특정 항목과 상호 작용 (클릭, 시청, 구매 등) 했을 때 볼 수 있었던 항목의 목록입니다. 자세한 정보는 [노출 데이터](#)를 참조하세요.

다음 데이터세트는 선택 사항이며 권장사항을 개선할 수 있습니다.

- 사용자 데이터세트 — Amazon Personalize는 사용자 데이터세트의 데이터를 사용하여 사용자와 관심사를 더 잘 이해할 수 있습니다. 또한 사용자 데이터 세트의 데이터를 사용하여 추천을 필터링할 수 있습니다. 가져올 수 있는 사용자 데이터에 대한 자세한 내용은 [사용자 데이터 세트](#) 섹션을 참조하세요.
- 아이템 데이터세트 — Amazon Personalize는 아이템 데이터세트의 데이터를 사용하여 아이템 동작의 연관성과 패턴을 식별할 수 있습니다. 이를 통해 Amazon Personalize는 사용자와 관심사를 이해할 수 있습니다. 또한 아이템 데이터세트의 데이터를 사용하여 권장 사항을 필터링할 수 있습니다. 가져올 수 있는 항목 데이터에 대한 자세한 내용은 [항목 데이터 세트](#) 섹션을 참조하세요.

속성 및 하이퍼파라미터

사용자-개인 맞춤 레시피의 속성은 다음과 같습니다.

- 명칭 - aws-user-personalization
- 레시피 리소스 이름(ARN) - arn:aws:personalize:::recipe/aws-user-personalization
- 알고리즘 ARN - arn:aws:personalize:::algorithm/aws-user-personalization

자세한 내용은 [레시피 선택](#) 단원을 참조하세요.

다음 표에서는 사용자-개인 맞춤 레시피의 하이퍼파라미터를 설명합니다. 하이퍼파라미터는 모델 성능을 향상시키기 위해 조정할 수 있는 알고리즘 파라미터입니다. 알고리즘 하이퍼파라미터는 모델이

성능을 내는 방식을 제어합니다. 특성화 하이퍼파라미터는 학습에서 사용할 데이터를 필터링하는 방법을 제어합니다. 하이퍼파라미터에 대한 최적 값을 선택하는 과정을 하이퍼파라미터 최적화(HPO)라고 합니다. 자세한 정보는 [하이퍼파라미터 및 HPO](#)을 참조하세요.

표에는 각 하이퍼파라미터에 대한 다음 정보가 표시됩니다.

- 범위: [하한, 상한]
- 값 형식: 정수, 연속(부동 소수점), 범주(부울, 목록, 문자열)
- 조정 가능한 HPO: 파라미터가 HPO에 참여할 수 있습니까?

명칭	설명
알고리즘 하이퍼파라미터	
hidden_dimension	<p>모델에 사용된 숨겨진 변수 개수. 숨겨진 변수는 사용자의 구매 이력 및 항목 통계를 재생성하여 순위 점수를 생성합니다. 항목 상호 작용 데이터 세트에 더 복잡한 패턴이 포함되는 경우 더 많은 숨겨진 차원을 지정합니다. 더 많은 숨겨진 차원을 사용하는 데는 더 큰 데이터세트와 더 많은 처리 시간이 필요합니다. 최상의 값을 결정하려면 HPO를 사용합니다. HPO를 사용하려면 CreateSolution 및 CreateSolutionVersion 작업 호출 시 <code>performHPO</code> 를 <code>true</code>로 설정합니다.</p> <p>기본 값: 149</p> <p>범위: [32, 256]</p> <p>값 형식: 정수</p> <p>HPO 조정 가능: 예</p>
bptt	<p>시간 기법을 통한 역전파를 사용할지 여부를 결정합니다. 시간을 기준으로 한 역전파는 반복 신경망 기반 알고리즘의 가중치를 업데이트하는 기법입니다. 장기 크레딧에 대한 bptt를 사용하여 지연된 보상을 초기 이벤트에 연결합니다. 예를 들어 지연된 보상은 여러 번 클릭 후 이루어진 구매일 수 있습니다. 초기 이벤트는</p>

명칭	설명
	<p>초기 클릭일 수 있습니다. 클릭과 같은 동일한 이벤트 유형 내에서도 장기 효과를 고려하고 전체 보상을 최대화하는 것이 좋습니다. 장기 효과를 고려하려면 더 큰 bptt값을 사용합니다. 더 큰 bptt값을 사용하는 데는 더 큰 데이터세트와 더 많은 처리 시간이 필요합니다.</p> <p>기본 값: 32</p> <p>범위: [2, 32]</p> <p>값 형식: 정수</p> <p>HPO 조정 가능: 예</p>
recency_mask	<p>모델이 항목 상호 작용 데이터 세트의 최신 인기 트렌드를 고려해야 하는지 여부를 결정합니다. 최신 인기 트렌드에는 상호작용 이벤트 기본 패턴의 갑작스러운 변경이 포함될 수 있습니다. 최신 이벤트에 더 많은 가중치를 부여하는 모델을 학습시키려면 recency_mask 를 true로 설정합니다. 지난 모든 상호작용에 동일한 가중치를 부여하는 모델을 학습시키려면 recency_mask 를 false로 설정합니다. 동일한 가중치를 사용하여 좋은 추천을 받으려면 더 큰 학습 데이터세트가 필요할 수 있습니다.</p> <p>기본 값: True</p> <p>범위: True또는 False</p> <p>값 형식: 부울</p> <p>HPO 조정 가능: 예</p>

특성화 하이퍼파라미터

명칭	설명
<p><code>min_user_history_length_percentile</code></p>	<p>모델 학습에 포함시킬 사용자 이력 길이의 최소 백분위수. 이력 길이는 사용자에게 대한 데이터의 총 용량입니다. 이력 길이가 짧은 사용자의 백분율을 제외시키려면 <code>min_user_history_length_percentile</code> 을 사용합니다. 이력 길이가 짧은 사용자는 사용자의 개인 요구나 필요 대신에 항목 인기 기반의 패턴을 종종 보입니다. 이 패턴을 제거하면 데이터의 기본 패턴에 더 많이 집중하며 모델을 학습시킬 수 있습니다. 히스토그램 또는 유사한 도구를 사용하여 사용자 이력 길이를 검토한 후 적절한 값을 선택합니다. 다수의 항목을 유지하되 극단적 경우를 제거하는 값을 설정하는 것이 좋습니다.</p> <p>예를 들어 <code>min_user_history_length_percentile to 0.05</code> 및 <code>max_user_history_length_percentile to 0.95</code> 를 설정하면 이력 길이가 하위 또는 상위 5%에 해당하는 사용자를 제외한 모든 사용자가 포함됩니다.</p> <p>기본 값: 0.0</p> <p>범위: [0.0, 1.0]</p> <p>값 형식: 부동 소수점</p> <p>HPO 조정 가능: 아니요</p>

명칭	설명
<p>max_user_history_length_percentile</p>	<p>모델 학습에 포함시킬 사용자 이력 길이의 최대 백분위수. 이력 길이는 사용자에 대한 데이터의 총 용량입니다. 이력 길이가 긴 사용자의 백분율을 제외시키려면 max_user_history_length_percentile 을 사용합니다. 이러한 사용자의 데이터에는 노이즈가 포함되는 경향이 있기 때문입니다. 예를 들어 로봇은 자동화된 상호작용의 긴 목록을 포함할 수 있습니다. 이러한 사용자를 제거하는 경우 학습 시 노이즈가 제한됩니다. 히스토그램 또는 유사한 도구를 사용하여 사용자 이력 길이를 검토한 후 적절한 값을 선택합니다. 다수의 항목을 유지하되 극단적 경우를 제거하는 값을 설정하는 것이 좋습니다.</p> <p>예를 들어 min_user_history_length_percentile to 0.05 및 max_user_history_length_percentile to 0.95 를 설정하면 이력 길이가 하위 또는 상위 5%에 해당하는 사용자를 제외한 모든 사용자가 포함됩니다.</p> <p>기본 값: 0.99</p> <p>범위: [0.0, 1.0]</p> <p>값 형식: 부동 소수점</p> <p>HPO 조정 가능: 아니요</p>

항목 탐색, 캠페인 구성, 하이퍼파라미터

명칭	설명
exploration_weight	<p>항목 상호 작용 데이터나 관련성이 낮은 항목을 추천에 포함시키는 빈도를 결정합니다. 값이 1에 가까울수록 더 많이 탐색합니다. 0이면 탐색이 수행되지 않으며 추천이 현재 데이터(관련성)를 기반으로 합니다. 자세한 내용은 the section called “CampaignConfig” 단원을 참조하세요.</p> <p>기본 값: 0.3</p> <p>범위: [0.0, 1.0]</p> <p>값 형식: 부동 소수점</p> <p>HPO 조정 가능: 아니요</p>

명칭	설명
exploration_item_age_cut_off	<p>항목 상호 작용 데이터 세트의 모든 항목에서 최근 상호 작용이 발생한 이후 최대 항목 사용 기간(일)을 지정합니다. 이렇게 하면 항목 기간에 따른 항목 탐색 범위가 정의됩니다. Amazon Personalize는 생성 타임스탬프 또는 생성 타임스탬프 데이터가 누락된 경우 항목 상호 작용 데이터를 기반으로 항목 사용 기간을 결정합니다. Personalize가 항목의 수명을 결정하는 방법에 대한 자세한 내용은 생성 타임스탬프 데이터 단원을 참조하세요.</p> <p>Personalize가 탐색 중에 고려하는 항목을 늘리려면 더 큰 값을 입력합니다. 최소값은 1일이고, 기본값은 30일입니다. 지정한 항목 보관 기간 한도보다 오래된 항목이 추천에 포함될 수 있습니다. 이는 이러한 항목이 사용자와 관련이 있고 탐색에서 해당 항목을 식별할 수 없었기 때문입니다.</p> <p>기본 값: 30.0</p> <p>범위: 양의 부동 소수점</p> <p>값 형식: 부동 소수점</p> <p>HPO 조정 가능: 아니요</p>

사용자-개인 맞춤 레시피를 사용한 학습(콘솔)

사용자-개인 맞춤 레시피를 사용하여 콘솔에서 추천을 생성하려면 먼저 레시피를 사용하여 새 솔루션 버전을 학습시킵니다. 그런 다음 솔루션 버전을 사용하여 캠페인을 배포하고 캠페인을 사용하여 추천을 받습니다.

사용자-개인 맞춤 레시피를 사용한 새 솔루션 버전 학습(콘솔)

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.

2. 새 스키마를 사용하여 사용자 지정 데이터세트 그룹을 만들고 노출 데이터가 포함된 데이터세트를 업로드합니다. Personalize가 항목의 수명을 더 정확하게 계산하고 콜드 항목을 식별할 수 있도록 항목 데이터세트에 [CREATION_TIMESTAMP](#) 및 [비정형 텍스트 메타데이터](#) 데이터를 선택적으로 포함시킵니다.

데이터 가져오기에 대한 자세한 내용은 [단계 2: 데이터 준비 및 가져오기](#) 단원을 참조하세요.

3. 데이터세트 그룹 페이지에서 노출 데이터가 포함된 데이터세트를 포함하는 새 데이터세트 그룹을 선택합니다.
4. 탐색 창에서 솔루션 및 레시피를 선택하고, 솔루션 생성을 선택합니다.
5. 솔루션 생성 페이지에서 솔루션 이름에 새 솔루션의 이름을 입력합니다.
6. 솔루션 유형에 대해 항목 추천을 선택하여 사용자를 위한 항목 추천을 받습니다.
7. 레시피에서 선택하세요 `aws-user-personalization`. 여러 구성 옵션을 제공하는 솔루션 구성 섹션이 나타납니다.
8. 추가 구성에서 아이템 상호작용 데이터세트에 `EVENT_TYPE`이 있거나 `EVENT_TYPE`과 `EVENT_VALUE` 열이 모두 있는 경우 선택적으로 이벤트 유형 및 이벤트 값 임계값 필드를 사용하여 Amazon Personalize가 모델을 학습할 때 사용하는 아이템 상호 작용 데이터를 선택할 수 있습니다. 자세한 내용은 [교육에 사용할 항목 상호 작용 데이터 선택](#) 단원을 참조하세요.
9. 솔루션에 맞게 하이퍼파라미터를 구성할 수도 있습니다. 사용자-개인 맞춤 레시피 속성 및 하이퍼파라미터의 목록은 [속성 및 하이퍼파라미터](#) 단원을 참조하세요.
10. 솔루션 생성 및 학습을 선택하여 학습을 시작합니다. 대시보드 페이지가 표시됩니다.

솔루션 세부 정보 페이지로 이동하여 솔루션 버전 섹션에서 학습 진행 상황을 추적할 수 있습니다. 상태가 활성인 경우 학습이 완료된 것입니다.

캠페인 생성 및 추천 받기(콘솔)

솔루션 버전 상태가 활성이면 다음과 같이 캠페인을 만들고 추천을 받을 수 있습니다.

1. 솔루션 세부 정보 페이지 또는 캠페인 페이지에서 새 캠페인 생성을 선택합니다.
2. 새 캠페인 생성 페이지에서 캠페인 세부 정보에 대해 다음 정보를 제공합니다.
 - 캠페인 이름: 캠페인 이름을 입력합니다. 여기에 입력하는 텍스트는 캠페인 대시보드 및 세부 정보 페이지에 표시됩니다.
 - 솔루션: 방금 만든 솔루션을 선택합니다.
 - 솔루션 버전 ID: 방금 만든 솔루션 버전의 ID를 선택합니다.

- 초당 최소 프로비저닝된 트랜잭션: Personalize에서 지원하는 초당 최소 프로비저닝된 트랜잭션을 설정합니다. 자세한 정보는 [CreateCampaign](#)작업을 참조하세요.

3. 캠페인 구성에서 다음 정보를 제공합니다.

- 탐색 가중치: 탐색할 양을 구성합니다. 탐색량을 많이 지정할수록 항목 상호 작용 데이터가 적거나 관련성이 낮은 항목이 추천에 포함되는 빈도가 높아집니다. 값이 1에 가까울수록 탐색 횟수가 더 많아집니다. 0이면 탐색이 수행되지 않으며 추천이 현재 데이터(관련성)를 기반으로 합니다.
- 탐색 항목 연령 커트라인: 항목 탐색 범위를 정의하려면 최근 상호작용 이후 최대 항목 수명(일)을 입력합니다. Personalize가 탐색 중에 고려하는 항목 수를 늘리려면 더 큰 값을 입력합니다.

예를 들어, 10을 입력하면 데이터 세트에서 가장 최근 상호 작용이 발생한 이후 10일 동안의 항목 상호 작용 데이터가 있는 항목만 탐색 중에 고려됩니다.

Note

추천에는 이 기간 외의 항목 상호 작용 데이터가 없는 항목이 포함될 수 있습니다. 이러한 항목은 사용자의 관심사와 관련이 있고 이러한 항목을 식별하기 위해 탐색할 필요가 없었기 때문입니다.

4. 캠페인 생성을 선택합니다.
5. 캠페인 세부 정보 페이지에서 캠페인 상태가 활성화일 때 캠페인을 사용하여 추천을 받고 노출 수를 기록할 수 있습니다. 자세한 내용은 “시작하기”의 [5단계: 추천 받기](#) 단원을 참조하세요.

Personalize는 새 데이터가 포함되도록 2시간마다 최신 솔루션 버전을 자동으로 업데이트합니다. 캠페인에서 업데이트된 솔루션 버전을 자동으로 사용합니다. 자세한 내용은 [자동 업데이트](#) 단원을 참조하세요.

캠페인을 수동으로 업데이트하려면 먼저 콘솔이나 `trainingMode`를 `update`으로 설정한 상태에서 [CreateSolutionVersion](#)작업을 사용하여 새 솔루션 버전을 만들고 학습시켜야 합니다. 그런 다음 콘솔의 캠페인 페이지에서 또는 [UpdateCampaign](#)작업을 사용하여 캠페인을 수동으로 업데이트합니다.

Note

Personalize는 2020년 11월 17일 이전에 생성한 솔루션 버전은 자동으로 업데이트하지 않습니다.

사용자-개인 맞춤 레시피를 사용한 학습(Python SDK)

데이터세트 그룹을 만들고 노출 데이터가 포함된 데이터세트를 업로드했다면 사용자-개인 맞춤 레시피로 솔루션을 학습시킬 수 있습니다. Personalize가 항목의 수명을 더 정확하게 계산하고 콜드 항목을 식별할 수 있도록 항목 데이터세트에 [CREATION_TIMESTAMP](#) 및 [비정형 텍스트 메타데이터](#) 데이터를 선택적으로 포함시킵니다. 데이터세트 그룹 생성 및 교육 데이터 업로드에 대한 자세한 내용은 [스키마](#) 단원을 참조하세요.

SDK를 사용하여 사용자-개인화 레시피로 솔루션을 훈련시키려면 AWS

1. `create_solution` 메서드를 사용하여 새 솔루션을 생성합니다.

`solution name`을 솔루션 이름으로 그리고 `dataset group arn`를 데이터세트 그룹의 리소스 이름(ARN)으로 각각 바꿉니다.

```
import boto3

personalize = boto3.client('personalize')

print('Creating solution')
create_solution_response = personalize.create_solution(name = 'solution name',
                                                       recipeArn = 'arn:aws:personalize:::recipe/aws-user-
personalization',
                                                       datasetGroupArn = 'dataset group arn',
                                                       )
solution_arn = create_solution_response['solutionArn']
print('solution_arn: ', solution_arn)
```

`aws-user-personalization` 레시피 속성 및 하이퍼파라미터 목록은 을 참조하십시오. [속성 및 하이퍼파라미터](#)

2. 업데이트된 학습 데이터로 새 솔루션 버전을 만들고 다음 코드 스니펫을 사용하여 `trainingMode`를 FULL으로 설정합니다. `solution arn`을 솔루션의 ARN으로 바꿉니다.

```
import boto3

personalize = boto3.client('personalize')

create_solution_version_response = personalize.create_solution_version(solutionArn
                               = 'solution arn',
                               trainingMode='FULL')
```

```
new_solution_version_arn = create_solution_version_response['solutionVersionArn']
print('solution_version_arn:', new_solution_version_arn)
```

3. Personalize가 솔루션 버전 생성을 완료하면 다음 파라미터를 사용하여 캠페인을 생성합니다.
- 새로운 campaign name과 2단계에서 생성한 solution version arn를 제공합니다.
 - explorationWeight 항목 탐색 구성 하이퍼파라미터를 수정하여 탐색할 양을 구성합니다. 항목 상호 작용 데이터나 관련성이 적은 항목은 값이 1.0에 가까울수록 더 자주 추천됩니다. 기본 값은 0.3입니다.
 - explorationItemAgeCutOff 항목 탐색 구성 하이퍼파라미터 파라미터를 수정하여 항목을 탐색해야 하는 최대 기간(일)을 최신 상호작용을 기준으로 지정합니다. 값이 클수록 탐색 중에 고려되는 항목이 많아집니다.

다음 Python 스니펫을 사용하여 탐색 커트라인을 30일로 제한한 탐색에 중점을 둔 새 캠페인을 만듭니다. 일반적으로 캠페인을 만드는 데 몇 분이면 되지만 한 시간 이상 걸릴 수 있습니다.

```
import boto3

personalize = boto3.client('personalize')

create_campaign_response = personalize.create_campaign(
    name = 'campaign name',
    solutionVersionArn = 'solution version arn',
    minProvisionedTPS = 1,
    campaignConfig = {"itemExplorationConfig": {"explorationWeight": "0.3",
"explorationItemAgeCutOff": "30"}}
)

campaign_arn = create_campaign_response['campaignArn']
print('campaign_arn:', campaign_arn)
```

사용자-개인 맞춤을 통해 Personalize는 새 데이터를 포함하도록 2시간마다 솔루션 버전을 자동으로 업데이트합니다. 캠페인에서 업데이트된 솔루션 버전을 자동으로 사용합니다. 자세한 내용은 [자동 업데이트](#) 단원을 참조하세요.

캠페인을 수동으로 업데이트하려면 먼저 콘솔이나 trainingMode를 update으로 설정한 상태에서 [CreateSolutionVersion](#) 작업을 사용하여 새 솔루션 버전을 만들고 학습시켜야 합니다. 그런 다음 콘솔의 캠페인 페이지에서 또는 [UpdateCampaign](#) 작업을 사용하여 캠페인을 수동으로 업데이트합니다.

Note

Personalize는 2020년 11월 17일 이전에 생성한 솔루션 버전은 자동으로 업데이트하지 않습니다.

추천 받기 및 노출 기록(Python용 SDK(Boto3))

캠페인이 생성되면 이를 사용하여 사용자에게 대한 추천을 받고 노출을 기록할 수 있습니다. AWS SDK를 사용하여 배치 권장 사항을 얻는 방법에 대한 자세한 내용은 [일괄 추천 작업 \(AWS SDK\) 생성](#)

추천을 받고 노출을 기록하려면

1. `get_recommendations` 메서드를 호출합니다. `campaign arn`를 새 캠페인의 ARN으로 그리고 `user id`를 사용자의 UserID로 각각 바꿉니다.

```
import boto3

rec_response = personalize_runtime.get_recommendations(campaignArn = 'campaign arn',
                                                       userId = 'user id')
print(rec_response['recommendationId'])
```

2. PutEvents 요청 전송을 위한 새 이벤트 트래커를 만드세요. `event tracker name`을 이벤트 트래커의 이름으로 그리고 `dataset group arn`를 데이터세트 그룹의 ARN으로 바꿉니다.

```
import boto3

personalize = boto3.client('personalize')

event_tracker_response = personalize.create_event_tracker(
    name = 'event tracker name',
    datasetGroupArn = 'dataset group arn'
)
event_tracker_arn = event_tracker_response['eventTrackerArn']
event_tracking_id = event_tracker_response['trackingId']
print('eventTrackerArn:{},\n eventTrackingId:{}'.format(event_tracker_arn,
                                                         event_tracking_id))
```

3. 1단계의 recommendationId과 2단계의 event tracking id를 사용하여 새 PutEvents요청을 생성합니다. 이 요청은 사용자 세션의 새 노출 데이터를 기록합니다. user id를 사용자의 ID로 변경합니다.

```
import boto3

personalize_events.put_events(
    trackingId = 'event tracking id',
    userId= 'user id',
    sessionId = '1',
    eventList = [{
        'sentAt': datetime.now().timestamp(),
        'eventType' : 'click',
        'itemId' : rec_response['itemList'][0]['itemId'],
        'recommendationId': rec_response['recommendationId'],
        'impression': [item['itemId'] for item in rec_response['itemList']],
    }]
)
```

샘플 Jupyter Notebook

사용자-개인 맞춤 레시피를 사용하는 방법을 보여주는 샘플 Jupyter Notebook은 [탐색을 통한 사용자-개인 맞춤](#) 단원을 참조하세요.

레거시 사용자 개인 맞춤 레시피

Note

레거시 HRNN 레시피는 더 이상 사용할 수 없습니다. 이 설명서는 참조용입니다. 기존 HRNN 레시피보다 aws-user-personalization (사용자-개인화) 레시피를 사용하는 것이 좋습니다. 사용자-개인 맞춤은 HRNN 레시피가 제공하는 기능을 개선하고 통합한 레시피입니다. 자세한 내용은 [사용자-개인 맞춤 레시피](#) 단원을 참조하세요.

다음은 레거시 USER_PERSONALIZATION 레시피입니다.

- [HRNN 레시피\(레거시\)](#)
- [HRNN-Coldstart 레시피\(레거시\)](#)
- [HRNN-메타데이터 레시피\(레거시\)](#)

HRNN 레시피(레거시)

Note

레거시 HRNN 레시피는 더 이상 사용할 수 없습니다. 이 설명서는 참조용입니다. 레거시 HRNN 레시피보다 `aws-user-personalization`(사용자-개인 맞춤) 레시피를 사용하는 것이 좋습니다. 사용자-개인 맞춤은 HRNN 레시피가 제공하는 기능을 개선하고 통합한 레시피입니다. 자세한 내용은 [사용자-개인 맞춤 레시피](#) 내용을 참조하세요.

Amazon Personalize 계층적 반복 신경망(HRNN) 레시피 모델은 사용자 행동의 변화를 모델링하여 세션 중에 추천을 제공합니다. 세션은 예를 들어 요구를 충족하는 특정 항목을 찾는 것을 목적으로 한 지정된 시간 내의 사용자 상호작용의 세트입니다. 사용자의 최근 상호작용에 더 높게 가중치를 부여함으로써 세션 중에 더 관련된 추천을 제공할 수 있습니다.

HRNN은 시간의 흐름에 따라 변경될 수 있는 사용자 의도와 관심사를 수용합니다. 그리고 정렬된 사용자 이력을 가져와 더 나은 추천을 하도록 자동으로 가중치를 부여합니다. HRNN에서는 게이트 메커니즘을 사용하여 할인 가중치를 항목과 타임스탬프의 학습 가능한 함수로 모델링합니다.

Amazon Personalize는 데이터 세트에서 각 사용자의 특성을 추출합니다. 실시간 데이터 통합을 수행한 경우 이러한 특성은 사용자 활동에 따라 실시간으로 업데이트됩니다. 추천을 받으려면 `USER_ID`만을 제공합니다. 또한 `ITEM_ID`를 제공해도 경우 Amazon Personalize가 무시합니다.

HRNN 레시피의 속성은 다음과 같습니다.

- 명칭 - `aws-hrnn`
- 레시피 Amazon 리소스 이름(ARN) - `arn:aws:personalize:::recipe/aws-hrnn`
- 알고리즘 ARN - `arn:aws:personalize:::algorithm/aws-hrnn`
- 특성 변환 ARN - `arn:aws:personalize:::feature-transformation/JSON-percentile-filtering`
- 레시피 유형 - `USER_PERSONALIZATION`

다음 표에서는 HRNN 레시피의 하이퍼파라미터를 설명합니다. 하이퍼파라미터는 모델 성능을 향상시키기 위해 조정할 수 있는 알고리즘 파라미터입니다. 알고리즘 하이퍼파라미터는 모델이 성능을 내는 방식을 제어합니다. 특성화 하이퍼파라미터는 학습에서 사용할 데이터를 필터링하는 방법을 제어합니다. 하이퍼파라미터에 대한 최적 값을 선택하는 과정을 하이퍼파라미터 최적화(HPO)라고 합니다. 자세한 내용은 [하이퍼파라미터 및 HPO](#) 단원을 참조하세요.

표에는 각 하이퍼파라미터에 대한 다음 정보도 표시됩니다.

- 범위: [하한, 상한]
- 값 형식: 정수, 연속(부동 소수점), 범주(부울, 목록, 문자열)
- 조정 가능한 HPO: 파라미터가 HPO에 참여할 수 있습니까?

명칭	Description
알고리즘 하이퍼파라미터	
hidden_dimension	<p>모델에 사용된 숨겨진 변수 개수. 숨겨진 변수는 사용자의 구매 이력 및 항목 통계를 재생성하여 순위 점수를 생성합니다. 항목 상호 작용 데이터 세트에 더 복잡한 패턴이 포함되는 경우 더 많은 숨겨진 차원을 지정합니다. 더 많은 숨겨진 차원을 사용하는 데는 더 큰 데이터 세트와 더 많은 처리 시간이 필요합니다. 최적 값을 결정하려면 HPO를 사용합니다. HPO를 사용하려면 CreateSolution 및 CreateSolutionVersion 작업 호출 시 <code>performHPO</code> 를 <code>true</code>로 설정합니다.</p> <p>기본 값: 43</p> <p>범위: [32, 256]</p> <p>값 형식: 정수</p> <p>HPO 조정 가능: 예</p>
bptt	<p>시간 기법을 통한 역전파를 사용할지 여부를 결정합니다. 시간을 기준으로 한 역전파는 반복 신경망 기반 알고리즘의 가중치를 업데이트하는 기법입니다. 장기 크레딧에 대한 bptt를 사용하여 지연된 보상을 초기 이벤트에 연결합니다. 예를 들어 지연된 보상은 여러 번 클릭 후 이루어진 구매일 수 있습니다. 초기 이벤트는 초기 클릭일 수 있습니다. 클릭과 같은 동일한 이벤트 유형 내에서도 장기 효과를 고려하고 전체 보상을 최대화하는 것이 좋습니다. 장기 효과를 고려하려면 더 큰</p>

명칭	Description
	<p>bptt 값을 사용합니다. 더 큰 bptt 값을 사용하는 데는 더 큰 데이터 세트와 더 많은 처리 시간이 필요합니다.</p> <p>기본 값: 32</p> <p>범위: [2, 32]</p> <p>값 형식: 정수</p> <p>HPO 조정 가능: 예</p>
recency_mask	<p>모델이 항목 상호 작용 데이터 세트의 최신 인기 트렌드를 고려해야 하는지 여부를 결정합니다. 최신 인기 트렌드에는 상호작용 이벤트 기본 패턴의 갑작스러운 변경이 포함될 수 있습니다. 최신 이벤트에 더 많은 가중치를 부여하는 모델을 학습시키려면 recency_mask 를 true로 설정합니다. 지난 모든 상호작용에 동일한 가중치를 부여하는 모델을 학습시키려면 recency_mask 를 false로 설정합니다. 동일한 가중치를 사용하여 좋은 추천을 받으려면 더 큰 학습 데이터 세트가 필요할 수 있습니다.</p> <p>기본 값: True</p> <p>범위: True 또는 False</p> <p>값 형식: 부울</p> <p>HPO 조정 가능: 예</p>

특성화 하이퍼파라미터

명칭	Description
<p><code>min_user_history_length_percentile</code></p>	<p>모델 학습에 포함시킬 사용자 이력 길이의 최소 백분위수. 이력 길이는 사용자에 대한 데이터의 총 용량입니다. 이력 길이가 짧은 사용자의 백분율을 제외시키려면 <code>min_user_history_length_percentile</code> 을 사용합니다. 이력 길이가 짧은 사용자는 사용자의 개인 요구나 필요 대신에 항목 인기 기반의 패턴을 종종 보입니다. 이 패턴을 제거하면 데이터의 기본 패턴에 더 많이 집중하며 모델을 학습시킬 수 있습니다. 히스토그램 또는 유사한 도구를 사용하여 사용자 이력 길이를 검토한 후 적절한 값을 선택합니다. 다수의 사용자를 유지하되 극단적 경우를 제거하는 값을 설정하는 것이 좋습니다.</p> <p>예를 들어 <code>min_user_history_length_percentile to 0.05</code> 및 <code>max_user_history_length_percentile to 0.95</code> 를 설정하면 이력 길이가 하위 또는 상위 5%에 해당하는 사용자를 제외한 모든 사용자가 포함됩니다.</p> <p>기본 값: 0.0</p> <p>범위: [0.0, 1.0]</p> <p>값 형식: 부동 소수점</p> <p>HPO 조정 가능: 아니요</p>

명칭	Description
max_user_history_length_percentile	<p>모델 학습에 포함시킬 사용자 이력 길이의 최대 백분위수. 이력 길이는 사용자에 대한 데이터의 총 용량입니다. 이력 길이가 긴 사용자의 백분율을 제외시키려면 max_user_history_length_percentile 을 사용합니다. 이러한 사용자의 데이터에는 노이즈가 포함되는 경향이 있기 때문입니다. 예를 들어 로봇은 자동화된 상호작용의 긴 목록을 포함할 수 있습니다. 이러한 사용자를 제거하는 경우 학습 시 노이즈가 제한됩니다. 히스토그램 또는 유사한 도구를 사용하여 사용자 이력 길이를 검토한 후 적절한 값을 선택합니다. 다수의 사용자를 유지하되 극단적 경우를 제거하는 값을 설정하는 것이 좋습니다.</p> <p>예를 들어 min__user_history_length_percentile to 0.05 및 max_user_history_length_percentile to 0.95 를 설정하면 이력 길이가 하위 또는 상위 5%에 해당하는 사용자를 제외한 모든 사용자가 포함됩니다.</p> <p>기본 값: 0.99</p> <p>범위: [0.0, 1.0]</p> <p>값 형식: 부동 소수점</p> <p>HPO 조정 가능: 아니요</p>

HRNN-메타데이터 레시피(레거시)

Note

레거시 HRNN 레시피는 더 이상 사용할 수 없습니다. 이 설명서는 참조용입니다. 레거시 HRNN 레시피보다 aws-user-personalization(사용자-개인 맞춤) 레시피를 사용하는 것이 좋습니다. 사용자-개인 맞춤은 HRNN 레시피가 제공하는 기능을 개선하고 통합한 레시피입니다. 자세한 내용은 [사용자-개인 맞춤 레시피](#) 단원을 참조하세요.

HRNN-메타데이터 레시피는 사용자가 상호작용할 항목을 예측합니다. 컨텍스트, 사용자 및 항목 메타데이터(각각 상호작용, 사용자, 항목 데이터 세트)에서 파생된 추가 기능이 있는 [HRNN](#) 레시피와 비슷합니다. HRNN-메타데이터는 고품질 메타데이터를 사용할 수 있을 때 비 메타데이터 모델보다 정확도가 더 높습니다. 이 레시피를 사용하려면 학습 시간이 더 필요할 수도 있습니다.

HRNN-메타데이터 레시피의 속성은 다음과 같습니다.

- 명칭 - aws-hrnn-metadata
- 레시피 Amazon 리소스 이름(ARN) - arn:aws:personalize:::recipe/aws-hrnn-metadata
- 알고리즘 ARN - arn:aws:personalize:::algorithm/aws-hrnn-metadata
- 특성 변환 ARN - arn:aws:personalize:::feature-transformation/featurize_metadata
- 레시피 유형 - USER_PERSONALIZATION

다음 표에서는 HRNN-메타데이터 레시피의 하이퍼파라미터를 설명합니다. 하이퍼파라미터는 모델 성능을 향상시키기 위해 조정할 수 있는 알고리즘 파라미터입니다. 알고리즘 하이퍼파라미터는 모델이 성능을 내는 방식을 제어합니다. 특성화 하이퍼파라미터는 학습에서 사용할 데이터를 필터링하는 방법을 제어합니다. 하이퍼파라미터에 대한 최적 값을 선택하는 과정을 하이퍼파라미터 최적화(HPO)라고 합니다. 자세한 내용은 [하이퍼파라미터 및 HPO](#) 단원을 참조하세요.

표에는 각 하이퍼파라미터에 대한 다음 정보도 표시됩니다.

- 범위: [하한, 상한]
- 값 형식: 정수, 연속(부동 소수점), 범주(부울, 목록, 문자열)
- 조정 가능한 HPO: 파라미터가 하이퍼파라미터 최적화(HPO)에 참여할 수 있습니까?

명칭	Description
알고리즘 하이퍼파라미터	
hidden_dimension	모델에 사용된 숨겨진 변수 개수. 숨겨진 변수는 사용자의 구매 이력 및 항목 통계를 재생성하여 순위 점수를 생성합니다. 항목 상호 작용 데이터 세트에 더 복잡한 패턴이 포함되는 경우 더 많은 숨겨진 차원을 지정합니다. 더 많은 숨겨진 차원을 사용하는 데는 더 큰 데이터 세트와 더 많은 처리 시간이 필요합니다. 최적 값

명칭	Description
	<p>을 결정하려면 HPO를 사용합니다. HPO를 사용하려면 CreateSolution 및 CreateSolutionVersion 작업 호출 시 performHPO 를 true로 설정합니다.</p> <p>기본 값: 43</p> <p>범위: [32, 256]</p> <p>값 형식: 정수</p> <p>HPO 조정 가능: 예</p>
bptt	<p>시간 기법을 통한 역전파를 사용할지 여부를 결정합니다. 시간을 기준으로 한 역전파는 반복 신경망 기반 알고리즘의 가중치를 업데이트하는 기법입니다. 장기 크레딧에 대한 bptt를 사용하여 지연된 보상을 초기 이벤트에 연결합니다. 예를 들어 지연된 보상은 여러 번 클릭 후 이루어진 구매일 수 있습니다. 초기 이벤트는 초기 클릭일 수 있습니다. 클릭과 같은 동일한 이벤트 유형 내에서도 장기 효과를 고려하고 전체 보상을 최대화하는 것이 좋습니다. 장기 효과를 고려하려면 더 큰 bptt 값을 사용합니다. 더 큰 bptt 값을 사용하는 데는 더 큰 데이터 세트와 더 많은 처리 시간이 필요합니다.</p> <p>기본 값: 32</p> <p>범위: [2, 32]</p> <p>값 형식: 정수</p> <p>HPO 조정 가능: 예</p>

명칭	Description
recency_mask	<p>모델이 항목 상호 작용 데이터 세트의 최신 인기 트렌드를 고려해야 하는지 여부를 결정합니다. 최신 인기 트렌드에는 상호작용 이벤트 기본 패턴의 갑작스러운 변경이 포함될 수 있습니다. 최신 이벤트에 더 많은 가중치를 부여하는 모델을 학습시키려면 recency_mask 를 true로 설정합니다. 지난 모든 상호작용에 동일한 가중치를 부여하는 모델을 학습시키려면 recency_mask 를 false로 설정합니다. 동일한 가중치를 사용하여 좋은 추천을 받으려면 더 큰 학습 데이터 세트가 필요할 수 있습니다.</p> <p>기본 값: True</p> <p>범위: True 또는 False</p> <p>값 형식: 부울</p> <p>HPO 조정 가능: 예</p>
특성화 하이퍼파라미터	

명칭	Description
<p><code>min_user_history_length_percentile</code></p>	<p>모델 학습에 포함시킬 사용자 이력 길이의 최소 백분위수. 이력 길이는 사용자에 대한 데이터의 총 용량입니다. 이력 길이가 짧은 사용자의 백분율을 제외시키려면 <code>min_user_history_length_percentile</code> 을 사용합니다. 이력 길이가 짧은 사용자는 사용자의 개인 요구나 필요 대신에 항목 인기 기반의 패턴을 종종 보입니다. 이 패턴을 제거하면 데이터의 기본 패턴에 더 많이 집중하며 모델을 학습시킬 수 있습니다. 히스토그램 또는 유사한 도구를 사용하여 사용자 이력 길이를 검토한 후 적절한 값을 선택합니다. 다수의 사용자를 유지하되 극단적 경우를 제거하는 값을 설정하는 것이 좋습니다.</p> <p>예를 들어 <code>min__user_history_length_percentile to 0.05</code> 및 <code>max_user_history_length_percentile to 0.95</code> 를 설정하면 이력 길이가 하위 또는 상위 5%에 해당하는 사용자를 제외한 모든 사용자가 포함됩니다.</p> <p>기본 값: 0.0</p> <p>범위: [0.0, 1.0]</p> <p>값 형식: 부동 소수점</p> <p>HPO 조정 가능: 아니요</p>

명칭	Description
max_user_history_length_percentile	<p>모델 학습에 포함시킬 사용자 이력 길이의 최대 백분위수. 이력 길이는 사용자에 대한 데이터의 총 용량입니다. 이력 길이가 긴 사용자의 백분율을 제외시키려면 <code>max_user_history_length_percentile</code> 을 사용합니다. 이러한 사용자의 데이터에는 노이즈가 포함되는 경향이 있기 때문입니다. 예를 들어 로봇은 자동화된 상호작용의 긴 목록을 포함할 수 있습니다. 이러한 사용자를 제거하는 경우 학습 시 노이즈가 제한됩니다. 히스토그램 또는 유사한 도구를 사용하여 사용자 이력 길이를 검토한 후 적절한 값을 선택합니다. 다수의 사용자를 유지하되 극단적 경우를 제거하는 값을 설정하는 것이 좋습니다.</p> <p>예를 들어 <code>min__user_history_length_percentile to 0.05</code> 및 <code>max_user_history_length_percentile to 0.95</code> 를 설정하면 이력 길이가 하위 또는 상위 5%에 해당하는 사용자를 제외한 모든 사용자가 포함됩니다.</p> <p>기본 값: 0.99</p> <p>범위: [0.0, 1.0]</p> <p>값 형식: 부동 소수점</p> <p>HPO 조정 가능: 아니요</p>

HRNN-Coldstart 레시피(레거시)

Note

레거시 HRNN 레시피는 더 이상 사용할 수 없습니다. 이 설명서는 참조용입니다. 레거시 HRNN 레시피보다 `aws-user-personalization`(사용자-개인 맞춤) 레시피를 사용하는 것이 좋습니다. 사용자-개인 맞춤은 HRNN 레시피가 제공하는 기능을 개선하고 통합한 레시피입니다. 자세한 내용은 [사용자-개인 맞춤 레시피](#) 단원을 참조하세요.

새로운 항목과 상호작용을 수시로 추가하고 이러한 항목에 대한 추천을 즉시 받고 싶은 경우 HRNN-Coldstart 레시피를 사용하면 사용자가 상호작용할 항목을 예측할 수 있습니다. HRNN-Coldstart 레시피는 [HRNN-메타데이터](#) 레시피와 유사하지만 이를 통해 새 항목에 대한 추천을 받을 수 있습니다.

또한 상호작용이 매우 특이하여 학습 시 노이즈가 발생할 수 있거나, 최신 인기 추세로 인해 상호작용이 많은 학습 항목에서 제외하고자 할 때도 HRNN-Coldstart 레시피를 사용할 수 있습니다. HRNN-Coldstart를 사용하면 관련성이 적은 항목을 필터링으로 걸러내어 학습의 하위 집합을 생성할 수 있습니다. 콜드 항목이라고 하는 항목의 하위 집합은 관련된 상호 작용 이벤트가 항목 상호 작용 데이터 세트에 있는 항목입니다. 항목은 다음과 같은 경우에 콜드 항목으로 간주됩니다.

- 지정된 최대 상호작용 수보다 상호작용이 더 적은 경우. 레시피의 `cold_start_max_interactions` 하이퍼파라미터에서 이 값을 지정합니다.
- 최대 기간보다 상대 기간이 더 짧은 경우. 레시피의 `cold_start_max_duration` 하이퍼파라미터에서 이 값을 지정합니다.

콜드 항목 수를 줄이려면 `cold_start_max_interactions` 또는 `cold_start_max_duration`에 대한 값을 작게 설정합니다. 콜드 항목 수를 늘리려면 `cold_start_max_interactions` 또는 `cold_start_max_duration`에 대한 값을 크게 설정합니다.

HRNN-Coldstart의 콜드 항목 한도는 다음과 같습니다.

- Maximum cold start items: 80,000
- Minimum cold start items: 100

콜드 항목 수가 이 범위를 벗어나는 경우 솔루션 생성 시도는 실패합니다.

HRNN-Coldstart 레시피의 속성은 다음과 같습니다.

- 명칭 - `aws-hrnn-coldstart`
- 레시피 Amazon 리소스 이름(ARN) - `arn:aws:personalize:::recipe/aws-hrnn-coldstart`
- 알고리즘 ARN - `arn:aws:personalize:::algorithm/aws-hrnn-coldstart`
- 특성 변환 ARN - `arn:aws:personalize:::feature-transformation/featurize_coldstart`
- 레시피 유형 - `USER_PERSONALIZATION`

자세한 내용은 [레시피 선택](#) 섹션을 참조하세요.

다음 표에서는 HRNN-Coldstart 레시피의 하이퍼파라미터를 설명합니다. 하이퍼파라미터는 모델 성능을 향상시키기 위해 조정할 수 있는 알고리즘 파라미터입니다. 알고리즘 하이퍼파라미터는 모델이 성능을 내는 방식을 제어합니다. 특성화 하이퍼파라미터는 학습에서 사용할 데이터를 필터링하는 방법을 제어합니다. 하이퍼파라미터에 대한 최적 값을 선택하는 과정을 하이퍼파라미터 최적화(HPO)라고 합니다. 자세한 내용은 [하이퍼파라미터 및 HPO](#) 단원을 참조하세요.

표에는 각 하이퍼파라미터에 대한 다음 정보도 표시됩니다.

- 범위: [하한, 상한]
- 값 형식: 정수, 연속(부동 소수점), 범주(부울, 목록, 문자열)
- 조정 가능한 HPO: 파라미터가 HPO에 참여할 수 있습니까?

명칭	Description
알고리즘 하이퍼파라미터	
hidden_dimension	<p>모델에 사용된 숨겨진 변수 개수. 숨겨진 변수는 사용자의 구매 이력 및 항목 통계를 재생성하여 순위 점수를 생성합니다. 항목 상호 작용 데이터 세트에 더 복잡한 패턴이 포함되는 경우 더 많은 숨겨진 차원을 지정합니다. 더 많은 숨겨진 차원을 사용하는 데는 더 큰 데이터 세트와 더 많은 처리 시간이 필요합니다. 최적 값을 결정하려면 HPO를 사용합니다. HPO를 사용하려면 CreateSolution 및 CreateSolutionVersion 작업 호출 시 performHPO 를 true로 설정합니다.</p> <p>기본 값: 149</p> <p>범위: [32, 256]</p> <p>값 형식: 정수</p> <p>HPO 조정 가능: 예</p>
bptt	<p>시간 기법을 통한 역전파를 사용할지 여부를 결정합니다. 시간을 기준으로 한 역전파는 반복 신경망 기반 알고리즘의 가중치를 업데이트하는 기법입니다. 장기 크</p>

명칭	Description
	<p>레딧에 대한 bptt를 사용하여 지연된 보상을 초기 이벤트에 연결합니다. 예를 들어 지연된 보상은 여러 번 클릭 후 이루어진 구매일 수 있습니다. 초기 이벤트는 초기 클릭일 수 있습니다. 클릭과 같은 동일한 이벤트 유형 내에서도 장기 효과를 고려하고 전체 보상을 최대화하는 것이 좋습니다. 장기 효과를 고려하려면 더 큰 bptt 값을 사용합니다. 더 큰 bptt 값을 사용하는 데는 더 큰 데이터 세트와 더 많은 처리 시간이 필요합니다.</p> <p>기본 값: 32</p> <p>범위: [2, 32]</p> <p>값 형식: 정수</p> <p>HPO 조정 가능: 예</p>
recency_mask	<p>모델이 항목 상호 작용 데이터 세트의 최신 인기 트렌드를 고려해야 하는지 여부를 결정합니다. 최신 인기 트렌드에는 상호작용 이벤트 기본 패턴의 갑작스러운 변경이 포함될 수 있습니다. 최신 이벤트에 더 많은 가중치를 부여하는 모델을 학습시키려면 recency_mask 를 true로 설정합니다. 지난 모든 상호작용에 동일한 가중치를 부여하는 모델을 학습시키려면 recency_mask 를 false로 설정합니다. 동일한 가중치를 사용하여 좋은 추천을 받으려면 더 큰 학습 데이터 세트가 필요할 수 있습니다.</p> <p>기본 값: True</p> <p>범위: True 또는 False</p> <p>값 형식: 부울</p> <p>HPO 조정 가능: 예</p>

특성화 하이퍼파라미터

명칭	Description
cold_start_max_interactions	<p>항목에 대한 최대 사용자 항목 상호작용 수를 콜드 항목으로 간주해야 될 수 있습니다.</p> <p>기본 값: 15</p> <p>범위: 양의 정수</p> <p>값 형식: 정수</p> <p>HPO 조정 가능: 아니요</p>
cold_start_max_duration	<p>콜드 스타트 항목으로 간주될 사용자-항목 상호작용에 대한 시작점에 비례하는 최대 기간(일). 사용자-항목 상호작용의 시작점을 설정하려면 cold_start_relative_from 하이퍼파라미터를 설정합니다.</p> <p>기본 값: 5.0</p> <p>범위: 양의 부동 소수점</p> <p>값 형식: 부동 소수점</p> <p>HPO 조정 가능: 아니요</p>

명칭	Description
cold_start_relative_from	<p>cold_start_max_duration 을 계산하는 HRNN-Coldstart 레시피의 시작점을 결정합니다. 현재 시간부터 계산하려면 currentTime 을 선택합니다.</p> <p>항목 상호 작용 데이터 세트에 있는 최신 항목의 타임스탬프에서 cold_start_max_duration 을 계산하려면 latestItem 을 선택합니다. 이 설정은 새 항목을 수시로 추가하는 경우 유용합니다.</p> <p>기본 값: latestItem</p> <p>범위: currentTime , latestItem</p> <p>값 형식: 문자열</p> <p>HPO 조정 가능: 아니요</p>

명칭	Description
<p><code>min_user_history_length_percentile</code></p>	<p>모델 학습에 포함시킬 사용자 이력 길이의 최소 백분위수. 이력 길이는 사용자에 대한 데이터의 총 용량입니다. 이력 길이가 짧은 사용자의 백분율을 제외시키려면 <code>min_user_history_length_percentile</code> 을 사용합니다. 이력 길이가 짧은 사용자는 사용자의 개인 요구나 필요 대신에 항목 인기 기반의 패턴을 종종 보입니다. 이 패턴을 제거하면 데이터의 기본 패턴에 더 많이 집중하며 모델을 학습시킬 수 있습니다. 히스토그램 또는 유사한 도구를 사용하여 사용자 이력 길이를 검토한 후 적절한 값을 선택합니다. 다수의 사용자를 유지하되 극단적 경우를 제거하는 값을 설정하는 것이 좋습니다.</p> <p>예를 들어 <code>min__user_history_length_percentile to 0.05</code> 및 <code>max_user_history_length_percentile to 0.95</code> 를 설정하면 이력 길이가 하위 또는 상위 5%에 해당하는 사용자를 제외한 모든 사용자가 포함됩니다.</p> <p>기본 값: 0.0</p> <p>범위: [0.0, 1.0]</p> <p>값 형식: 부동 소수점</p> <p>HPO 조정 가능: 아니요</p>

명칭	Description
max_user_history_length_percentile	<p>모델 학습에 포함시킬 사용자 이력 길이의 최대 백분위수. 이력 길이는 사용자에 대한 데이터의 총 용량입니다. 이력 길이가 긴 사용자의 백분율을 제외시키려면 <code>max_user_history_length_percentile</code> 을 사용합니다. 이러한 사용자의 데이터에는 노이즈가 포함되는 경향이 있기 때문입니다. 예를 들어 로봇은 자동화된 상호작용의 긴 목록을 포함할 수 있습니다. 이러한 사용자를 제거하는 경우 학습 시 노이즈가 제한됩니다. 히스토그램 또는 유사한 도구를 사용하여 사용자 이력 길이를 검토한 후 적절한 값을 선택합니다. 다수의 사용자를 유지하되 극단적 경우를 제거하는 값을 설정하는 것이 좋습니다.</p> <p>예를 들어 <code>min__user_history_length_percentile to 0.05</code> 및 <code>max_user_history_length_percentile to 0.95</code> 를 설정하면 이력 길이가 하위 또는 상위 5%에 해당하는 사용자를 제외한 모든 사용자가 포함됩니다.</p> <p>기본 값: 0.99</p> <p>범위: [0.0, 1.0]</p> <p>값 형식: 부동 소수점</p> <p>HPO 조정 가능: 아니요</p>

AutoML을 사용하여 HRNN 레시피 선택(API만 해당)

Personalize에서는 입력 데이터 분석에 따라 가장 적합한 계층적 순환 신경망(HRNN) 레시피를 자동으로 선택할 수 있습니다. 이 옵션을 AutoML이라고 합니다. AutoML을 수행하려면 [CreateSolution](#) API를 호출할 때 `performAutoML` 파라미터를 `true`로 설정합니다.

사용자가 지정하는 지표에 따라 Personalize에서 최적 레시피를 결정하기 위해 살펴보는 레시피 목록도 사용자가 지정할 수 있습니다. 이 경우, `true` 작업을 호출하고 `recipeArn` 파라미터에 대해

performAutoML를 지정한 다음, solutionConfig파라미터를 생략하고 metricName파라미터를 포함시켜서 recipeList 및 CreateSolution을 autoMLConfig객체의 일부로 지정합니다.

다음 표에는 레시피 선택 방식이 나와 있습니다. 두 가지 모두 지정하지 말고 performAutoML 또는 recipeArn 중 하나만 지정해야 합니다. AutoML이 오로지 HRNN 레시피를 사용하여 수행됩니다.

performAutoML	recipeArn	solutionConfig	Result
true	omit	생략됨	Personalize가 레시피를 선택합니다
true	omit	autoMLConfig : metricName 및 recipeList 지정됨	Personalize가 목록에서 지표를 최적화하는 레시피 하나를 선택합니다
omit	지정됨	생략됨	사용자가 레시피를 지정합니다
omit	지정됨	지정됨	레시피를 지정하고 기본 학습 속성을 무시합니다

Note

performAutoML이 true인 경우 autoMLConfig를 제외하고 solutionConfig객체의 모든 파라미터가 무시됩니다.

POPULAR_ITEMS

속보 기사나 인기 소셜 미디어 콘텐츠 등 유행 항목이나 인기 항목을 추천하려면 POPULAR_ITEMS 레시피를 사용합니다. 사용자에게 빠르게 인기를 얻고 있는 항목에 대한 추천을 생성하려면 [지금-유행 레시피](#) 레시피를 사용합니다. 비교를 위한 기준을 생성하려면 [인기도-집계](#) 레시피를 사용하는 것이 좋습니다. 이 POPULAR_ITEMS 레시피는 상호작용을 계산하여 가장 인기 있는 항목을 추천합니다.

POPULAR_ITEMS 레시피는 다음과 같습니다.

- [지금-유행 레시피](#)
- [인기도-집계 레시피](#)

지금-유행 레시피

지금-유행 레시피(aws-trending-now)는 사용자들에게 빠르게 인기를 얻고 있는 항목에 대한 추천을 생성합니다. 인기를 얻고 있는 항목이 고객과 더 관련이 있는 경우 지금-유행 레시피를 사용할 수 있습니다. 예를 들어, 고객이 다른 사용자가 접하는 콘텐츠를 높이 평가할 수 있습니다. 일반적인 용도로는 바이럴 소셜 미디어 콘텐츠 추천, 속보 기사 또는 최근 스포츠 동영상 추천 등이 있습니다.

지금-유행은 구성 가능한 시간 간격 동안 각 항목의 상호작용 증가량을 계산하여 가장 인기 있는 항목을 자동으로 식별합니다. 증가율이 가장 높은 항목은 유행 항목으로 간주됩니다. 시간은 항목 상호작용 데이터 세트의 타임스탬프 데이터를 기반으로 합니다. 고려되는 항목은 대량으로 그리고 점진적으로 가져온 상호작용 데이터에서 나온 것입니다. 상호작용 데이터의 새 항목을 고려하기 위해 지금-유행의 새 솔루션 버전을 수동으로 만들 필요가 없습니다.

솔루션을 만들 때 Trend discovery frequency를 입력하여 시간 간격을 지정할 수 있습니다. 예를 들어 데이터를 30분마다 Trend discovery frequency에 대해 30 minutes을 지정하면 Personalize는 마지막 평가 이후 상호작용 증가율이 가장 높은 항목을 식별합니다. 가능한 빈도는 30분, 1시간, 3시간, 1일 등입니다. 상호작용 데이터의 분포에 맞는 빈도를 선택합니다. 선택한 구간 동안의 데이터가 누락되면 추천 정확도가 떨어질 수 있습니다. 지난 두 기간 동안 상호작용이 전혀 없는 항목을 가져온 경우 Personalize는 유행 항목 대신 인기 있는 항목만 추천합니다.

지금-유행을 사용하면 Personalize 콘솔의 캠페인 테스트 페이지에서 [GetRecommendations](#) 작업을 호출하거나 추천을 받을 수 있습니다. Personalize가 최고 유행 항목을 반환합니다. 필요한 필터를 적용한 경우에만 요청에 userId를 전달합니다. GetRecommendations API를 사용하면 numResults 파라미터와 함께 반환되는 트렌드 항목 수를 구성할 수 있습니다. 지금-유행 레시피로는 배치 추천을 받을 수 없습니다.

Trending-Now를 사용하려면 EventType 및 (제공된 경우) 기준으로 필터링한 후 1,000개 이상의 고유한 과거 및 이벤트 상호 작용을 결합한 항목 상호 작용 데이터세트를 만들어야 합니다. eventValueThreshold 인기 항목 추천을 생성할 때 Trending-Now는 항목 또는 사용자 데이터세트의 데이터를 사용하지 않습니다. 하지만 이들 데이터세트의 데이터를 기반으로 추천을 필터링할 수는 있습니다. 자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 단원을 참조하세요.

주제

- [속성 및 하이퍼파라미터](#)
- [솔루션 생성\(Python용 SDK\(Boto3\)\)](#)
- [샘플 Jupyter Notebook](#)

속성 및 하이퍼파라미터

지금-유행 레시피의 속성은 다음과 같습니다.

- 명칭 - aws-trending-now
- 레시피 리소스 이름(ARN) - arn:aws:personalize:::recipe/aws-trending-now
- 알고리즘 ARN - arn:aws:personalize:::algorithm/aws-trending-now-custom

자세한 내용은 [레시피 선택](#) 단원을 참조하세요.

다음 표에서는 지금-유행 레시피의 하이퍼파라미터를 설명합니다. 하이퍼파라미터는 모델 성능을 향상시키기 위해 조정할 수 있는 알고리즘 파라미터입니다. 알고리즘 하이퍼파라미터는 모델이 성능을 내는 방식을 제어합니다. 하이퍼파라미터에 대한 최적 값을 선택하는 과정을 하이퍼파라미터 최적화(HPO)라고 합니다. 자세한 내용은 [하이퍼파라미터 및 HPO](#) 단원을 참조하세요.

표에는 각 하이퍼파라미터에 대한 다음 정보도 표시됩니다.

- 범위: [하한, 상한]
- 값 형식: 정수, 연속(부동 소수점), 범주(부울, 목록, 문자열)
- 조정 가능한 HPO: 파라미터가 HPO에 참여할 수 있습니까?

명칭	설명
특성 변환 하이퍼파라미터	
Trend discovery frequency	<p>Personalize가 상호작용 데이터를 평가하고 유행 항목을 식별하는 빈도를 지정합니다. 예를 들어, Trend discovery frequency에 대해 30 minutes을 지정하면 30분마다 Personalize가 30분 간격으로 상호작용 증가율이 가장 높은 항목을 식별합니다.</p> <p>가능한 빈도로는 30분, 1시간, 3시간 및 1일이 있습니다. 상호작용 데이터의 분포에 맞는 빈도를 선택합니다. 선택한 구간 동안의 데이터가 누락되면 추천 정확도가 떨어질 수 있습니다. CreateSolution API 작업을 사용하고 값을 지정하지 않는 경우 기본값은 2시간마다입니다.</p>

명칭	설명
	기본값: 2시간
	가능한 값은 30분, 1시간, 3시간, 1일입니다.
	값 형식: 문자열
	HPO 조정 가능: 아니요

솔루션 생성(Python용 SDK(Boto3))

다음 코드는 Python용 SDK(Boto3)에서 지금-유행 레시피로 솔루션을 만드는 방법을 보여줍니다. `trend_discovery_frequency`에 가능한 값은 30 minutes, 1 hour, 3 hours 및 1 day입니다. 콘솔을 사용하여 솔루션을 생성하는 방법에 대한 자세한 내용은 [솔루션 생성\(콘솔\)](#) 단원을 참조하세요.

```
import boto3

personalize = boto3.client("personalize")

create_solution_response = personalize_client.create_solution(
    name="solution name",
    recipeArn="arn:aws:personalize:::recipe/aws-trending-now",
    datasetGroupArn="dataset group ARN",
    solutionConfig={
        "featureTransformationParameters": {
            "trend_discovery_frequency": "1 hour"
        }
    }
)

print(create_solution_response['solutionArn'])
```

샘플 Jupyter Notebook

트렌딩-나우 레시피를 사용하는 방법을 보여주는 샘플 주피터 노트북은 Amazon Personalize 샘플 리포지토리의 [trending_now_example.ipynb](#)를 참조하십시오. GitHub

인기도-집계 레시피

인기도-집계는 상호작용 데이터를 기반으로 가장 인기 있는 항목을 추천합니다. 가장 인기 있는 항목은 순 사용자의 상호작용 데이터가 가장 많은 항목입니다. 이 레시피는 모든 사용자에게 대해 동일한 인

기 항목을 반환합니다. 인기도-집계는 솔루션 버전을 생성할 때 Amazon Personalize가 생성하는 평가 지표를 사용하여 다른 레시피와 비교하기에 좋은 기준입니다. 자세한 내용은 [지표를 사용하여 솔루션 버전 평가](#) 단원을 참조하세요.

솔루션 버전을 생성한 후에는 솔루션 버전과 데이터를 최신 상태로 유지해야 합니다. Popularity-Count 를 사용하면 Amazon Personalize의 새 솔루션 버전을 수동으로 생성 (모델 재교육) 하여 추천할 새 항목을 고려하고 사용자의 최신 동작으로 모델을 업데이트해야 합니다. 그런 다음 솔루션 버전을 사용하여 모든 캠페인을 업데이트해야 합니다. 자세한 설명은 [추천 관련성 유지](#) 섹션을 참조하세요.

이 미리 정의된 레시피의 속성은 다음과 같습니다.

- 명칭 - aws-popularity-count
- 레시피 ARN - arn:aws:personalize:::recipe/aws-popularity-count
- 알고리즘 ARN - arn:aws:personalize:::algorithm/aws-popularity-count
- 특성 변환 ARN - arn:aws:personalize:::feature-transformation/sims
- 레시피 유형 - USER_PERSONALIZATION

인기도-집계에는 하이퍼파라미터가 표시되지 않습니다.

PERSONALIZED_RANKING

PERSONALIZED_RANKING 레시피는 예상 관심 수준에 따라 순위가 매겨진 순서대로 추천을 제공합니다.

맞춤형 랭킹 레시피는 다음과 같습니다.

주제

- [맞춤형 랭킹-v2 레시피](#)
- [개인 맞춤형-순위 레시피](#)

맞춤형 랭킹-v2 레시피

맞춤형 랭킹-v2 레시피는 아이템의 맞춤형 순위를 생성합니다. 개인화된 순위란 특정 사용자에게 관련성에 따라 순위가 다시 매겨진 추천 항목 목록입니다. 이 방법은 검색 결과, 프로모션 또는 선별된 목록과 같은 주문 항목이 많이 있고 각 사용자에게 개인 맞춤형 순위 재조정을 제공하려는 경우에 유용합니다.

맞춤형 랭킹-v2는 항목 상호 작용 및 항목 데이터세트에서 최대 5백만 개의 항목을 학습할 수 있습니다. 또한 보다 정확한 순위를 생성하며 지연 시간은 더 짧습니다. [개인 맞춤형-순위](#)

맞춤형 랭킹-v2를 사용하는 경우 API 작업에서 순위를 매길 항목을 지정합니다.

[GetPersonalizedRanking](#) 상호 작용 데이터가 없는 항목을 지정하는 경우 Amazon Personalize는 GetPersonalizedRanking API 응답에 권장 점수 없이 해당 항목을 반환합니다.

이 레시피는 트랜스포머 기반 아키텍처를 사용하여 컨텍스트를 학습하고 데이터의 관계 및 패턴을 추적하는 모델을 학습시킵니다. 트랜스포머는 입력 시퀀스를 출력 시퀀스로 변환하거나 변경하는 일종의 신경망 아키텍처입니다. Amazon Personalize의 경우 입력 시퀀스는 데이터에 있는 사용자의 품목 상호 작용 기록입니다. 출력 시퀀스는 개인화된 권장 사항입니다. 트랜스포머에 대한 자세한 내용은 [인공 지능의 트랜스포머란 무엇입니까?](#) 를 참조하십시오. AWS 클라우드 컴퓨팅 개념 허브에서

맞춤형 랭킹-v2는 다른 레시피와 다른 가격 책정 모델을 사용합니다. 요금에 대한 자세한 내용은 [Amazon Personalize](#) 요금을 참조하십시오.

주제

- [레시피 기능](#)
- [필수 및 선택형 데이터 세트](#)
- [속성 및 하이퍼파라미터](#)

레시피 기능

맞춤형 랭킹-v2는 아이템 순위를 매길 때 다음과 같은 Amazon Personalize 레시피 기능을 사용합니다.

- 실시간 개인화 — 실시간 개인화를 통해 Amazon Personalize는 사용자의 변화하는 관심에 따라 항목 추천을 업데이트하고 조정합니다. 자세한 정보는 [실시간 개인 맞춤](#)을 참조하세요.
- 추천이 포함된 메타데이터 — Personalized-Ranking-v2 레시피를 사용하면 메타데이터 열이 하나 이상인 항목 데이터 세트가 있는 경우 캠페인에 추천 결과에 항목 메타데이터를 포함할 수 있는 옵션이 자동으로 제공됩니다. 캠페인의 메타데이터를 수동으로 활성화할 필요는 없습니다. 메타데이터를 통해 사용자 인터페이스에서 추천을 강화할 수 있습니다(예: 캐러셀에 영화 장르 추가). 자세한 정보는 [권장 사항의 항목 메타데이터](#)을 참조하세요.

필수 및 선택형 데이터 세트

개인화된 랭킹-v2를 사용하려면 최소 1,000개의 항목 상호 작용을 [항목 상호 작용 데이터 세트](#) 생성하고 가져와야 합니다. Amazon Personalize는 주로 항목 상호 작용 데이터를 기반으로 순위를 생성합니다.

다. 맞춤형 Ranking-v2는 항목 상호 작용 및 항목 데이터 세트에서 최대 5백만 개의 항목을 학습할 수 있습니다.

맞춤형 랭킹-v2를 사용하면 Amazon Personalize는 다음을 포함하는 아이템 상호 작용 데이터를 사용할 수 있습니다.

- 이벤트 유형 및 이벤트 값 데이터 — Amazon Personalize는 클릭 또는 시청 이벤트 유형과 같은 이벤트 유형 데이터를 사용하여 행동 패턴을 통해 사용자 의도와 관심을 식별합니다. 또한 훈련 전에 이벤트 유형 및 이벤트 값 데이터를 사용하여 레코드를 필터링할 수 있습니다. 자세한 정보는 [이벤트 유형 및 이벤트 값 데이터](#)를 참조하세요.

Note

맞춤형 랭킹-v2를 사용하면 이벤트 유형이나 값을 기준으로 필터링하기 전에 상호 작용 데이터를 기반으로 훈련 비용이 책정됩니다. 요금에 대한 자세한 내용은 [Amazon Personalize](#) 요금을 참조하십시오.

- 상황별 메타데이터 — 상황별 메타데이터는 이벤트 발생 시 사용자 환경에서 수집하는 상호 작용 데이터입니다 (예: 위치 또는 디바이스 유형). 자세한 정보는 [컨텍스트 메타데이터](#)를 참조하세요.

다음 데이터세트는 선택사항이며 권장사항을 개선할 수 있습니다.

- 사용자 데이터세트 — Amazon Personalize는 사용자 데이터세트의 데이터를 사용하여 사용자와 관심사를 더 잘 이해할 수 있습니다. 또한 사용자 데이터 세트의 데이터를 사용하여 추천을 필터링할 수 있습니다. 가져올 수 있는 사용자 데이터에 대한 자세한 내용은 [사용자 데이터 세트](#) 섹션을 참조하세요.
- 아이템 데이터세트 — Amazon Personalize는 아이템 데이터세트의 데이터를 사용하여 아이템 동작의 연관성과 패턴을 식별할 수 있습니다. 이를 통해 Amazon Personalize는 사용자와 관심사를 이해할 수 있습니다. 또한 아이템 데이터세트의 데이터를 사용하여 권장 사항을 필터링할 수 있습니다. 가져올 수 있는 항목 데이터에 대한 자세한 내용은 [항목 데이터 세트](#) 섹션을 참조하세요.

속성 및 하이퍼파라미터

맞춤형 랭킹-v2 레시피에는 다음과 같은 속성이 있습니다.

- 명칭 - aws-personalized-ranking-v2
- 레시피 리소스 이름(ARN) - arn:aws:personalize:::recipe/aws-personalized-ranking-v2

- 알고리즘 ARN - `arn:aws:personalize:::algorithm/aws-personalized-ranking-v2`

자세한 정보는 [레시피 선택](#)을 참조하세요.

다음 표에는 개인화된 랭킹-v2 레시피의 하이퍼파라미터가 설명되어 있습니다. 하이퍼파라미터는 모델 성능을 향상시키기 위해 조정할 수 있는 알고리즘 파라미터입니다. 알고리즘 하이퍼파라미터는 모델이 성능을 내는 방식을 제어합니다. 하이퍼파라미터에 대한 최적 값을 선택하는 과정을 하이퍼파라미터 최적화(HPO)라고 합니다. 맞춤형 랭킹-v2를 사용할 경우 자동 교육을 켜면 Amazon Personalize는 90일마다 HPO를 자동으로 수행합니다. 자동 교육이 없으면 HPO가 발생하지 않습니다.

표에는 각 하이퍼파라미터에 대한 다음 정보가 표시됩니다.

- 범위: [하한, 상한]
- 값 형식: 정수, 연속(부동 소수점), 범주(부울, 목록, 문자열)

명칭	설명
알고리즘 하이퍼파라미터	
<code>apply_recency_bias</code>	<p>모델이 항목 상호 작용 데이터셋의 최신 항목 상호 작용 데이터에 더 많은 가중치를 부여해야 하는지 여부를 결정합니다. 가장 최근의 상호작용 데이터에는 상호작용 이벤트의 기본 패턴의 급격한 변화가 포함될 수 있습니다.</p> <p>최신 이벤트에 더 많은 가중치를 부여하는 모델을 학습시키려면 <code>apply_recency_bias</code> 를 <code>true</code>로 설정합니다. 지난 모든 상호작용에 동일한 가중치를 부여하는 모델을 학습시키려면 <code>apply_recency_bias</code> 를 <code>false</code>로 설정합니다.</p> <p>기본 값: <code>true</code></p> <p>범위: <code>true</code>또는 <code>false</code></p> <p>값 형식: 부울</p> <p>HPO 조정 가능: 아니요</p>

개인 맞춤형-순위 레시피

개인 맞춤형-순위 레시피는 항목의 개인 맞춤형 순위를 생성합니다. 개인 맞춤형 순위는 특정 사용자에 맞춰 순위가 다시 매겨진 추천 항목 목록입니다. 이 방법은 검색 결과, 프로모션 또는 선별된 목록과 같은 주문 항목이 많이 있고 각 사용자에게 개인 맞춤형 순위 재조정을 제공하려는 경우에 유용합니다. 예를 들어, 개인 맞춤형 순위를 사용하면 Amazon Personalize는 사용자가 생성한 검색 결과의 순위를 다시 매길 수 있습니다. [OpenSearch](#)

모델을 교육하기 위해 개인 맞춤형-순위 레시피는 항목 상호 작용 데이터 세트의 데이터를 사용하고, 생성한 경우 데이터 세트 그룹의 항목 데이터 세트와 사용자 데이터 세트의 데이터를 사용합니다. 두 데이터 세트는 선택 사항입니다. 개인 맞춤형-순위를 사용하면 항목 데이터 세트가 [비정형 텍스트 메타데이터](#)를, 항목 상호 작용 데이터 세트가 [컨텍스트 메타데이터](#)를 각각 포함시킬 수 있습니다. 개인 맞춤형 순위를 받으려면 [GetPersonalizedRanking](#) API를 호출합니다.

솔루션 버전을 만든 후에는 솔루션 버전과 데이터를 최신 상태로 유지해야 합니다. Personalized-Ranking을 사용하면 Amazon Personalize용 새 솔루션 버전을 수동으로 생성 (모델 재교육) 하여 추천할 새 항목을 고려하고 사용자의 최신 행동으로 모델을 업데이트해야 합니다. 그런 다음 솔루션 버전을 사용하여 모든 캠페인을 업데이트해야 합니다. 자세한 설명은 [추천 관련성 유지](#) 섹션을 참조하세요.

Note

순위를 매길 상호 작용 데이터가 없는 항목을 제공하는 경우 Amazon Personalize는 GetPersonalizedRanking API 응답에 권장 점수 없이 해당 항목을 반환합니다.

이 레시피의 속성은 다음과 같습니다.

- 명칭 - aws-personalized-ranking
- 레시피 Amazon 리소스 이름(ARN) - arn:aws:personalize:::recipe/aws-personalized-ranking
- 알고리즘 ARN - arn:aws:personalize:::algorithm/aws-personalized-ranking
- 특성 변환 ARN - arn:aws:personalize:::feature-transformation/JSON-percentile-filtering
- 레시피 유형 - PERSONALIZED_RANKING

하이퍼파라미터

다음 표에서는 개인 맞춤형-순위 레시피의 하이퍼파라미터를 설명합니다. 하이퍼파라미터는 모델 성능을 향상시키기 위해 조정할 수 있는 알고리즘 파라미터입니다. 알고리즘 하이퍼파라미터는 모델이 성능을 내는 방식을 제어합니다. 특성화 하이퍼파라미터는 학습에서 사용할 데이터를 필터링하는 방법을 제어합니다. 하이퍼파라미터에 대한 최적 값을 선택하는 과정을 하이퍼파라미터 최적화(HPO)라고 합니다. 자세한 내용은 [하이퍼파라미터 및 HPO](#) 단원을 참조하세요.

표에는 각 하이퍼파라미터에 대한 다음 정보도 표시됩니다.

- 범위: [하한, 상한]
- 값 형식: 정수, 연속(부동 소수점), 범주(부울, 목록, 문자열)
- 조정 가능한 HPO: 파라미터가 하이퍼파라미터 최적화(HPO)에 참여할 수 있습니까?

명칭	설명
알고리즘 하이퍼파라미터	
hidden_dimension	<p>모델에 사용된 숨겨진 변수 개수. 숨겨진 변수는 사용자의 구매 이력 및 항목 통계를 재생성하여 순위 점수를 생성합니다. 항목 상호 작용 데이터 세트에 더 복잡한 패턴이 포함되는 경우 더 많은 숨겨진 차원을 지정합니다. 더 많은 숨겨진 차원을 사용하는 데는 더 큰 데이터 세트와 더 많은 처리 시간이 필요합니다. 최적 값을 결정하려면 HPO를 사용합니다. HPO를 사용하려면 CreateSolution 및 CreateSolutionVersion 작업 호출 시 <code>performHPO</code> 를 <code>true</code>로 설정합니다.</p> <p>기본 값: 149</p> <p>범위: [32, 256]</p> <p>값 형식: 정수</p> <p>HPO 조정 가능: 예</p>
bptt	<p>시간 기법을 통한 역전파를 사용할지 여부를 결정합니다. 시간을 기준으로 한 역전파는 반복 신경망 기반 알고리즘의 가중치를 업데이트하는 기법입니다. 장기 크</p>

명칭	설명
	<p>레딧에 대한 bptt를 사용하여 지연된 보상을 초기 이벤트에 연결합니다. 예를 들어 지연된 보상은 여러 번 클릭 후 이루어진 구매일 수 있습니다. 초기 이벤트는 초기 클릭일 수 있습니다. 클릭과 같은 동일한 이벤트 유형 내에서도 장기 효과를 고려하고 전체 보상을 최대화하는 것이 좋습니다. 장기 효과를 고려하려면 더 큰 bptt 값을 사용합니다. 더 큰 bptt 값을 사용하는 데는 더 큰 데이터 세트와 더 많은 처리 시간이 필요합니다.</p> <p>기본 값: 32</p> <p>범위: [2, 32]</p> <p>값 형식: 정수</p> <p>HPO 조정 가능: 예</p>
recency_mask	<p>모델이 항목 상호 작용 데이터 세트의 최신 인기 트렌드를 고려해야 하는지 여부를 결정합니다. 최신 인기 트렌드에는 상호작용 이벤트 기본 패턴의 갑작스러운 변경이 포함될 수 있습니다. 최신 이벤트에 더 많은 가중치를 부여하는 모델을 학습시키려면 recency_mask 를 true로 설정합니다. 지난 모든 상호작용에 동일한 가중치를 부여하는 모델을 학습시키려면 recency_mask 를 false로 설정합니다. 동일한 가중치를 사용하여 좋은 추천을 받으려면 더 큰 학습 데이터 세트가 필요할 수 있습니다.</p> <p>기본 값: True</p> <p>범위: True 또는 False</p> <p>값 형식: 부울</p> <p>HPO 조정 가능: 예</p>

특성화 하이퍼파라미터

명칭	설명
<p><code>min_user_history_length_percentile</code></p>	<p>모델 학습에 포함시킬 사용자 이력 길이의 최소 백분위수. 이력 길이는 사용자에 대한 데이터의 총 용량입니다. 이력 길이가 짧은 사용자의 백분율을 제외시키려면 <code>min_user_history_length_percentile</code> 을 사용합니다. 이력 길이가 짧은 사용자는 사용자의 개인 요구나 필요 대신에 항목 인기 기반의 패턴을 종종 보입니다. 이 패턴을 제거하면 데이터의 기본 패턴에 더 많이 집중하며 모델을 학습시킬 수 있습니다. 히스토그램 또는 유사한 도구를 사용하여 사용자 이력 길이를 검토한 후 적절한 값을 선택합니다. 다수의 사용자를 유지하되 극단적 경우를 제거하는 값을 설정하는 것이 좋습니다.</p> <p>예를 들어 <code>min__user_history_length_percentile to 0.05</code> 및 <code>max_user_history_length_percentile to 0.95</code> 를 설정하면 이력 길이가 하위 또는 상위 5%에 해당하는 사용자를 제외한 모든 사용자가 포함됩니다.</p> <p>기본 값: 0.0</p> <p>범위: [0.0, 1.0]</p> <p>값 형식: 부동 소수점</p> <p>HPO 조정 가능: 아니요</p>

명칭	설명
max_user_history_length_percentile	<p>모델 학습에 포함시킬 사용자 이력 길이의 최대 백분위수. 이력 길이는 사용자에 대한 데이터의 총 용량입니다. 이력 길이가 긴 사용자의 백분율을 제외시키려면 <code>max_user_history_length_percentile</code> 을 사용합니다. 이러한 사용자의 데이터에는 노이즈가 포함되는 경향이 있기 때문입니다. 예를 들어 로봇은 자동화된 상호작용의 긴 목록을 포함할 수 있습니다. 이러한 사용자를 제거하는 경우 학습 시 노이즈가 제한됩니다. 히스토그램 또는 유사한 도구를 사용하여 사용자 이력 길이를 검토한 후 적절한 값을 선택합니다. 다수의 사용자를 유지하되 극단적 경우를 제거하는 값을 설정하는 것이 좋습니다.</p> <p>예를 들어 <code>min__user_history_length_percentile to 0.05</code> 및 <code>max_user_history_length_percentile to 0.95</code> 를 설정하면 이력 길이가 하위 또는 상위 5%에 해당하는 사용자를 제외한 모든 사용자가 포함됩니다.</p> <p>기본 값: 0.99</p> <p>범위: [0.0, 1.0]</p> <p>값 형식: 부동 소수점</p> <p>HPO 조정 가능: 아니요</p>

개인 맞춤형-순위 샘플 노트북

개인 맞춤형-순위 레시피를 사용하는 방법을 보여주는 Jupyter Notebook 샘플은 [순위 개인 맞춤 예제 단원을 참조하세요.](#)

RELATED_ITEMS

Note

모든 RELATED_ITEMS 레시피는 상호작용 데이터를 사용합니다. 항목 메타데이터도 있고 Amazon Personalize에서 메타데이터를 사용하여 유사한 항목을 찾으려면 유사-항목 레시피를 선택합니다. 또는 모델에 더 많은 하이퍼파라미터를 구성하려는 경우 SIMS 레시피를 선택합니다.

RELATED_ITEMS 레시피는 추천을 받을 때 지정한 항목과 유사한 항목을 반환합니다.

RELATED_ITEMS 레시피는 다음과 같습니다.

- [유사-항목 레시피](#)
- [SIMS 레시피](#)

유사-항목

유사-항목 레시피는 지정한 항목과 유사한 항목에 대한 추천을 생성합니다. 상호작용 데이터와 항목 메타데이터(제공된 경우)를 기반으로 유사도를 계산합니다. Amazon Personalize에서 추천 요청에 지정한 항목 ID를 찾을 수 없는 경우 레시피는 인기 항목을 추천으로 반환합니다. 유사-항목은 추천을 생성할 때 사용자 데이터 세트의 데이터를 사용하지 않습니다. 하지만 사용자 데이터 세트의 데이터를 기반으로 추천을 필터링할 수는 있습니다. 자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 단원을 참조하세요.

SIMS

항목-항목 유사성(SIMS) 레시피는 항목 상호 작용 데이터 세트의 사용자 기록에 있는 해당 항목의 동시 발생에 근거하여 주어진 항목과 유사한 항목들을 생성합니다. 항목에 대한 사용자 동작 데이터가 충분히 제공되지 않거나, 지정된 항목 ID를 찾을 수 없는 경우, 레시피는 인기 항목을 추천으로 반환합니다.

유사-항목 레시피

Note

모든 RELATED_ITEMS 레시피는 상호작용 데이터를 사용합니다. 항목 메타데이터도 있고 Amazon Personalize에서 메타데이터를 사용하여 유사한 항목을 찾으려면 유사-항목을

선택합니다. 또는 모델에 더 많은 하이퍼파라미터를 구성하려는 경우 [SIMS 레시피](#)를 선택합니다.

`aws-similar-items` `Similar-Items ()` 레시피는 지정한 항목과 유사한 항목에 대한 권장 사항을 생성합니다. 유사-항목을 사용하면 고객이 이전 동작 및 항목 메타데이터를 기반으로 카탈로그에서 새 항목을 발견하도록 지원할 수 있습니다. 유사한 항목을 추천하면 애플리케이션에 대한 사용자 참여, 클릭률 및 전환율을 높일 수 있습니다.

유사-항목은 상호작용 데이터와 사용자가 제공하는 모든 항목 메타데이터를 기반으로 유사성을 계산합니다. 상호작용 데이터셋 세트의 사용자 이력에서 해당 항목이 동시에 발생하는 현상과 모든 항목 메타데이터의 유사성을 고려합니다. 예를 들어, 유사-항목을 사용하면 Amazon Personalize는 고객이 비슷한 스타일([범주형 메타데이터](#))과 함께 자주 구매하는 항목을 추천하거나 다른 사용자가 비슷한 설명([비정형 텍스트 메타데이터](#))으로 시청한 영화를 추천할 수 있습니다.

유사-항목을 사용하면 [GetRecommendations](#) 작업(또는 Amazon Personalize 콘솔)에서 항목 ID를 입력하면 Amazon Personalize는 유사한 항목의 목록을 반환합니다. 또는 배치 워크플로를 사용하여 재고에 있는 모든 항목에 대해 유사한 항목을 받을 수 있습니다([배치 권장 사항 및 사용자 세그먼트\(사용자 지정 리소스\)](#) 참조). 유사한 항목을 받으면 요청에 지정한 항목의 속성을 기준으로 항목을 필터링할 수 있습니다. 이렇게 하려면 필터에 `CurrentItem.attribute` 요소를 추가합니다. 예시는 [item data filter examples](#) 단원을 참조하세요.

유사-항목을 사용하려면 1,000개 이상의 고유한 과거 상호 작용과 이벤트 상호 작용(합산)이 포함된 항목 상호 작용 데이터 세트를 만들어야 합니다. 더 정확한 예측을 위해서는 항목 데이터 세트를 만들고 카탈로그의 항목에 대한 메타데이터도 가져오는 것이 좋습니다. 유사-항목은 추천을 생성할 때 사용자 데이터세트의 데이터를 사용하지 않습니다. 여전히 사용자 데이터 세트의 데이터를 기반으로 추천을 필터링할 수 있습니다. 자세한 설명은 [추천 및 사용자 세그먼트 필터링](#) 섹션을 참조하세요.

텍스트 데이터 및 항목 제목 데이터가 포함된 항목 데이터 세트가 있는 경우 배치 추천으로 관련 항목의 테마를 생성할 수 있습니다. 자세한 정보는 [콘텐츠 생성기에서 테마가 있는 배치 추천](#)를 참조하세요.

콜드 항목(상호작용이 5회 미만인 항목)과 유사한 항목에 한 추천을 받을 수 있습니다. Amazon Personalize에서 추천 요청 또는 배치 입력 파일에 지정한 항목 ID를 찾을 수 없는 경우 레시피는 인기 항목을 추천으로 반환합니다.

솔루션 버전을 만든 후에는 솔루션 버전과 데이터를 최신 상태로 유지해야 합니다. `Similar-Items`를 사용하면 Amazon Personalize의 새 솔루션 버전을 수동으로 생성(모델 재교육)하여 추천할 새 항목을

고려하고 사용자의 최신 동작으로 모델을 업데이트해야 합니다. 그런 다음 솔루션 버전을 사용하여 모든 캠페인을 업데이트해야 합니다. 자세한 설명은 [추천 관련성 유지](#) 섹션을 참조하세요.

속성 및 하이퍼파라미터

유사-항목 레시피의 속성은 다음과 같습니다.

- 명칭 - aws-similar-items
- 레시피 Amazon 리소스 이름(ARN) - arn:aws:personalize:::recipe/aws-similar-items
- 알고리즘 ARN - arn:aws:personalize:::algorithm/aws-similar-items

자세한 설명은 [레시피 선택](#) 섹션을 참조하세요.

다음 표에서는 유사-항목 레시피의 하이퍼파라미터를 설명합니다. 하이퍼파라미터는 모델 성능을 향상시키기 위해 조정할 수 있는 알고리즘 파라미터입니다. 알고리즘 하이퍼파라미터는 모델이 성능을 내는 방식을 제어합니다. 하이퍼파라미터에 대한 최적 값을 선택하는 과정을 하이퍼파라미터 최적화(HPO)라고 합니다. 자세한 내용은 [하이퍼파라미터 및 HPO](#) 단원을 참조하세요.

표에는 각 하이퍼파라미터에 대한 다음 정보도 표시됩니다.

- 범위: [하한, 상한]
- 값 형식: 정수, 연속(부동 소수점), 범주(부울, 목록, 문자열)
- 조정 가능한 HPO: 파라미터가 HPO에 참여할 수 있습니까?

명칭	설명
알고리즘 하이퍼파라미터	
popularity_discount_factor	<p>인기도가 추천에 미치는 영향을 구성합니다. 더 인기 있는 항목을 포함시키려면 0에 가까운 값을 지정합니다. 인기도 강조를 줄이려면 1에 가까운 값을 지정합니다.</p> <p>기본 값: 0.0</p> <p>범위: [0.0, 1.0]</p> <p>값 형식: 부동 소수점</p>

명칭	설명
	HPO 조정 가능: 아니요
item_id_hidden_dim	<p>Amazon Personalize가 상호작용 데이터를 기반으로 항목 ID 임베딩을 모델링하는 데 사용하는 숨겨진 변수의 수입니다. 숨겨진 변수는 사용자의 구매 이력 및 항목 통계를 재생성하여 순위 점수를 생성합니다. <code>item_id_hidden_dim</code> 를 사용하려면 HPO를 사용하고 최소 및 최대 범위 값을 제공해야 합니다. Amazon Personalize는 HPO를 사용하여 지정한 범위 내에서 최상의 값을 찾습니다. 항목 상호 작용 데이터 세트가 큰 경우 최대 값을 더 크게 지정합니다. 더 큰 최대값을 사용하면 처리하는 데 더 많은 시간이 필요합니다.</p> <p>HPO를 사용하려면 CreateSolution 작업 호출 시 <code>performHPO</code> 를 <code>true</code>로 설정합니다.</p> <p>기본 값: 100</p> <p>범위: [30, 200]</p> <p>값 형식: 정수</p> <p>HPO 조정 가능: 예</p>

명칭	설명
item_metadata_hidden_dim	<p>Amazon Personalize에서 항목 메타데이터를 모델링하는 데 사용하는 숨겨진 변수의 수입니다. <code>item_metadata_hidden_dim</code> 를 사용하려면 HPO를 사용하고 최소 및 최대 범위 값을 제공해야 합니다. Amazon Personalize는 HPO를 사용하여 지정한 범위 내에서 최상의 값을 찾습니다. 항목 상호 작용 데이터 세트가 큰 경우 최대 값을 더 크게 지정합니다. 더 큰 최대값을 사용하면 처리하는 데 더 많은 시간이 필요합니다.</p> <p>HPO를 사용하려면 CreateSolution 작업 호출 시 <code>performHPO</code> 를 <code>true</code>로 설정합니다.</p> <p>기본 값: 100</p> <p>범위: [30, 200]</p> <p>값 형식: 정수</p> <p>HPO 조정 가능: 예</p>

SIMS 레시피

Note

모든 RELATED_ITEMS 레시피는 상호작용 데이터를 사용합니다. 모델에 더 많은 하이퍼파라미터를 구성하려는 경우 SIMS를 선택합니다. 항목 메타데이터가 있고 Amazon Personalize에서 해당 메타데이터를 사용하여 유사한 항목을 찾으려면 [유사-항목 레시피](#)를 선택합니다.

Item-to-item 유사성 (SIMS) 레시피는 공동 필터링을 사용하여 추천을 받을 때 지정한 항목과 가장 유사한 항목을 추천합니다. SIMS는 색상이나 가격 같은 항목 메타데이터가 아닌 항목 상호 작용 데이터 세트를 사용하여 유사성을 판단합니다. SIMS는 상호작용 데이터 세트의 사용자 이력에서 해당 항목이 동시에 발생한 경우를 식별하여 유사한 항목을 추천합니다. 예를 들어 Amazon Personalize는 SIMS를 사용하여 고객이 자주 함께 구매하는 커피숍 항목이나 다른 사용자가 함께 본 영화를 추천할 수 있습니다.

유사한 항목 추천을 받으면 요청에 지정한 항목의 속성을 기준으로 항목을 필터링할 수 있습니다. 이렇게 하려면 필터에 `CurrentItem.attribute` 요소를 추가합니다. 예시는 [item data filter examples](#) 단원을 참조하세요.

SIMS를 사용하려면 1,000개 이상의 고유한 과거 상호 작용과 이벤트 상호 작용(합산)이 포함된 항목 상호 작용 데이터 세트를 만들어야 합니다. SIMS는 추천을 생성할 때 사용자 또는 항목 데이터 세트의 데이터를 사용하지 않습니다. 여전히 이러한 데이터 세트의 데이터를 기반으로 추천을 필터링할 수 있습니다. 자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 단원을 참조하세요.

항목에 대한 사용자 행동 데이터가 충분하지 않거나 제공한 항목 ID를 찾을 수 없는 경우 SIMS는 인기 있는 항목을 추천합니다. 솔루션 버전을 만든 후에는 솔루션 버전과 데이터를 최신 상태로 유지해야 합니다. SIMS를 사용하면 Amazon Personalize용 새 솔루션 버전을 수동으로 생성 (모델 재교육) 하여 새 항목을 추천하고 사용자의 최신 동작으로 모델을 업데이트해야 합니다. 그런 다음 솔루션 버전을 사용하여 모든 캠페인을 업데이트해야 합니다. 자세한 설명은 [추천 관련성 유지](#) 섹션을 참조하세요.

SIMS 레시피의 속성은 다음과 같습니다.

- 명칭 - `aws-sims`
- 레시피 Amazon 리소스 이름(ARN) - `arn:aws:personalize:::recipe/aws-sims`
- 알고리즘 ARN - `arn:aws:personalize:::algorithm/aws-sims`
- 특성 변환 ARN - `arn:aws:personalize:::feature-transformation/sims`
- 레시피 유형 - `RELATED_ITEMS`

다음 표에서는 SIMS 레시피의 하이퍼파라미터를 설명합니다. 하이퍼파라미터는 모델 성능을 향상시키기 위해 조정할 수 있는 알고리즘 파라미터입니다. 알고리즘 하이퍼파라미터는 모델이 성능을 내는 방식을 제어합니다. 특성화 하이퍼파라미터는 학습에서 사용할 데이터를 필터링하는 방법을 제어합니다. 하이퍼파라미터에 대한 최적 값을 선택하는 과정을 하이퍼파라미터 최적화(HPO)라고 합니다. 자세한 내용은 [하이퍼파라미터 및 HPO](#) 단원을 참조하세요.

표에는 각 하이퍼파라미터에 대한 다음 정보도 표시됩니다.

- 범위: [하한, 상한]
- 값 형식: 정수, 연속(부동 소수점), 범주(부울, 목록, 문자열)
- 조정 가능한 HPO: 파라미터가 하이퍼파라미터 최적화(HPO)에 참여할 수 있습니까?

명칭	설명
알고리즘 하이퍼파라미터	
popularity_discount_factor	<p>인기도가 추천에 미치는 영향을 구성합니다. 더 인기 있는 항목을 포함시키려면 0에 가까운 값을 지정합니다. 인기도 강조를 줄이려면 1에 가까운 값을 지정합니다.</p> <p>기본 값: 0.5</p> <p>범위: [0.0, 1.0]</p> <p>값 형식: 부동 소수점</p> <p>HPO 조정 가능: 예</p>
min_cointeraction_count	<p>항목 한 쌍 간의 유사성을 계산하는 데 필요한 최소 공동 상호작용의 수. 예를 들어 3의 값은 유사성을 계산하기 위해 알고리즘에 대해 양쪽 항목과 상호작용한 세 명 이상의 사용자가 필요함을 의미합니다.</p> <p>기본 값: 3</p> <p>범위: [0, 10]</p> <p>값 형식: 정수</p> <p>HPO 조정 가능: 예</p>
특성화 하이퍼파라미터	
min_user_history_length_percentile	<p>모델 학습에 포함시킬 사용자 이력 길이의 최소 백분위 수. 이력 길이는 사용자에게 대해 사용 가능한 데이터의 총 용량입니다. 이력 길이가 짧은 사용자의 백분율을 제외시키려면 min_user_history_length_percentile 을 사용합니다. 이력 길이가 짧은 사용자는 사용자의 개인 요구나 필요 대신에 항목 인기 기반의 패턴을 종종 보입니다. 이 패턴을 제거하면 데이터의 기본 패턴에 더 많이 집중하며 모델을 학습시킬 수 있습니다. 히스토그램 또는 유사한 도구를 사용하여 사용자 이력</p>

명칭	설명
	<p>길이를 검토한 후 적절한 값을 선택합니다. 다수의 사용자를 유지하되 극단적 경우를 제거하는 값을 설정하는 것이 좋습니다.</p> <p>기본 값: 0.005</p> <p>범위: [0.0, 1.0]</p> <p>값 형식: 부동 소수점</p> <p>HPO 조정 가능: 아니요</p>
max_user_history_length_percentile	<p>모델 학습에 포함시킬 사용자 이력 길이의 최대 백분위 수. 이력 길이는 사용자에게 대해 사용 가능한 데이터의 총 용량입니다. 이력 길이가 짧은 사용자의 백분율을 제외시키려면 max_user_history_length_percentile 을 사용합니다. 이력 길이가 긴 사용자는 노이즈를 포함하는 경향이 있습니다. 예를 들어 로봇은 자동화된 상호작용의 긴 목록을 포함할 수 있습니다. 이러한 사용자를 제거하는 경우 학습 시 노이즈가 제한됩니다. 히스토그램 또는 유사한 도구를 사용하여 사용자 이력 길이를 검토한 후 적절한 값을 선택합니다. 다수의 사용자를 유지하되 극단적 경우를 제거하는 값을 설정하는 것이 좋습니다.</p> <p>예를 들어 min_hist_length_percentile = 0.05 및 max_hist_length_percentile = 0.95를 설정하면 이력 길이가 하위 또는 상위 5%에 해당하는 사용자를 제외한 모든 사용자가 포함됩니다.</p> <p>기본 값: 0.995</p> <p>범위: [0.0, 1.0]</p> <p>값 형식: 부동 소수점</p> <p>HPO 조정 가능: 아니요</p>

명칭	설명
min_item_interaction_count_percentile	<p>모델 학습에 포함시킬 항목 상호작용 수의 최대 백분위수. 상호작용의 이력이 짧은 항목의 백분율을 제외시키려면 min_item_interaction_count_percentile 을 사용합니다. 이력이 짧은 항목은 종종 새 항목입니다. 이 항목을 제거하면 이력이 알려진 항목에 더 집중하며 모델을 학습시킬 수 있습니다. 히스토그램 또는 유사한 도구를 사용하여 사용자 이력 길이를 검토한 후 적절한 값을 선택합니다. 다수의 사용자를 유지하되 극단적 경우를 제거하는 값을 설정하는 것이 좋습니다.</p> <p>기본 값: 0.01</p> <p>범위: [0.0, 1.0]</p> <p>값 형식: 부동 소수점</p> <p>HPO 조정 가능: 아니요</p>

명칭	설명
max_item_interaction_count_percentile	<p>모델 학습에 포함시킬 항목 상호작용 수의 최소 백분위수. 상호작용의 이력이 긴 항목의 백분율을 제외시키려면 max_item_interaction_count_percentile 을 사용합니다. 이력이 긴 항목은 오래된 경향이 있고 날짜가 만료될 수 있습니다. 예를 들어 절판된 영화 릴리스입니다. 이 항목을 제거하면 더 많은 관련 항목에 집중할 수 있습니다. 히스토그램 또는 유사한 도구를 사용하여 사용자 이력 길이를 검토한 후 적절한 값을 선택합니다. 다수의 항목을 유지하되 극단적 경우를 제거하는 값을 설정하는 것이 좋습니다.</p> <p>예를 들어 min_item_interaction_count_percentile = 0.05 및 max_item_interaction_count_percentile = 0.95 에는 상호작용 수가 하위 또는 상위 5%에 해당하는 항목을 제외한 모든 항목이 포함됩니다.</p> <p>기본 값: 0.9</p> <p>범위: [0.0, 1.0]</p> <p>값 형식: 부동 소수점</p> <p>HPO 조정 가능: 아니요</p>

SIMS 샘플 노트북

SIMS 레시피의 사용 방법을 보여주는 Jupyter Notebook 샘플은 [유사 항목 찾기 + HPO](#)를 참조하세요.

PERSONALIZED_ACTIONS

로열티 프로그램에 가입하거나, 앱을 다운로드하거나, 신용카드를 신청하는 등 사용자에게 차선책 작업을 실시간으로 추천하려면 PERSONALIZED_ACTIONS 레시피를 사용하면 됩니다. 차선책 작업을 추천하면 고객 충성도를 높이고, 더 많은 수익을 창출하며, 사용자 경험을 개선할 수 있습니다.

PERSONALIZED_ACTIONS 레시피는 다음과 같습니다.

- [차선택-작업 레시피](#)

차선택-작업 레시피

Next-Best-Action (aws-next-best-action) 레시피는 사용자를 위한 차선택을 위한 실시간 추천을 생성합니다. 사용자를 위한 차선택 작업은 사용자가 수행할 가능성이 가장 높은 작업입니다. 로열티 프로그램에 등록하거나, 앱을 다운로드하거나, 신용카드를 신청하는 경우를 예로 들 수 있습니다.

차선택-작업을 사용하면 사용자가 애플리케이션을 사용할 때 개인 맞춤형 작업 추천을 제공할 수 있습니다. 사용자에게 적합한 작업을 제안하면 더 많은 사용자가 작업을 수행하도록 유도할 수 있습니다. 추천하려는 작업에 따라 고객 충성도와 수익을 높이며 애플리케이션의 사용자 경험을 개선할 수 있습니다. 개인 맞춤형 작업 추천이 전자 상거래 애플리케이션에 어떤 이점을 줄 수 있는지를 설명하는 사용 사례는 [사용 사례 예제](#) 섹션을 참조하세요.

Amazon Personalize는 사용자가 작업 데이터 세트로 가져온 작업으로부터 차선택 작업을 예측합니다. 작업 및 항목과의 상호 작용을 기반으로 사용자가 수행할 가능성이 가장 높은 작업을 식별합니다. 작업 데이터에 작업 값이 포함된 경우 Amazon Personalize는 해당 작업의 가치를 고려합니다. 사용자가 서로 다른 두 작업을 수행할 가능성이 동일한 경우 Amazon Personalize는 더 가치 높은 작업의 순위를 보다 높게 매깁니다.

사용자에 대한 실시간 작업 추천을 받으면 Amazon Personalize는 구성 가능한 기간(action optimization period) 내에 사용자가 수행할 가능성이 가장 높은 작업 목록을 반환합니다. 향후 14일 이내에 수행할 가능성이 가장 높은 작업을 예로 들 수 있습니다. 목록은 성향 점수를 기준으로 내림차순으로 정렬됩니다. 이 점수는 사용자가 작업을 수행할 가능성을 나타냅니다.

행동 상호 작용 데이터를 가져올 때까지 Amazon Personalize는 개인화가 없는 사용자의 행동을 권장하며 성향 점수는 0.0입니다. 다음 조건을 충족하면 해당 작업에 점수가 부여됩니다.

- TAKEN 이벤트 유형과의 작업 상호 작용이 50회 이상 있어야 합니다.
- NOT_TAKEN 또는 VIEWED 이벤트 유형과의 동작 상호 작용이 50개 이상 있어야 합니다.

이러한 액션 인터랙션은 최신 솔루션 버전 교육 시 존재해야 하며, 액션 인터랙션 데이터셋의 최신 인터랙션 타임스탬프로부터 6주 이내에 발생해야 합니다.

차선택-작업 레시피가 사용하는 데이터에 대한 자세한 내용은 [필수 및 선택형 데이터 세트](#) 섹션을 참조하세요.

차선책-작업 레시피로 솔루션을 생성하는 경우 action optimization period 특성화 하이퍼파라미터를 사용하여 Amazon Personalize가 작업을 예측할 때 사용하는 시간을 구성할 수 있습니다. 자세한 정보는 [속성 및 하이퍼파라미터](#)을 참조하세요.

주제

- [사용 사례 예제](#)
- [레시피 기능](#)
- [필수 및 선택형 데이터 세트](#)
- [속성 및 하이퍼파라미터](#)

사용 사례 예제

사용자에게 적합한 작업을 제안하면 더 많은 사용자가 작업을 수행하도록 유도할 수 있습니다. 추천하려는 작업에 따라 잠재적으로 고객 충성도를 높이고, 더 많은 수익을 창출하며, 애플리케이션의 사용자 경험을 개선할 수 있습니다.

예를 들어, 다음과 같은 다양한 작업을 제안하는 전자 상거래 애플리케이션이 있을 수 있습니다.

- 로열티 프로그램 구독
- 모바일 앱 다운로드
- 주얼리 범주에서의 구매
- 뷰티 및 그루밍 범주에서의 구매

사이트에서 자주 쇼핑하면서 주얼리와 뷰티 및 그루밍 구매 작업을 반복적으로 행한 사용자가 있을 수 있습니다. 이 사용자에 대한 Amazon Personalize 작업 추천 및 점수에는 다음이 포함될 수 있습니다.

- 로열티 프로그램 구독

성향 점수 - 1.00

- 주얼리 범주에서의 구매

성향 점수 - 0.86

- 뷰티 및 그루밍 범주에서의 구매

성향 점수 - 0.85

이러한 작업 추천을 사용하면 사용자에게 로열티 프로그램에 등록하도록 유도할 수 있습니다. 이 작업은 성향 점수가 가장 높으며, 사용자가 행할 가능성이 가장 높은 작업입니다. 사용자가 매장에서 자주 쇼핑하고 로열티 프로그램의 혜택을 누릴 가능성이 높기 때문입니다.

레시피 기능

차선택-작업 레시피는 작업 추천을 생성할 때 다음과 같은 Amazon Personalize 레시피 기능을 사용합니다.

- **실시간 개인 맞춤:** Amazon Personalize는 실시간 개인 맞춤을 통해 사용자의 변화하는 관심에 따라 작업 추천을 업데이트하고 조정합니다. 자세한 정보는 [실시간 개인 맞춤](#)을 참조하세요.
- **탐색:** 탐색의 경우 추천에는 새로운 작업이나 상호 작용 데이터가 적은 작업이 포함됩니다. 탐색에 대한 자세한 내용은 [탐색](#) 단원을 참조하세요.
- **자동 업데이트:** Amazon Personalize는 자동 업데이트를 통해 2시간마다 최신 모델(솔루션 버전)을 자동으로 업데이트하여 탐색을 통한 추천에 새로운 작업을 포함합니다. 자세한 정보는 [자동 업데이트](#)를 참조하세요.

필수 및 선택형 데이터 세트

차선택-작업 레시피를 사용하려면 다음과 같은 데이터 세트를 만들어야 합니다.

- **작업:** 작업에 대한 데이터(예: 값)를 Amazon Personalize 작업 데이터 세트로 가져옵니다.

작업 데이터에서 각 작업에 대한 EXPIRATION_TIMESTAMP를 제공할 수 있습니다. 작업이 완료된 경우 Amazon Personalize는 해당 작업을 추천에 포함하지 않습니다. 각 작업에 대해 REPEAT_FREQUENCY를 제공할 수도 있습니다. 이는 Amazon Personalize에서 사용자가 작업과 상호 작용한 후 작업을 다시 추천하기 전에 기다려야 하는 시간을 나타냅니다. 작업 데이터 세트가 저장할 수 있는 데이터에 대한 자세한 내용은 [작업 데이터 세트](#) 섹션을 참조하세요.

- **항목 상호 작용:** 항목 상호 작용 데이터 세트에는 최소 1,000개의 항목 상호 작용이 있어야 합니다. Amazon Personalize는 항목 상호 작용을 통해 사용자의 현재 상태와 관심사를 이해합니다. 항목 상호 작용 데이터에 대한 자세한 내용은 [항목 상호 작용 데이터 세트](#) 섹션을 참조하세요.

다음 데이터 세트는 선택 사항입니다.

- **작업 상호 작용 데이터 세트:** 작업 상호 작용은 작업 데이터 세트에 있는 사용자와 작업을 포함하는 상호 작용입니다. 수행한 작업, 수행하지 않은 작업, 확인한 작업 상호 작용을 가져올 수 있습니다. 이 데이터는 선택 사항이지만, 품질 추천의 경우 작업 상호 작용 데이터를 가져오는 것이 좋습니다. 작

업 상호 작용 데이터가 없는 경우 빈 작업 상호 작용 데이터 세트를 만들고 [PutActionInteractions](#) API 작업을 사용하여 고객과 작업의 상호 작용을 기록할 수 있습니다.

행동 상호 작용 데이터를 가져올 때까지 Amazon Personalize는 개인화가 없는 사용자의 행동을 권장하며 성향 점수는 0.0입니다. 다음 조건을 충족하면 해당 작업에 점수가 부여됩니다.

- TAKEN 이벤트 유형과의 작업 상호 작용이 50회 이상 있어야 합니다.
- NOT_TAKEN 또는 VIEWED 이벤트 유형과의 동작 상호 작용이 50개 이상 있어야 합니다.

이러한 액션 인터랙션은 최신 솔루션 버전 교육 시 존재해야 하며, 액션 인터랙션 데이터셋의 최신 인터랙션 타임스탬프로부터 6주 이내에 발생해야 합니다.

가져올 수 있는 작업 상호 작용 데이터에 대한 자세한 내용은 [작업 상호 작용 데이터 세트](#) 섹션을 참조하세요. 작업 상호 작용 이벤트 기록에 대한 자세한 내용은 [작업 상호 작용 이벤트 기록](#) 섹션을 참조하세요.

Note

차선책-작업을 사용하면 Amazon Personalize는 작업 상호 작용 데이터 세트의 노출 데이터 또는 상황별 메타데이터를 사용하지 않습니다.

- 사용자: Amazon Personalize는 사용자 데이터 세트의 모든 데이터를 바탕으로 사용자와 관심사를 더 잘 이해합니다. 또한 사용자 데이터 세트의 데이터를 사용하여 작업 추천을 필터링할 수 있습니다. 가져올 수 있는 사용자 데이터에 대한 자세한 내용은 [사용자 데이터 세트](#) 섹션을 참조하세요.
- 항목: Amazon Personalize는 항목 상호 작용 데이터 세트와 함께 항목 데이터 세트의 모든 데이터를 사용하여 동작의 연관성과 패턴을 식별합니다. 이를 통해 Amazon Personalize는 사용자와 관심사를 이해할 수 있습니다. 가져올 수 있는 항목 데이터에 대한 자세한 내용은 [항목 데이터 세트](#) 섹션을 참조하세요.

속성 및 하이퍼파라미터

차선책-작업 레시피는 하이퍼파라미터 최적화를 지원하지 않습니다. 차선책-작업 레시피에는 다음과 같은 속성이 있습니다.

- 명칭 - `aws-next-best-action`
- 레시피 리소스 이름(ARN) - `arn:aws:personalize:::recipe/aws-next-best-action`
- 알고리즘 ARN - `arn:aws:personalize:::algorithm/aws-next-best-action`

다음 표에는 레시피의 특성화 하이퍼파라미터가 설명되어 있습니다. `aws-next-best-action` 하이퍼파라미터는 모델 성능을 향상시키기 위해 조정할 수 있는 알고리즘 파라미터입니다. 특성화 하이퍼파라미터는 학습에서 사용할 데이터를 필터링하는 방법을 제어합니다.

표에는 각 하이퍼파라미터에 대한 다음 정보도 표시됩니다.

- 범위: [하한, 상한]
- 값 형식: 정수, 연속(부동 소수점), 범주(부울, 목록, 문자열)
- 조정 가능한 HPO: 파라미터가 HPO에 참여할 수 있는지 여부

명칭	설명
특성화 하이퍼파라미터	
<code>action_optimization_period</code>	<p>Amazon Personalize가 사용자의 차선택 작업을 예측할 때 사용하는 기간입니다. 사용자가 향후 14일 이내에 수행할 가능성이 가장 높은 작업을 예로 들 수 있습니다.</p> <p>작업 상호 작용 데이터가 많지 않은 경우 더 큰 값을 지정하세요. 어떤 값을 지정해야 할지 잘 모르겠으면 기본 값을 사용하면 됩니다.</p> <p>기본값: 14</p> <p>범위: [7, 28]</p> <p>값 형식: 정수</p> <p>HPO 조정 가능: 아니요</p>

USER_SEGMENTATION

USER_SEGMENTATION 레시피는 항목 입력 데이터를 기반으로 사용자 세그먼트를 생성합니다. 각 사용자 세그먼트는 각 사용자가 인벤토리의 항목과 상호작용할 확률을 기준으로 내림차순으로 정렬됩니다. USER_SEGMENTATION 레시피를 사용하면 항목 또는 항목 속성 기본 설정에 따라 카탈로그와 상호작용할 가능성이 가장 높은 사용자 세그먼트를 생성할 수 있습니다. 예를 들어 특정 영화를 보거나 브랜드별로 특정 제품을 구매할 가능성이 높은 사용자를 대상으로 하는 타겟 마케팅 캠페인을 만들 수 있습니다.

항목-친화도

항목-친화도(aws-item-affinity) 레시피는 지정한 각 항목에 대해 사용자 세그먼트를 생성하는 USER_SEGMENTATION 레시피입니다.

항목-친화도 레시피는 모델을 학습시키는 데 데이터 세트 그룹의 상호작용 및 항목 데이터 세트를 사용합니다. [사용자 세그먼트](#)를 만들려면 항목-친화도 레시피로 솔루션 버전을 학습시킨 다음 배치 세그먼트 작업을 생성해야 합니다.

항목-속성-친화도

항목-속성-친화도(aws-item-attribute-affinity) 레시피는 지정한 각 항목 속성에 대해 사용자 세그먼트를 생성하는 USER_SEGMENTATION 레시피입니다.

항목-속성-친화도 레시피는 모델을 학습시키는 데 데이터 세트 그룹의 상호작용 데이터 세트와 항목 데이터 세트를 사용합니다. 사용자 세그먼트를 만들려면 항목-속성-친화도 레시피로 솔루션 버전을 학습시킨 다음 [배치 세그먼트 작업](#)을 생성해야 합니다.

항목-친화도 레시피

Item-Affinity (aws-item-affinity) 레시피는 지정한 각 항목에 대해 사용자 세그먼트 (사용자 그룹) 를 생성하는 USER_SEGMENTATION 레시피입니다. Personalize에서 각 항목과 상호작용할 가능성이 가장 높다고 예측하는 사용자는 다음과 같습니다. 항목-친화도를 사용하여 사용자에게 대해 자세히 알아보고 각 사용자 세그먼트를 기반으로 행동을 취합니다.

예를 들어 카탈로그의 항목에 대한 사용자 기본 설정을 기반으로 소매용 애플리케이션을 위한 마케팅 캠페인을 만들 수 있습니다. 항목-친화도는 상호작용 및 항목 데이터세트의 데이터를 기반으로 각 항목에 대한 사용자 세그먼트를 생성합니다. 이를 통해 사용자가 행동(예: 항목 클릭 또는 항목 구매)을 취할 가능성에 따라 여러 사용자 세그먼트에 여러 항목을 홍보할 수 있습니다. 다른 용도로는 여러 사용자에게 제품을 교차 판매하거나 예비 구직자를 식별하는 것이 포함될 수 있습니다.

항목을 기반으로 사용자 세그먼트를 가져오려면 항목-친화도 레시피로 솔루션과 솔루션 버전을 생성한 다음 JSON 형식의 항목 목록을 S3 버킷에 추가하고 [배치 세그먼트 작업](#)을 생성합니다. Personalize는 각 항목에 대한 사용자 세그먼트를 S3의 출력 위치로 출력합니다. 입력 데이터는 사용자 세그먼트를 가져올 항목 수가 최대 500개일 수 있습니다. 배치 세그먼트 작업을 위한 입력 데이터 준비에 대한 자세한 내용은 [배치 추천을 위한 입력 데이터 준비단원](#)을 참조하세요.

항목-친화도를 사용하려면 항목 상호 작용 데이터 세트가 있어야 합니다. 항목 및 사용자 데이터세트는 선택사항입니다. 배치 세그먼트 작업으로 사용자 세그먼트를 가져올 수 있습니다. 자세한 내용은 [배치 권장 사항 및 사용자 세그먼트\(사용자 지정 리소스\)](#) 단원을 참조하세요.

솔루션 버전을 만든 후에는 솔루션 버전과 데이터를 최신 상태로 유지해야 합니다. Item-Affinity를 사용하면 Amazon Personalize용 새 솔루션 버전을 생성하여 사용자 세그먼트의 새 사용자를 고려하고 사용자의 최신 행동으로 모델을 업데이트해야 합니다. 항목에 대한 사용자 세그먼트를 가져오려면 솔루션 버전을 생성할 때 항목이 존재했어야 합니다.

항목-친화도 레시피의 속성은 다음과 같습니다.

- 명칭 - `aws-item-affinity`
- 레시피 리소스 이름(ARN) - `arn:aws:personalize:::recipe/aws-item-affinity`
- 알고리즘 ARN - `arn:aws:personalize:::algorithm/aws-item-affinity`
- 특성 변환 ARN - `arn:aws:personalize:::feature-transformation/item-affinity`
- 레시피 유형 - `USER_SEGMENTATION`

다음 표에서는 항목-친화도 레시피의 하이퍼파라미터를 설명합니다. 하이퍼파라미터는 모델 성능을 향상시키기 위해 조정할 수 있는 알고리즘 파라미터입니다. 알고리즘 하이퍼파라미터는 모델이 성능을 내는 방식을 제어합니다. 항목-친화성 레시피에는 하이퍼파라미터 최적화(HPO)를 사용할 수 없습니다.

표에는 각 하이퍼파라미터에 대한 다음 정보도 표시됩니다.

- 범위: [하한, 상한]
- 값 형식: 정수, 연속(부동 소수점), 범주(부울, 목록, 문자열)

명칭	설명
알고리즘 하이퍼파라미터	
<code>hidden_dimension</code>	<p>모델에 사용된 숨겨진 변수 개수. 숨겨진 변수는 사용자의 구매 이력 및 항목 통계를 재생성하여 순위 점수를 생성합니다. 상호작용 데이터세트에 더 복잡한 패턴이 포함되는 경우 더 많은 숨겨진 차원을 지정합니다. 더 많은 숨겨진 차원을 사용하는 데는 더 큰 데이터세트와 더 많은 처리 시간이 필요합니다.</p> <p>기본 값: 149</p> <p>범위: [32, 256]</p>

명칭	설명
	값 형식: 정수

항목-속성-친화도 레시피

항목-속성-선호도 (aws-item-attribute-affinity) 레시피는 지정한 각 항목 속성에 대해 사용자 세그먼트 (사용자 그룹) 를 생성하는 USER_SEGMENTATION 레시피입니다. Personalize가 예측하는 사용자는 특정 속성을 가진 항목과 상호작용할 가능성이 가장 높은 사용자입니다. 항목-속성-친화도를 사용하여 사용자에게 대해 자세히 알아보고 해당 사용자 세그먼트를 기반으로 행동을 취합니다.

예를 들어 카탈로그의 신발 유형에 대한 사용자 선호도를 기반으로 소매용 애플리케이션을 위한 마케팅 캠페인을 만들 수 있습니다. 항목-속성-친화도는 상호작용 및 항목 데이터세트의 데이터를 기반으로 각 신발 유형에 대한 사용자 세그먼트를 생성합니다. 이를 통해 사용자가 행동(예: 신발 클릭 또는 신발 구매)을 취할 가능성에 따라 다양한 사용자 세그먼트에 다양한 신발을 홍보할 수 있습니다. 다른 용도로는 다양한 사용자에게 다양한 영화 장르를 홍보하거나 직무 유형에 따라 예비 구직자를 식별하는 것이 포함될 수 있습니다.

항목 속성을 기반으로 사용자 세그먼트를 가져오려면 항목-속성-친화도 레시피로 솔루션과 솔루션 버전을 생성한 다음 JSON 형식의 항목 속성 목록을 S3 버킷에 추가하고 [배치 세그먼트 작업](#)을 생성합니다. Personalize는 각 항목에 대한 사용자 세그먼트를 S3의 출력 위치로 출력합니다. 입력 데이터에 최대 10개의 쿼리가 포함될 수 있으며, 각 쿼리는 하나 이상의 항목 속성입니다. 배치 세그먼트 작업을 위한 입력 데이터 준비에 대한 자세한 내용은 [배치 추천을 위한 입력 데이터 준비](#) 단원을 참조하세요.

항목-속성-친화도를 사용하려면 항목 상호 작용 데이터 세트와 항목 데이터 세트가 있어야 합니다. 항목 데이터세트에는 열(텍스트가 아닌 예약되지 않은 메타데이터 열)이 하나 이상 있어야 합니다. 배치 세그먼트 작업으로 사용자 세그먼트를 가져올 수 있습니다. 자세한 내용은 [배치 권장 사항 및 사용자 세그먼트\(사용자 지정 리소스\)](#) 단원을 참조하세요.

솔루션 버전을 만든 후에는 솔루션 버전과 데이터를 최신 상태로 유지해야 합니다. Item-Attribute-Affinity를 사용하면 Amazon Personalize용 새 솔루션 버전을 생성하여 사용자 세그먼트의 새 사용자를 고려하고 사용자의 최신 행동으로 모델을 업데이트해야 합니다. 항목 속성에 대한 사용자 세그먼트를 가져오려면 솔루션 버전을 생성할 때 항목 속성이 존재했어야 합니다.

항목-속성-친화도 레시피의 속성은 다음과 같습니다.

- 명칭 - aws-item-attribute-affinity
- 레시피 리소스 이름(ARN) - arn:aws:personalize:::recipe/aws-item-attribute-affinity

- 알고리즘 ARN - `arn:aws:personalize:::algorithm/aws-item-attribute-affinity`
- 특성 변환 ARN - `arn:aws:personalize:::feature-transformation/item-attribute-affinity`
- 레시피 유형 - `USER_SEGMENTATION`

다음 표에서는 항목-속성-친화도 레시피의 하이퍼파라미터를 설명합니다. 하이퍼파라미터는 모델 성능을 향상시키기 위해 조정할 수 있는 알고리즘 파라미터입니다. 알고리즘 하이퍼파라미터는 모델이 성능을 내는 방식을 제어합니다. 항목-속성-친화도 레시피에는 하이퍼파라미터 최적화(HPO)를 사용할 수 없습니다.

표에는 각 하이퍼파라미터에 대한 다음 정보도 표시됩니다.

- 범위: [하한, 상한]
- 값 형식: 정수, 연속(부동 소수점), 범주(부울, 목록, 문자열)

명칭	설명
알고리즘 하이퍼파라미터	
hidden_dimension	<p>모델에 사용된 숨겨진 변수 개수. 숨겨진 변수는 사용자의 구매 이력 및 항목 통계를 재생성하여 순위 점수를 생성합니다. 상호작용 데이터세트에 더 복잡한 패턴이 포함되는 경우 더 많은 숨겨진 차원을 지정합니다. 더 많은 숨겨진 차원을 사용하는 데는 더 큰 데이터세트와 더 많은 처리 시간이 필요합니다.</p> <p>기본 값: 149</p> <p>범위: [32, 256]</p> <p>값 형식: 정수</p>

준비 체크리스트

Personalize의 작동 방식을 검토하고 시작하기 연습을 완료한 후에는 자체 데이터로 Personalize를 사용할 준비를 시작할 수 있습니다. 이 체크리스트는 Personalize 기능, 요구 사항 및 데이터 지침 목록을 제공합니다. 계획을 세우는 데 도움이 될 수도 있고 Personalize에서 리소스를 생성할 때 참조로 사용할 수도 있습니다.

주제

- [사용 사례를 Personalize 리소스와 일치시켰습니까?](#)
- [항목 상호 작용 데이터가 충분합니까?](#)
- [실시간 이벤트 스트리밍 아키텍처가 마련되어 있습니까?](#)
- [데이터가 Personalize에 최적화되어 있습니까?](#)
- [추천을 개선할 수 있는 선택적 데이터를 수집합니까?](#)
- [추천을 테스트할 계획이 있으십니까?](#)
- [추가 비즈니스 목표가 있나요?](#)

사용 사례를 Personalize 리소스와 일치시켰습니까?

Personalize 추천은 다음과 같은 사용 사례를 해결할 수 있습니다.

- 사용자를 위한 개인 맞춤형 추천 생성
- 유사 항목 또는 관련 항목 추천
- 유행 항목 또는 인기 항목 추천
- 사용자를 위한 차선책 작업 추천
- 관련성에 따른 재정렬(사용자 지정 리소스만 해당)
- 사용자 세그먼트 생성(사용자 지정 리소스만 사용)

Personalize는 도메인 기반 리소스와 이러한 사용 사례에 맞게 구성된 사용자 지정 리소스를 제공합니다. 먼저, 도메인 데이터세트 그룹 또는 사용자 지정 데이터세트 그룹을 다음과 같이 생성합니다.

- 도메인 데이터세트 그룹을 사용하여 VIDEO_ON_DEMAND 또는 ECOMMERCE 도메인에 맞게 사전 구성 및 최적화된 리소스를 생성합니다.

스트리밍 비디오 또는 전자 상거래 애플리케이션이 있는 경우, 도메인 데이터세트 그룹으로 시작하는 것이 좋습니다. 사용자 지정 사용 사례에 맞게 학습된 솔루션 및 솔루션 버전과 같은 사용자 지정 리소스는 계속해서 추가할 수 있습니다. 또한 사용자 지정 리소스를 사용하여 일괄 추천을 받을 수도 있습니다. 도메인 데이터세트 그룹에는 액션 및 액션 인터랙션 데이터세트를 비롯한 차선택 액션 리소스를 만들 수 없습니다.

- 사용자 지정 데이터세트 그룹을 사용하여 사용 사례에 맞는 레시피를 선택합니다. 그런 다음 구성 가능한 솔루션과 솔루션 버전(학습된 Personalize 추천 모델)만 학습하고 배포합니다. 준비가 되면 캠페인에 솔루션 버전을 배포하여 실시간 추천을 받을 수 있습니다. 또는 캠페인 없이 일괄 추천을 받을 수도 있습니다.

스트리밍 비디오 또는 전자 상거래 애플리케이션이 없는 경우 사용자 지정 데이터세트 그룹을 생성하는 것이 좋습니다. 그렇지 않으면 도메인 데이터세트 그룹으로 시작하되 필요에 따라 사용자 지정 리소스를 추가합니다.

Personalize에서 사용할 수 있는 사용 사례 및 사용자 지정 레시피에 대한 자세한 내용은 [도메인 사용 사례 및 사용자 지정 레시피](#) 단원을 참조하세요.

항목 상호 작용 데이터가 충분합니까?

모든 사용 사례와 레시피에 대해 고유 사용자 25명의 상호 작용이 각각 2회 이상 포함된 최소 1,000개의 항목 상호 작용이 있어야 합니다. 품질 추천의 경우, 각 2번 이상의 항목 상호 작용을 갖는 1,000명 이상의 사용자로부터 50,000건 이상의 항목 상호 작용을 갖는 것이 좋습니다.

데이터가 충분한지 확실하지 않은 경우 Personalize 콘솔을 사용하여 데이터를 가져와서 분석할 수 있습니다. 자세한 내용은 [데이터셋의 데이터 품질 및 수량 분석](#) 단원을 참조하세요.

실시간 이벤트 스트리밍 아키텍처가 마련되어 있습니까?

항목 상호 작용 데이터가 충분하지 않은 경우 Amazon Personalize를 사용하여 추가 실시간 이벤트 데이터를 수집할 수 있습니다. 일부 레시피와 사용 사례를 통해 Personalize는 사용자가 애플리케이션을 사용할 때 사용자의 최신 활동에서 학습하고 추천을 업데이트할 수 있습니다.

이벤트가 추천에 미치는 영향, 타사 이벤트 추적 서비스 목록, 샘플 구현을 비롯한 이벤트 기록에 대한 자세한 내용은 [이벤트 기록](#) 단원을 참조하세요.

데이터가 Personalize에 최적화되어 있습니까?

데이터에서 다음 사항을 확인하는 것이 좋습니다.

- 누락된 값이 있는지 확인합니다. 레코드 중 최소 70%에 모든 속성에 대한 데이터를 포함시키는 것이 좋습니다. null 값을 70% 이상 완료할 수 있는 열을 사용하는 것이 좋습니다.
- 일관되지 않은 명명 규칙, 항목의 중복 범주, 데이터세트의 ID 불일치, ID 중복과 같은 데이터의 부정확성이나 문제를 수정합니다. 이러한 문제는 추천에 부정적인 영향을 미치거나 예기치 않은 동작으로 이어질 수 있습니다. 예를 들어 데이터에 "N/A"와 "해당 없음"이 모두 있지만 "N/A"만 기준으로 추천을 필터링할 수 있습니다. "해당 없음"으로 표시된 항목은 필터를 통해 제거되지 않습니다.
- 항목, 사용자 또는 작업이 여러 범주를 가질 수 있는 경우(예: 장르가 여러 개 있는 영화) 범주별 값을 하나의 속성으로 결합하고 | 연산자를 사용하여 각 값을 구분합니다. 예를 들어 영화의 GENRES 데이터는 액션 | 어드벤처 | 스릴러일 수 있습니다.
- 컬럼에 사용할 수 있는 카테고리가 1000개를 넘지 않도록 하세요 (컬럼에 필터링 목적으로만 사용되는 데이터가 포함된 경우 제외).

데이터 추천의 전체 목록과 Personalize를 사용하여 문제를 식별하는 방법에 대한 지침은 [데이터셋의 데이터 품질 및 수량 분석](#) 단원을 참조하세요.

추천을 개선할 수 있는 선택적 데이터를 수집합니까?

다음 데이터는 추천 관련성을 개선하는 데 도움이 될 수 있습니다.

- 이벤트 유형(모든 도메인 데이터세트 그룹 사용 사례에 필요)
- 이벤트 값
- 컨텍스트 메타데이터
- 항목 및 사용자 메타데이터
- 작업 상호 작용 데이터(PERSONALIZED_ACTIONS 레시피에서만 사용)

Personalize에서 사용할 수 있는 데이터 유형에 대한 자세한 내용은 [Personalize가 사용할 수 있는 데이터 유형](#) 단원을 참조하세요.

추천을 테스트할 계획이 있으십니까?

A/B 테스트를 사용하여 여러 사용자 그룹이 서로 다른 모델의 추천과 상호작용한 결과를 비교할 수 있습니다. A/B 테스트를 통해 다양한 추천 전략을 비교하고 추천이 비즈니스 목표를 달성하는 데 도움이 되는지 확인할 수 있습니다. 자세한 내용은 [A/B 테스트를 통한 권장 사항 영향 측정](#) 단원을 참조하세요.

추가 비즈니스 목표가 있나요?

사용자에게 적절한 추천을 제공하는 것 외에도 목표를 세워야 하는 경우도 있습니다. 예를 들어 수익을 극대화하거나 특정 범주의 특정 유형의 항목을 홍보하고 싶을 수 있습니다. 다음과 같은 Personalize 기능이 도움이 될 수 있습니다.

- **홍보:** 홍보를 사용하여 특정 비율의 항목이 비즈니스 요구 사항을 충족하는지 확인할 수 있습니다. 자세한 내용은 [추천의 항목 홍보](#) 단원을 참조하세요.
- **비즈니스 목표 최적화:** 일부 사용자 지정 데이터세트 그룹 레시피의 경우 스트리밍 시간 최대화 또는 수익 증대와 같은 사용자 지정 목표에 맞게 솔루션을 최적화할 수 있습니다. 자세한 내용은 [추가 목표에 맞게 솔루션 최적화](#) 단원을 참조하세요.
- **추천 필터링:** 필터를 사용하여 비즈니스 규칙을 추천에 적용합니다. 필터를 사용하여 추천에 특정 유형의 항목을 포함시키거나 제외시킬 수 있습니다. 자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 단원을 참조하세요.

Personalize 워크플로우

[준비 체크리스트](#)의 검토를 마치게 되면 Personalize 워크플로를 완료할 준비가 된 것입니다.

1. 데이터세트 그룹 생성

데이터세트 그룹은 Personalize 리소스를 위한 컨테이너입니다. 생성하는 데이터세트 그룹 유형에 따라 Personalize 워크플로의 3단계에서 생성할 수 있는 리소스가 결정됩니다.

- 도메인 데이터세트 그룹을 사용하면 VIDEO_ON_DEMAND 또는 ECOMMERCE 도메인 사용 사례에 맞게 구성된 추천자를 생성할 수 있습니다. 추천자를 사용하여 추천을 받을 수 있습니다. Personalize는 추천자의 구성, 학습 및 업데이트를 관리합니다. 도메인 데이터세트 그룹으로 시작하는 경우에도 사용자 지정 리소스를 추가할 수 있습니다. 도메인 데이터세트 그룹에서는 액션 및 액션 인터랙션 데이터세트를 비롯한 차선책 액션 리소스를 만들 수 없습니다.
- 사용자 지정 데이터세트 그룹을 사용하면 사용자 지정 리소스만 생성할 수 있습니다. 여기에는 솔루션, 솔루션 버전, 캠페인이 포함됩니다. 이들 리소스의 경우 구성, 업데이트 및 재학습을 더 잘 제어할 수 있습니다.

2. 데이터 준비 및 가져오기

항목 상호 작용, 작업 상호 작용, 항목, 사용자, 작업 레코드를 데이터 세트(데이터용 Amazon Personalize 컨테이너)로 가져옵니다. 레코드를 대량으로 또는 개별적으로 가져올 수 있습니다. 대량 데이터를 가져올 때 Amazon 데이터 랭글러를 사용하여 40개 이상의 소스에서 SageMaker 데이터를 가져와서 Amazon Personalize에 사용할 수 있도록 준비할 수 있습니다. 자세한 정보는 [Amazon SageMaker 데이터 랭글러를 사용하여 데이터 준비 및 가져오기](#)를 참조하세요.

Personalize 데이터세트로 데이터를 가져온 후에는, 데이터를 분석하고, S3 버킷으로 내보내고, 업데이트하거나, 데이터세트를 삭제하여 데이터를 삭제할 수 있습니다. 자세한 내용은 [데이터셋의 훈련 데이터 관리](#) 단원을 참조하세요.

3. 도메인 추천자 또는 사용자 지정 리소스 생성

데이터를 가져온 후에는 도메인 추천자(도메인 데이터세트 그룹의 경우) 또는 사용자 지정 리소스(사용자 지정 데이터세트 그룹의 경우)를 만들어 데이터에 따라 모델을 학습시킵니다. 이들 리소스를 사용하여 추천을 생성할 수 있습니다.

4. 추천 받기

추천자 또는 사용자 지정 캠페인을 사용하여 추천을 받습니다. 사용자 지정 데이터세트 그룹을 사용하면 배치 추천 또는 사용자 세그먼트를 가져올 수도 있습니다.

Personalize 워크플로를 처음으로 완료한 후에는 데이터를 최신 상태로 유지하고 모든 사용자 지정 솔루션을 정기적으로 재학습시킵니다. 이를 통해 모델은 사용자의 가장 최근 활동을 학습하고 추천의 관련성을 유지하고 개선할 수 있습니다. 자세한 내용은 [추천 관련성 유지](#) 단원을 참조하세요.

1단계: 데이터세트 그룹 생성

Personalize를 사용하기 시작하면 데이터세트 그룹을 생성합니다. 데이터세트 그룹은 데이터세트, 도메인 추천, 사용자 지정 리소스를 비롯한 Personalize 리소스의 컨테이너입니다. 높으면 요금이 인상됩니다. 데이터세트 그룹에서는 리소스가 독립적인 컬렉션으로 정리되어 있는데, 한 데이터세트 그룹의 리소스는 다른 데이터세트 그룹의 리소스에 영향을 미칠 수 없습니다.

각 비즈니스 도메인에 대한 데이터세트 그룹을 생성합니다. 예를 들면, 비디오 스트리밍을 위한 추천 항목을 제공하는 애플리케이션과 오디오 북을 추천하는 애플리케이션이 있을 수 있습니다. Personalize에서는 각 애플리케이션에 대한 데이터세트 그룹을 생성합니다. 이렇게 하면 한 애플리케이션의 데이터가 Personalize에서 다른 애플리케이션에 대해 생성하는 추천에 영향을 주지 않습니다.

도메인 데이터세트 그룹 또는 사용자 지정 데이터세트 그룹을 다음과 같이 생성할 수 있습니다.

- 도메인 데이터세트 그룹을 사용하여 다양한 사용 사례에 맞추어 사전 구성 및 최적화된 리소스를 생성합니다. 데이터세트 그룹을 생성할 때는 VIDEO_ON_DEMAND 또는 ECOMMERCE의 도메인을 지정하여 해당 그룹을 도메인 데이터세트 그룹으로 만듭니다.

스트리밍 비디오 또는 전자 상거래 애플리케이션이 있는 경우, 도메인 데이터세트 그룹을 생성하는 것이 좋습니다. 사용자 지정 사용 사례에 맞게 학습된 솔루션 및 솔루션 버전과 같은 사용자 지정 리소스는 계속해서 추가할 수 있습니다. 도메인 데이터세트 그룹에서는 액션 및 액션 인터랙션 데이터 세트를 비롯한 차선택 액션 리소스를 만들 수 없습니다.

- 사용자 지정 데이터세트 그룹에는 사용 사례에 따라 구성하는 사용자 지정 리소스만 포함됩니다. 사용자 지정 리소스를 사용하면 비즈니스 니즈에 따라 구성 가능한 솔루션 및 솔루션 버전(교육된 Personalize 권장 모델)을 교육 및 배포할 수 있습니다. VIDEO_ON_DEMAND 또는 ECOMMERCE 애플리케이션이 없는 경우, 사용자 지정 데이터세트 그룹을 생성하는 것이 좋습니다. 그렇지 않으면 도메인 데이터세트 그룹으로 시작하여 필요에 따라 사용자 지정 리소스를 추가하는 것이 좋습니다.

Amazon Personalize 콘솔 AWS Command Line Interface ,AWS CLI() 또는 SDK를 사용하여 데이터세트 그룹을 생성할 수 있습니다. AWS

주제

- [데이터세트 그룹 생성\(콘솔\)](#)

- [데이터세트 그룹 생성\(AWS CLI\)](#)
- [데이터세트 그룹 \(SDK\) 생성AWS](#)

데이터세트 그룹 생성(콘솔)

Personalize 콘솔에서 데이터세트 그룹의 이름을 지정하여 데이터세트 그룹을 생성합니다.

데이터세트 그룹을 생성하려면

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터세트 그룹 생성을 선택합니다.
3. Personalize를 처음 사용하는 경우, 데이터세트 그룹 생성 페이지의 새 데이터세트 그룹에서 시작하기를 선택합니다.
4. 데이터세트 그룹 세부 정보의 데이터세트 그룹 이름에서 데이터세트 그룹의 이름을 지정합니다.
5. 도메인을 선택합니다.
 - 전자상거래를 선택하여 ECOMMERCE 도메인 데이터세트 그룹을 생성합니다.
 - 온디맨드 비디오를 선택하여 VIDEO_ON_DEMAND 도메인 데이터세트 그룹을 생성합니다.
 - 사용자 지정을 선택하여 솔루션, 캠페인, 배치 추론 작업과 같은 사용자 지정 리소스만 포함된 사용자 지정 데이터세트 그룹을 생성합니다.
6. 태그의 경우, 원하는 태그를 추가할 수 있습니다. Personalize 리소스 태그 지정에 대한 자세한 내용은 [Personalize 리소스에 태그 지정](#) 단원을 참조하세요.
7. 데이터세트 그룹 생성을 선택합니다. 개요 페이지가 표시됩니다. 이제 데이터를 가져올 준비가 되었습니다. [단계 2: 데이터 준비 및 가져오기](#) 단원을 참조하세요.

데이터세트 그룹 생성(AWS CLI)

데이터세트 그룹을 생성하려면 create-dataset-group 작업을 사용하세요. 도메인 데이터세트 그룹을 생성하려면 도메인에 대해 ECOMMERCE 또는 VIDEO_ON_DEMAND를 지정합니다. 사용자 지정 데이터세트 그룹을 생성하려면 도메인을 지정합니다. 태그 파라미터를 사용하여 원하는 대로 Personalize에서 리소스에 태그를 지정할 수 있습니다. 샘플은 [태그 추가\(AWS CLI\)](#) 단원을 참조하세요.

다음 코드는 VIDEO_ON_DEMAND 도메인에 대한 도메인 데이터세트 그룹을 생성합니다.

```
aws personalize create-dataset-group \
--name dataset-group-name \
--domain VIDEO_ON_DEMAND
```

성공할 경우, 데이터세트 그룹 리소스 이름(ARN)이 다음과 같이 표시됩니다.

```
{
  "datasetGroupArn": "arn:aws:personalize:us-west-2:acct-id:dataset-group/
DatasetGroupName"
}
```

추후 사용을 위해 이 값을 기록하세요. 생성한 데이터세트 그룹을 표시하려면 `describe-dataset-group` 명령을 사용하고 반환되는 데이터세트 그룹 ARN을 지정합니다.

```
aws personalize describe-dataset-group \
--dataset-group-arn dataset group arn
```

데이터세트 그룹과 해당 속성은 다음과 같이 표시됩니다.

```
{
  "datasetGroup": {
    "name": "DatasetGroupName",
    "datasetGroupArn": "arn:aws:personalize:us-west-2:acct-id:dataset-group/
DatasetGroupName",
    "status": "ACTIVE",
    "creationDateTime": 1542392161.262,
    "lastUpdatedDateTime": 1542396513.377
  }
}
```

데이터세트 그룹의 `status`가 활성 상태이면 데이터를 가져올 준비가 된 것입니다. 자세한 정보는 [단계 2: 데이터 준비 및 가져오기](#)를 참조하세요.

데이터세트 그룹 (SDK) 생성AWS

다음 코드를 사용하여 도메인 데이터세트를 생성합니다. 도메인 데이터세트 그룹에 이름을 지정하고 `domain`에 대해 `ECOMMERCE` 또는 `VIDEO_ON_DEMAND`를 지정합니다. 사용자 지정 데이터세트 그룹을 생성하려면 코드를 수정하여 도메인 파라미터를 제거합니다.

API 작업에 대한 자세한 내용은 API 참조 단원에서 [CreateDatasetGroup](#)을 참조하세요. 태그 파라미터를 사용하여 원하는 대로 Personalize에서 리소스에 태그를 지정할 수 있습니다. 샘플은 [태그 추가 \(AWS SDK\)](#) 단원을 참조하세요.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_dataset_group(
    name = 'dataset group name',
    domain = 'business domain'
)
dsg_arn = response['datasetGroupArn']

description = personalize.describe_dataset_group(datasetGroupArn = dsg_arn)
['datasetGroup']

print('Name: ' + description['name'])
print('ARN: ' + description['datasetGroupArn'])
print('Status: ' + description['status'])
```

SDK for Java 2.x

```
public static String createDomainDatasetGroup(PersonalizeClient personalizeClient,
                                             String datasetGroupName,
                                             String domain) {

    try {
        CreateDatasetGroupRequest createDatasetGroupRequest =
        CreateDatasetGroupRequest.builder()
            .name(datasetGroupName)
            .domain(domain)
            .build();

        return
        personalizeClient.createDatasetGroup(createDatasetGroupRequest).datasetGroupArn();
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { CreateDatasetGroupCommand } from
  "@aws-sdk/client-personalize";
import { personalizeClient } from "../libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the domain dataset group parameters.
export const domainDatasetGroupParams = {
  name: 'NAME', /* required */
  domain: 'DOMAIN' /* required for a domain dsG, specify ECOMMERCE or
  VIDEO_ON_DEMAND */
}

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
    CreateDatasetGroupCommand(domainDatasetGroupParams));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

[DescribeDatasetGroup](#) 작업은 datasetGroupArn 및 작업의 상태를 반환합니다. 데이터세트 그룹의 status가 활성 상태이면 데이터를 가져올 준비가 된 것입니다. 자세한 내용은 [단계 2: 데이터 준비 및 가져오기](#) 단원을 참조하세요.

단계 2: 데이터 준비 및 가져오기

Amazon Personalize는 데이터를 사용하여 사용자 및 사용자 세그먼트에 대한 권장 사항을 생성합니다. Amazon Personalize는 사용자가 데이터 세트를 삭제할 때까지 데이터 세트에 데이터를 저장합니다. 모든 사용 사례(도메인 데이터 세트 그룹) 및 레시피(사용자 지정 리소스)에 대해 상호작용 데이터에는 다음이 포함되어야 합니다.

- 카탈로그의 항목과 상호 작용하는 사용자의 1,000개 이상의 항목 상호 작용 기록. 이러한 상호 작용은 대량 가져오기, 스트리밍된 이벤트 또는 이 둘 다에서 발생할 수 있습니다.
- 각각 2번 이상의 항목 상호 작용을 갖는 25개 이상의 고유 사용자 ID.

품질 추천의 경우, 각 2번 이상의 항목 상호 작용을 갖는 1,000명 이상의 사용자로부터 50,000건 이상의 항목 상호 작용을 갖는 것이 좋습니다.

데이터를 가져올 때 레코드를 대량으로, 개별적으로 또는 둘 모두의 방법으로 가져오는 것을 선택할 수 있습니다.

- 대량 가져오기에는 많은 수의 이력 레코드를 한 번에 가져오는 작업이 포함됩니다. SageMaker Data Wrangler 및 여러 데이터 소스를 사용하여 항목 상호 작용, 사용자, 항목 대량 데이터를 준비하고 가져올 수 있습니다. 또는 직접 대량 데이터를 준비하여 Amazon S3의 CSV 파일에서 Amazon Personalize로 직접 가져올 수 있습니다. Amazon Personalize의 대량 데이터 서식 지정 요구 사항에 대한 자세한 내용은 [데이터 형식 지침](#)을 참조하세요.
- 개별 가져오기의 경우 Amazon Personalize 콘솔 및 API 작업을 통해 개별 레코드를 가져옵니다. 또는 라이브 이벤트에서 실시간으로 상호작용 데이터를 가져올 수 있습니다.

Amazon Personalize 데이터 세트로 데이터를 가져온 후에는, 데이터를 분석하고, Amazon S3 버킷으로 내보내고, 업데이트하거나, 데이터 세트를 삭제하여 삭제할 수 있습니다. 자세한 내용은 [데이터셋의 훈련 데이터 관리](#) 섹션을 참조하세요.

카탈로그가 커지면 추가 대량 또는 개별 데이터 가져오기 작업을 통해 이력 데이터를 업데이트합니다. 실시간 추천을 받으려면 항목 상호 작용 데이터 세트를 사용자 행동에 따른 최신 상태로 유지하세요. 이렇게 하려면 이벤트 트래커 및 [PutEvents](#) 작업을 사용하여 실시간 상호작용 [이벤트](#)를 기록합니다. 자세한 내용은 [이벤트 기록](#) 단원을 참조하세요.

주제

- [대량 데이터 준비 및 가져오기](#)
- [개별 레코드 가져오기](#)

대량 데이터 준비 및 가져오기

데이터 세트를 생성했다면 Amazon Personalize로 대량 이력 데이터를 가져올 준비가 된 것입니다. 대량 레코드를 가져오려면 다음 두 가지 방법이 있습니다.

- 항목 상호 작용, 사용자 및 항목 데이터 세트의 경우 Amazon SageMaker Data Wrangler를 사용하여 40개 이상의 소스에서 데이터를 가져오고, 시각화와 Amazon Personalize의 특정 통찰력을 생성하고, 이를 Amazon Personalize 요구 사항에 맞게 변환할 수 있습니다.
- 모든 데이터 세트 유형에 대해 대량 데이터를 데이터 세트로 직접 가져올 수 있습니다. 직접 가져올 때는 Personalize 요구 사항을 충족하도록 데이터 형식을 수동으로 지정하고 S3에 업로드합니다. 그런 다음 스키마와 데이터세트를 생성하고 데이터세트 가져오기 작업을 통해 데이터를 데이터세트로 직접 가져옵니다.

다음 지침은 대량 데이터의 형식이 올바른지 확인하는 데 도움이 될 수 있습니다.

- 입력 데이터는 CSV(쉼표로 구분된 값) 파일에 있어야만 합니다.
- CSV 파일의 첫 번째 행에는 열 헤더가 포함되어야만 합니다. 헤더를 따옴표(")로 묶지 않습니다.
- 데이터세트 유형의 필수 필드가 있는지 확인하고, 해당 이름이 Personalize 요구 사항과 일치하는지 확인합니다. 예를 들어, 항목 데이터에는 각 항목의 ID가 포함된 ITEM_IDENTIFICATION_NUMBER로 불리는 열이 있을 수 있습니다. 이 열을 ITEM_ID 필드로 사용하려면 열 이름을 ITEM_ID로 바꾸세요. Data Wrangler를 사용하여 데이터 형식을 지정하는 경우, Personalize Data Wrangler 변환용 맵 열을 사용하여 열의 이름이 올바르게 지정되었는지 확인할 수 있습니다.

필수 필드에 대한 자세한 내용은 [스키마](#) 단원을 참조하세요. 데이터를 준비하기 위하여 Data Wrangler를 사용하는 것에 대한 자세한 내용은 [Amazon SageMaker 데이터 랭글러를 사용하여 데이터 준비 및 가져오기](#) 단원을 참조하세요.

- CSV 파일의 열 헤더 이름은 스키마에 매핑되어야만 합니다.
- CSV 파일의 각 레코드는 하나의 줄에 있어야만 합니다.
- 각 열의 데이터 유형은 스키마에 매핑되어야만 합니다. Data Wrangler를 사용하여 데이터의 형식을 지정한 경우, Data Wrangler 변환 [값을 유형으로 파싱](#)을 사용하여 데이터 유형을 변환할 수 있습니다.
- TIMESTAMP 및 CREATION_TIMESTAMP 데이터는 UNIX epoch 시간 형식이어야 합니다. 자세한 정보는 [타임스탬프 데이터](#)을 참조하세요.
- 아이템 ID, 사용자 ID 및 작업 ID 데이터에 문자나 특수 " 문자를 포함하지 마십시오.
- 데이터에 ASCII로 인코딩되지 않은 문자가 포함된 경우, CSV 파일은 UTF-8 형식으로 인코딩되어야만 합니다.
- [비정형 텍스트 메타데이터](#)에 설명된 대로 모든 텍스트 데이터의 형식을 지정해야만 합니다.
- [명시적 노출의 형식 지정](#) 및 [범주형 데이터의 형식 지정](#)에서 설명된 대로 노출 수 데이터와 범주형 데이터에 형식을 지정해야만 합니다.

Personalize의 대량 데이터 형식 지정 요구 사항에 대한 자세한 내용은 [데이터 형식 지침](#)을 참조하세요.

Personalize 데이터세트로 데이터를 가져온 후에는, 데이터를 분석하고, S3 버킷으로 내보내고, 업데이트하거나, 데이터세트를 삭제하여 삭제할 수 있습니다. 자세한 내용은 [데이터셋의 훈련 데이터 관리](#) 단원을 참조하세요.

이미 추천자를 생성했거나 캠페인과 함께 사용자 지정 솔루션 버전을 배포한 경우, 새 대량 레코드가 추천에 미치는 영향은 사용하는 도메인 사용 사례 또는 레시피에 따라 달라집니다. 자세한 내용은 [새 데이터가 실시간 추천에 영향을 미치는 방식](#) 단원을 참조하세요.

대량 레코드에 대한 필터 업데이트

대량 가져오기를 완료한 후 20분 이내에 Personalize는 데이터세트 그룹에서 생성한 모든 필터를 새 항목과 사용자 데이터로 업데이트합니다. 이 업데이트를 통해 Personalize는 사용자에게 대한 추천을 필터링할 때 최신 데이터를 사용할 수 있습니다.

주제

- [Amazon SageMaker 데이터 랭글러를 사용하여 데이터 준비 및 가져오기](#)
- [Personalize 데이터세트로 데이터 바로 가져오기](#)

Amazon SageMaker 데이터 랭글러를 사용하여 데이터 준비 및 가져오기

Important

데이터 랭글러를 사용하면 비용이 발생합니다. SageMaker 요금 및 가격의 전체 목록은 [Amazon SageMaker](#) 요금의 데이터 랭글러 탭을 참조하십시오. 추가 요금이 발생하지 않도록 하려면 작업을 마쳤으면 데이터 Data Wrangler 인스턴스를 종료하세요. 자세한 내용은 [Data Wrangler 종료](#)를 참조하세요.

데이터세트 그룹을 생성한 후, 아마존 데이터 랭글러 (SageMaker Data Wrangler) 를 사용하여 40개 이상의 소스에서 Amazon Personalize 데이터세트로 데이터를 가져올 수 있습니다. 데이터 랭글러는 Amazon SageMaker Studio Classic의 기능으로, 데이터를 가져오고, 준비하고, 변환하고, 분석할 수 있는 end-to-end 솔루션을 제공합니다. Data Wrangler를 사용하여 데이터를 준비하고 작업 데이터 세트 또는 작업 상호 작용 데이터 세트로 가져올 수는 없습니다.

Data Wrangler를 사용하여 데이터를 준비하고 가져올 때는 데이터 흐름을 사용합니다. 데이터 흐름은 데이터 가져오기부터 시작하여 일련의 기계 학습 데이터 준비 단계를 정의합니다. 흐름에 단계를 추가

할 때마다 Data Wrangler는 데이터를 변환하거나 시각화를 생성하는 등 데이터에 대한 조치를 취합니다.

다음은 Personalize의 데이터를 준비하기 위해 흐름에 추가할 수 있는 몇 가지 단계입니다.

- **인사이트:** Personalize의 특정 인사이트 단계를 흐름에 추가할 수 있습니다. 이러한 인사이트는 데이터에 대해 학습하고 개선하기 위해 어떤 조치를 취할 수 있는지 학습하는 데 도움이 될 수 있습니다.
- **시각화:** 시각화 단계를 추가하여 히스토그램 및 스캐터 차트와 같은 그래프를 생성할 수 있습니다. 그래프를 사용하면 데이터에서 특이값이나 누락된 값과 같은 문제를 검색하는 데 도움을 받을 수 있습니다.
- **변환:** Personalize의 구체적이고 일반적인 변환 단계를 사용하여 데이터가 Personalize 요구 사항을 충족하는지 확인할 수 있습니다. Personalize 변환을 사용하면 Personalize 데이터세트 유형에 따라 데이터 열을 필수 열에 매핑하는 데 도움을 받을 수 있습니다.

Personalize로 데이터를 가져오기 전에 Data Wrangler를 종료해야 하는 경우, [Personalize 콘솔에서 Data Wrangler를 실행](#)할 때 동일한 데이터세트 유형을 선택하여 중단한 부분으로 되돌아갈 수 있습니다. 또는 스튜디오 클래식을 통해 직접 데이터 랭글러에 액세스할 수도 있습니다. SageMaker

다음과 같이 Data Wrangler에서 Personalize로 데이터를 가져오는 것이 좋습니다. 변환, 시각화 및 분석 단계는 선택 사항이며 반복 가능하며 어떤 순서로든 완료할 수 있습니다.

1. [권한 설정](#) - Amazon Personalize 및 SageMaker 서비스 역할에 대한 권한을 설정합니다. 그리고 사용자에 대한 권한을 설정하세요.
2. Amazon [Personalize 콘솔에서 SageMaker 스튜디오 클래식에서 데이터 랭글러 실행](#) - Amazon [Personalize](#) 콘솔을 사용하여 스튜디오 클래식에서 도메인을 구성하고 SageMaker 데이터 랭글러를 시작합니다. SageMaker
3. [Data Wrangler로 데이터 가져오기](#) - 40개 이상의 소스에서 Data Wrangler로 데이터를 가져옵니다. 출처에는 Amazon Redshift, Amazon EMR 또는 Amazon Athena와 같은 AWS 서비스와 Snowflake 또는 같은 타사가 포함됩니다. DataBricks
4. [데이터 변환](#) - Data Wrangler를 사용하여 Personalize 요구 사항을 충족하도록 데이터를 변환합니다.
5. [데이터 시각화 및 분석](#) - Data Wrangler를 사용하여 데이터를 시각화하고 Personalize의 특정 인사이트를 통해 데이터를 분석합니다.
6. [Amazon Personalize로 데이터 처리 및 가져오기](#) - SageMaker 스튜디오 클래식 주피터 노트북을 사용하여 처리된 데이터를 Amazon Personalize로 가져올 수 있습니다.

추가 정보

다음 리소스는 Amazon SageMaker 데이터 랭글러 및 Amazon Personalize 사용에 대한 추가 정보를 제공합니다.

- 샘플 데이터세트를 처리하고 변환하는 과정을 안내하는 자습서는 Amazon 개발자 안내서의 [데모: 데이터 랭글러 타이타닉 데이터세트 안내서](#)를 참조하십시오. SageMaker 이 자습서에서는 Data Wrangler의 필드와 함수를 소개합니다.
- Amazon SageMaker 도메인 온보딩에 대한 자세한 내용은 Amazon SageMaker 개발자 안내서의 [Amazon SageMaker 도메인에 대한 빠른 온보딩](#)을 참조하십시오.
- Personalize 데이터세트 및 스키마 요구사항에 대한 자세한 내용은 [데이터 형식 지침](#) 및 [스키마](#) 단원을 참조하세요.

권한 설정

Data Wrangler로 데이터를 준비하려면 다음 권한을 설정해야만 합니다.

- Personalize용 서비스 역할 생성: 아직 생성하지 않았다면, [Personalize 설정](#)에서 지침을 완료하여 Personalize용 IAM 서비스 역할을 생성하세요. 이 역할에는 처리된 데이터를 저장하는 S3 버킷에 대한 GetObject 및 ListBucket 권한이 있어야만 합니다. 또한 모든 AWS KMS 키를 사용할 수 있는 권한이 있어야 합니다.

S3 버킷에 Personalize에 대한 액세스 권한을 부여하는 것과 관련된 자세한 내용은 [Personalize에 S3 리소스에 대한 액세스 권한 부여](#) 단원을 참조하세요. Amazon Personalize에 키에 대한 액세스 권한을 부여하는 AWS KMS 방법에 대한 자세한 내용은 [Amazon Personalize에 AWS KMS 키 사용 권한 부여](#)를 참조하십시오.

- 권한이 있는 관리 사용자 생성: 관리자는 도메인에 대한 전체 액세스 SageMaker 권한을 SageMaker 갖고 도메인을 생성할 수 있어야 합니다. SageMaker 자세한 내용은 Amazon SageMaker 개발자 안내서의 [관리 사용자 및 그룹 생성](#)을 참조하십시오.
- SageMaker 실행 역할 생성: SageMaker 리소스 및 Amazon Personalize 데이터 가져오기 작업에 액세스할 수 있는 SageMaker 실행 역할을 생성합니다. SageMaker 실행 역할에는 [AmazonSageMakerFullAccess](#) 정책이 첨부되어 있어야 합니다. 더 세분화된 데이터 랭글러 권한이 필요한 경우 Amazon 개발자 안내서의 [데이터 랭글러 보안 및 권한](#)을 참조하십시오. SageMaker [역할에 대한 자세한 내용은 역할](#)을 참조하십시오. [SageMaker SageMaker](#)

Amazon Personalize 데이터 가져오기 작업에 대한 액세스 권한을 부여하려면 다음 IAM 정책을 실행 역할에 연결하십시오. SageMaker 이 정책은 Personalize로 데이터를 가져오고 S3 버킷에 정책을 연결하는 데 필요한 권한을 부여합니다. 또한 서비스가 Personalize인 경우 PassRole 권한도 부여합

니다. Data Wrangler로 데이터를 준비한 후, S3 bucket-name을 서식이 지정된 데이터의 대상으로 사용하려는 S3 버킷의 이름으로 업데이트합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "personalize:Create*",
        "personalize:List*",
        "personalize:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutBucketPolicy"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "personalize.amazonaws.com"
        }
      }
    }
  ]
}
```

IAM 정책 생성에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 정책 생성](#)을 참조하세요. 역할에 정책을 연결하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 IAM 자격 증명 권한 추가 및 제거 단원을

참조하세요. https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_manage-attach-detach.html

아마존 Personalize에서 Data Wrangler 실행하기

Amazon Personalize에서 데이터 랭글러를 시작하려면 Amazon Personalize 콘솔을 사용하여 도메인을 구성하고 SageMaker 데이터 랭글러를 시작합니다.

Personalize에서 Data Wrangler 실행하기

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터세트 그룹 페이지에서 데이터세트 그룹을 선택합니다.
3. 데이터 세트 설정에서 데이터 세트 생성을 선택하고 생성할 데이터 세트 유형을 선택합니다. Data Wrangler를 사용하여 작업 데이터 세트 또는 작업 상호 작용 데이터 세트를 준비할 수 없습니다.
4. Data Wrangler를 사용하여 데이터 가져오기를 선택하고 다음을 선택합니다.
5. SageMaker 도메인의 경우 기존 도메인을 사용하거나 새 도메인을 생성하도록 선택합니다. SageMaker Studio Classic에서 데이터 랭글러에 액세스하려면 SageMaker 도메인이 필요합니다. 도메인 및 사용자 프로필에 대한 자세한 내용은 Amazon SageMaker 개발자 안내서의 [SageMaker 도메인을 참조하십시오](#).
6. 기존 도메인을 사용하려면 SageMaker 도메인과 사용자 프로필을 선택하여 도메인을 구성하십시오.
7. 새 도메인을 생성하려면:
 - 새 도메인에 이름을 지정합니다.
 - 사용자 프로필 이름을 선택합니다.
 - 실행 역할에 대해 [권한 설정](#)에서 생성한 역할을 선택합니다. 또는 CreateRole 권한이 있는 경우 역할 생성 마법사를 사용하여 새 역할을 생성하십시오. 사용하는 역할에 AmazonSageMakerFullAccess 정책을 연결해야만 합니다.
8. 다음을 선택합니다. 새 도메인을 생성하는 경우 도메인 생성이 SageMaker 시작됩니다. 이는 최대 10분이 걸릴 수 있습니다.
9. SageMaker 도메인의 세부 정보를 검토하세요.
10. 데이터 랭글러로 데이터 가져오기를 선택합니다. SageMaker Studio Classic에서 환경 만들기가 시작되고 완료되면 SageMaker Studio Classic의 데이터 랭글러 데이터 흐름 페이지가 새 탭에 열립니다. SageMaker Studio Classic에서 환경 만들기를 완료하는 데 최대 5분이 걸릴 수 있습니다.

니다. 작업이 끝나면 Data Wrangler로 데이터를 가져올 준비가 된 것입니다. 자세한 내용은 [Data Wrangler로 데이터 가져오기](#) 단원을 참조하세요.

Data Wrangler로 데이터 가져오기

SageMaker 도메인을 구성하고 새 탭에서 데이터 랭글러를 시작하면 소스에서 데이터 랭글러로 데이터를 가져올 준비가 된 것입니다. Data Wrangler를 사용하여 Personalize용 데이터를 준비할 때는 한 번에 하나의 데이터셋을 가져옵니다. 항목 상호 작용 데이터 세트로 시작하는 것이 좋습니다. Data Wrangler를 사용하여 작업 데이터 세트 또는 작업 상호 작용 데이터 세트를 준비할 수 없습니다.

데이터 흐름 페이지에서 시작합니다. 페이지는 다음과 같아야 합니다.

The screenshot displays the Amazon Personalize Data Flow interface. At the top right, it shows '16 vCPU + 64 GiB' and a 'Get help' button. A notification bar states: 'You can use Data Wrangler to import data from your data source into Amazon Personalize datasets. You start by specifying your data source and importing your data into Data Wrangler. Then you can analyze it, transform it and import it into Amazon Personalize. For information about importing data, see [Importing data using Data Wrangler](#).' Below this, the 'Data flow' section is active, with a 'Create job' button. The main content area features a flow diagram with three stages: 'Import Data' (represented by a data source icon), 'Prepare' (represented by a monitor with a magnifying glass), and 'Process' (represented by a server rack icon). Below the 'Import Data' stage is a blue button labeled 'Import data'. Below the 'Prepare' and 'Process' stages is a button labeled 'Use sample dataset'.

데이터 가져오기를 시작하려면 데이터 가져오기를 선택하고 데이터 소스를 지정합니다. Data Wrangler는 40개 이상의 소스를 지원합니다. 여기에는 Amazon Redshift, Amazon EMR 또는 Amazon

Athena와 같은 AWS 서비스와 Snowflake 또는 같은 제3자가 포함됩니다. DataBricks 데이터 소스마다 데이터 연결 및 가져오기 절차가 다릅니다.

사용 가능한 소스의 전체 목록과 데이터 가져오기 step-by-step 지침은 Amazon SageMaker 개발자 안내서의 [Import](#)를 참조하십시오.

Data Wrangler로 데이터를 가져오고 나면 데이터를 변환할 준비가 된 것입니다. 데이터 변환에 대한 자세한 내용은 [데이터 변환](#) 단원을 참조하세요.

데이터 변환

Data Wrangler에서 데이터를 변환하려면 데이터 흐름에 변환 단계를 추가합니다. Data Wrangler에는 Personalize용 맵 열 변환을 포함하여 데이터를 준비하는 데 사용할 수 있는 300개 이상의 변환이 포함되어 있습니다. 또한 일반 Data Wrangler 변환을 사용하여 특이값, 유형 문제, 누락된 값과 같은 문제를 해결할 수 있습니다.

데이터 변환을 완료한 후에는 Data Wrangler로 데이터를 분석할 수 있습니다. 또는 Data Wrangler에서 데이터 준비를 완료한 경우 데이터를 처리하고 Personalize로 가져올 수도 있습니다. 데이터 분석에 대한 자세한 내용은 [시각화 및 데이터 인사이트 생성](#) 단원을 참조하세요. 데이터 처리 및 가져오기에 대한 자세한 내용은 [데이터를 처리하고 Personalize로 가져오기](#) 단원을 참조하세요.

주제

- [Personalize용 매핑 열](#)
- [일반 Data Wrangler 변환](#)

Personalize용 매핑 열

Personalize 요구 사항을 충족하도록 데이터를 변환하려면 Personalize용 맵 열 변환을 추가하고, 열을 Personalize의 필수 및 선택 필드에 매핑합니다.

Personalize용 맵 열 변환을 사용하려면

1. 최신 변환의 경우 +를 선택하고 변환 추가를 선택합니다. 변환을 추가하지 않은 경우, 데이터 유형 변환에 대해 +를 선택합니다. Data Wrangler는 이 변환을 흐름에 자동으로 추가합니다.
2. 단계 추가를 선택합니다.
3. Personalize용 변환을 선택합니다. Personalize용 맵 열 변환은 기본 설정으로 선택됩니다.
4. 변환 필드를 사용하여 데이터를 필수 Personalize 속성에 매핑합니다.
 1. 데이터와 일치하는 데이터세트 유형(상호작용, 항목 또는 사용자)을 선택합니다.

2. 도메인(ECOMMERCE, VIDEO_ON_DEMAND 또는 사용자 지정)을 선택합니다. 선택한 도메인은 데이터세트 그룹을 생성할 때 지정한 도메인과 일치해야 합니다.
3. Personalize의 필수 및 선택 필드와 일치하는 열을 선택합니다. 예를 들어, item_ID 열의 경우 각 항목의 고유한 식별 정보를 저장하는 데이터 열을 선택합니다.

각 열 필드는 데이터 유형에 의해 필터링됩니다. Personalize 데이터 유형 요구 사항을 충족하는 데이터의 열만 사용할 수 있습니다. 데이터가 필수 유형이 아닌 경우, [유형으로 파스 값](#) Data Wrangler 변환을 사용하여 데이터를 변환할 수 있습니다.

일반 Data Wrangler 변환

다음과 같은 일반 Data Wrangler 변환은 Personalize에 사용할 데이터를 준비하는 데 도움이 될 수 있습니다.

- 데이터 유형 전환: 필드가 Personalize용 맵 열 변환에서 가능한 옵션으로 나열되지 않은 경우, 해당 데이터 유형의 전환이 필요할 수 있습니다. Data Wrangler 변환 [유형으로 파스 값](#) 데이터를 변환하는 데 도움이 될 수 있습니다. 또는 흐름을 생성할 때 Data Wrangler가 기본값으로 추가하는 데이터 유형 변환을 사용할 수 있습니다. 이 변환을 사용하려면 유형 드롭다운 목록에서 데이터 유형을 선택하고 미리보기 선택한 다음 업데이트를 선택합니다.

필드에 필요한 데이터 유형에 대한 자세한 내용은 [스키마](#)의 도메인 및 데이터세트 유형에 대한 단원을 참조하세요.

- 결측값 및 특이값 처리: 결측값 또는 특이값 인사이트를 생성하는 경우, Data Wrangler 변환 [특이값 처리](#) 및 [결측값 처리](#)를 사용하여 이러한 문제를 해결할 수 있습니다.
- 사용자 지정 변환: 데이터 랭글러를 사용하면 Python (사용자 정의 함수) PySpark, Pandas 또는 (SQL) 을 사용하여 자체 변환을 만들 수 있습니다. PySpark 사용자 지정 변환을 사용하여 중복된 열 삭제 또는 열별로 그룹화 등의 작업을 수행할 수 있습니다. 자세한 내용은 Amazon SageMaker 개발자 안내서의 [사용자 지정 변환](#)을 참조하십시오.

시각화 및 데이터 인사이트 생성

데이터를 Data Wrangler로 가져온 후 이를 사용하여 시각화 및 데이터 인사이트를 생성할 수 있습니다.

- [시각화](#): Data Wrangler는 히스토그램, 스캐터 차트 등 다양한 유형의 그래프를 생성할 수 있습니다. 예를 들어, 히스토그램을 생성하여 데이터의 특이값을 식별할 수 있습니다.

- [데이터 인사이트](#): Personalize의 데이터 품질 및 인사이트 보고서를 사용하여 데이터 인사이트와 열 및 행 통계를 통해 데이터에 대해 학습할 수 있습니다. 이 보고서를 통해 데이터에 유형 문제가 있는지 알 수 있습니다. 또한 데이터를 개선하기 위해 어떤 조치를 취할 수 있는지 학습할 수 있습니다. 이러한 조치는 모델 교육 요구 사항과 같은 Personalize 리소스 요구 사항을 충족하는 데 도움을 줄 수 있거나 추천을 개선할 수 있습니다.

시각화와 인사이트를 통해 데이터에 대해 학습하게 되면 이 정보를 사용하여 추가 변환을 적용하여 데이터를 개선할 수 있습니다. 또는 데이터 준비를 완료한 경우 데이터를 처리하고 Personalize로 가져올 수도 있습니다. 데이터 변환에 대한 자세한 내용은 [데이터 변환](#) 단원을 참조하세요. 데이터 처리 및 가져오기에 대한 자세한 내용은 [데이터를 처리하고 Personalize로 가져오기](#) 단원을 참조하세요.

시각화 생성

Data Wrangler를 사용하여 히스토그램, 스캐터 차트 등 다양한 유형의 그래프를 생성할 수 있습니다. 예를 들어, 히스토그램을 생성하여 데이터의 특이값을 식별할 수 있습니다. 데이터 시각화를 생성하려면 흐름에 분석 단계를 추가하고 분석 유형에서 생성하려는 시각화를 선택합니다.

Data Wrangler에서 시각화를 생성하는 방법에 대한 자세한 내용은 Amazon 개발자 안내서의 [분석 및 시각화](#)를 참조하십시오. SageMaker

데이터 인사이트 생성

Data Wrangler를 사용하여 데이터세트 유형에 맞는 Personalize용 데이터 품질 및 인사이트 보고서를 생성할 수 있습니다. 보고서를 생성하기 전에 Personalize 요구 사항을 충족하도록 데이터를 변환하는 것이 좋습니다. 이렇게 하면 보다 관련성이 높은 인사이트를 얻을 수 있습니다. 자세한 내용은 [데이터 변환](#) 단원을 참조하세요.

주제

- [보고서 콘텐츠](#)
- [보고서 생성](#)

보고서 콘텐츠

Personalize용 데이터 품질 및 인사이트 보고서에는 다음 단원이 포함됩니다.

- 요약: 보고서 요약에는 데이터세트 통계와 우선순위가 높은 경고가 포함됩니다.
 - 데이터세트 통계: 여기에는 Personalize의 특정 통계(예: 상호작용 데이터의 고유 사용자의 수) 및 일반 통계(예: 누락된 값 또는 특이값의 수)가 포함됩니다.

- 우선 순위가 높은 경고: 교육 또는 추천에 가장 큰 영향을 미치는 Personalize의 특정 인사이트입니다. 각 경고에는 문제를 해결하기 위해 취할 수 있는 권장 조치가 포함되어 있습니다.
- 중복된 행 및 불완전한 행: 이들 단원에는 누락된 값이 있는 행과 데이터에서 중복된 행에 대한 정보가 포함되어 있습니다.
- 특성 요약: 이 단원에는 각 열의 데이터 유형, 유효하지 않거나 누락된 데이터 정보, 경고 개수가 포함되어 있습니다.
- 특성 세부 정보: 이 단원에는 각 데이터 열에 대한 세부 정보가 포함된 하위 단원이 포함되어 있습니다. 각 하위 단원에는 범주형 값의 수, 누락값 정보와 같은 열에 대한 통계가 포함되어 있습니다. 그리고 각 하위 단원에는 데이터 열에 대한 Personalize의 특정 인사이트와 권장 조치가 포함되어 있습니다. 예를 들어, 한 열에 30개 이상의 범주가 있다는 인사이트가 표시될 수 있습니다.

데이터 유형 문제

보고서는 데이터 유형이 올바르지 않은 열을 식별하고 필수 유형을 지정합니다. 이러한 특성과 관련된 인사이트를 얻으려면 열의 데이터 유형을 변환하고 보고서를 다시 생성해야 합니다. 유형을 전환하려면 Data Wrangler 변환 [유형으로 파스 값](#)을 사용할 수 있습니다.

Personalize 인사이트

Personalize 인사이트에는 조사 결과와 제안된 조치가 포함됩니다. 이 조치는 선택 사항입니다. 예를 들어, 보고서에는 범주형 데이터 열에 대한 범주의 수와 관련된 인사이트와 조치가 포함될 수 있습니다. 열이 범주형이라고 생각하지 않으면 이 통계를 무시하고 어떤 조치도 취하지 않아도 됩니다.

사소한 표현의 차이를 제외하고, Personalize의 특정 인사이트는 Personalize로 데이터를 분석할 때 생성할 수 있는 단일 데이터세트 인사이트와 동일합니다. 예를 들어, Data Wrangler의 인사이트 보고서에는 “항목 상호 작용 데이터 세트에 상호 작용이 2번 이상 있는 순 사용자가 X명뿐입니다.”와 같은 인사이트가 포함되어 있습니다. 하지만 “항목 데이터 세트에 있는 항목 중 X%에 항목 상호 작용 데이터 세트에서 상호 작용이 없음”과 같은 인사이트는 포함되지 않습니다.

가능한 Personalize의 특정 인사이트 목록은 [데이터 인사이트](#)에서 여러 데이터세트를 참조하지 않는 인사이트를 참조하세요.

보고서 예

Personalize 보고서의 모양과 느낌은 Data Wrangler의 일반 인사이트 보고서와 동일합니다. 일반 인사이트 보고서의 예는 Amazon SageMaker 개발자 안내서의 [데이터 및 데이터 품질에 대한 인사이트 얻기](#)를 참조하십시오. 다음 예제는 항목 상호 작용 데이터 세트에 대한 보고서의 요약 섹션을 보여줍니다.

다. 여기에는 데이터 세트 통계와 우선순위가 높을 수 있는 몇 가지 항목 상호 작용 데이터 세트 경고가 포함됩니다.

SUMMARY

Dataset statistics

Key	Value	Feature type	Count
Number of features	6	numeric	2
Number of rows	31	categorical	0
Missing	0%	text	4
Valid	100%	datetime	0
Duplicate rows	6.45%	binary	0
Users with sufficient int...	0	unknown	0
Number of unique users	30		
Number of unique items	30		
Sparse rows	0%		
Distinct rows	30		

High Priority Warnings

4 high severity warnings were detected. See the list below.

Duplicate rows High

We found that 6.45% of the data are duplicate. Some data sources could include valid duplicates and in other cases these duplicates could point to problems in data collection. Duplicate samples resulting from faulty data collection, could derail machine learning processes that rely on splitting to independent training and validation folds. For example quick model scores, prediction power estimation and automatic hyper parameter tuning. Duplicate samples could be removed from the dataset using the **Drop duplicates** transform under **Manage rows**.

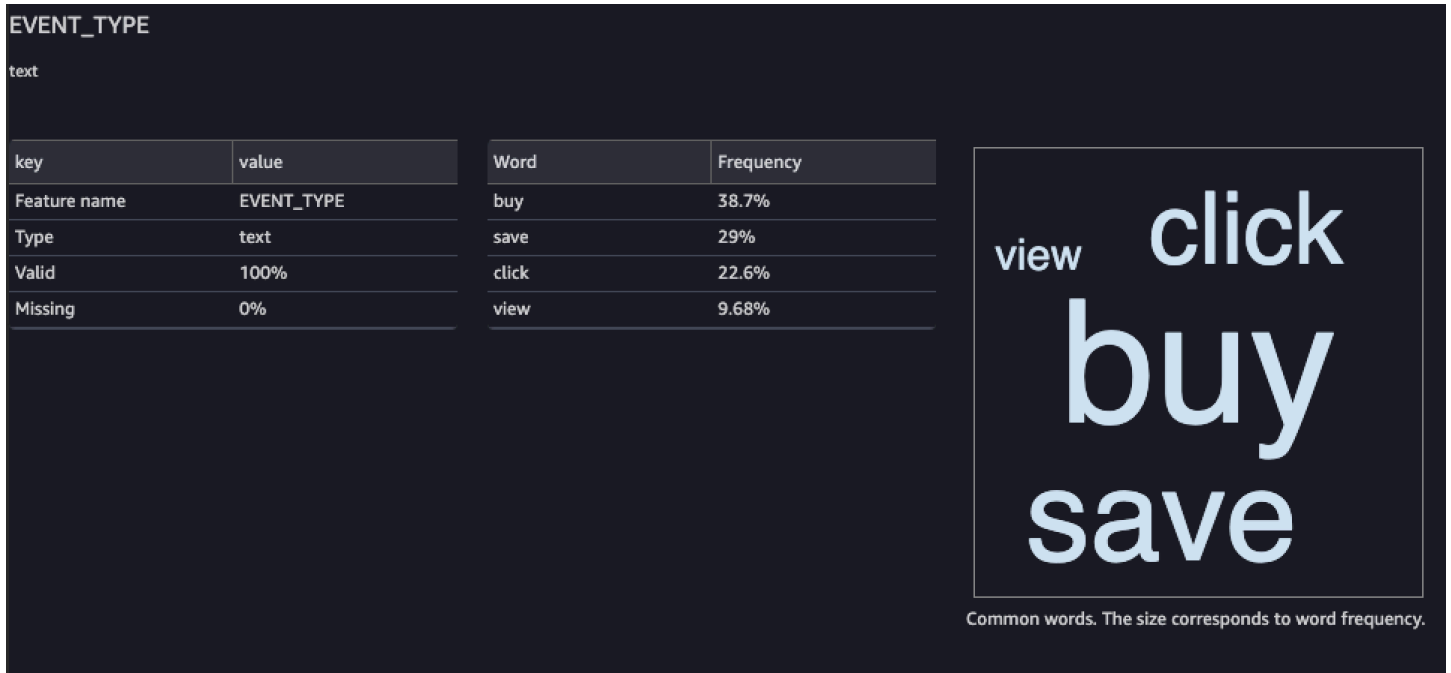
Insufficient interactions High

The Interactions dataset has only 30 interactions. Model training requires a minimum of 1,000 interactions. We recommend at least 50,000. Import 49970 additional unique interactions records before training a model.

Insufficient Users High

The Interactions dataset has only 0 unique users with two or more interactions. Model training requires at least 25 such users. We recommend at least 1,000. Import at least 2 interactions records each for 1000 additional users.

다음 예제는 항목 상호 작용 데이터 세트의 EVENT_TYPE 열에 대한 특성 세부 정보 섹션이 보고서에 어떻게 표시되는지 보여줍니다.



보고서 생성

Personalize용 데이터 품질 및 인사이트 보고서를 생성하려면 변환에 대한 데이터 인사이트 얻기를 선택하고 분석을 생성합니다.

Personalize용 데이터 품질 및 인사이트 보고서를 생성하려면

1. 분석 중인 변환에 대해 + 옵션을 선택합니다. 변환을 추가하지 않은 경우, 데이터 유형 변환에 대해 +를 선택합니다. Data Wrangler는 이 변환을 흐름에 자동으로 추가합니다.
2. 데이터 인사이트 얻기를 선택합니다. 분석 생성 패널이 표시됩니다.
3. 분석 유형의 경우 Personalize용 데이터 품질 및 인사이트 보고서를 선택합니다.
4. 데이터세트 유형의 경우 분석 중인 Personalize 데이터세트의 유형을 선택합니다.
5. 원하는 대로 전체 데이터에 대해 실행을 선택합니다. 기본 설정으로 Data Wrangler는 데이터 샘플에 대한 인사이트만 생성합니다.
6. 생성을 선택합니다. 분석이 완료되면 보고서가 표시됩니다.

데이터를 처리하고 Personalize로 가져오기

데이터 분석 및 변환을 마치면 데이터를 처리하고 Personalize로 가져올 준비가 된 것입니다.

- [데이터 처리](#) - 데이터를 처리하면 변환이 전체 데이터세트에 적용되고 지정한 대상으로 출력됩니다. 이 경우 S3 버킷을 지정합니다.

- [Amazon Personalize로 데이터 가져오기 — 처리된 데이터를 Amazon Personalize로 가져오려면](#) 스튜디오 클래식에서 제공하는 Jupyter 노트북을 실행합니다. SageMaker 이 노트북은 Personalize 데이터세트를 생성하고 데이터를 해당 데이터세트로 가져옵니다.

데이터 처리

Personalize로 데이터를 가져오기 전에 변환을 전체 데이터세트에 적용하고 S3 버킷으로 출력해야 합니다. 이렇게 하려면 대상이 S3 버킷으로 설정된 대상 노드를 생성한 다음 변환을 위한 처리 작업을 시작합니다.

대상 지정 및 프로세스 작업 시작에 대한 step-by-step 지침은 [Amazon SageMaker Data Wrangler를 사용하여 몇 번의 클릭으로 처리 작업 시작](#)을 참조하십시오. 대상을 추가할 때 S3을 선택합니다. 처리된 데이터를 Personalize로 가져올 때 이 위치를 사용하게 됩니다.

데이터 처리를 마치면 S3 버킷에서 Personalize로 데이터를 가져올 준비가 된 것입니다.

Personalize 데이터세트로 데이터 가져오기

데이터를 처리하고 나면 Personalize로 데이터를 가져올 준비가 된 것입니다. 처리된 데이터를 Amazon Personalize로 가져오려면 스튜디오 클래식에서 제공하는 Jupyter 노트북을 실행합니다. SageMaker 이 노트북은 Personalize 데이터세트를 생성하고 데이터를 해당 데이터세트로 가져옵니다.

처리된 데이터를 Personalize로 가져오려면

1. 내보내려는 변환의 경우, 내보내기를 선택하고 Personalize(Jupyter 노트북 사용)를 선택합니다.
2. 노트북을 수정하여 처리 작업의 데이터 대상으로 사용한 S3 버킷을 지정합니다. 원하는 대로 데이터세트 그룹의 도메인을 지정합니다. 기본 설정으로 노트북은 사용자 지정 데이터세트 그룹을 생성합니다.
3. 스키마를 생성하는 노트북 셀을 검토합니다. 셀을 실행하기 전에 스키마 필드에 예상 유형과 속성이 있는지 확인합니다.
 - null 데이터를 지원하는 필드가 유형 목록에 null나열된지 확인합니다. 다음 예제에서는 필드에 null을 추가하는 방법을 보여 줍니다.

```
{
  "name": "GENDER",
  "type": [
    "null",
```

```

    "string"
  ],
  "categorical": true
}

```

- 범주형 필드의 범주형 속성이 true로 설정되어 있는지 확인합니다. 다음 예제에서는 필드에 범주를 표시하는 방법을 보여 줍니다.

```

{
  "name": "SUBSCRIPTION_MODEL",
  "type": "string",
  "categorical": true
}

```

- 텍스트 필드의 텍스트 속성이 true로 설정되어 있는지 확인합니다. 다음 예제에서는 필드에 텍스트를 표시하는 방법을 보여 줍니다.

```

{
  "name": "DESCRIPTION",
  "type": [
    "null",
    "string"
  ],
  "textual": true
}

```

4. 노트북을 실행하여 스키마를 생성하고, 데이터셋을 생성하고, 데이터를 Personalize 데이터 세트로 가져옵니다. 스튜디오 클래식 외부에서 노트북을 실행하는 것처럼 노트북을 실행합니다. SageMaker Jupyter 노트북 실행에 대한 자세한 내용은 [코드 실행](#)을 참조하세요. SageMaker Studio Classic의 노트북에 대한 자세한 내용은 [Amazon SageMaker 개발자 SageMaker 안내서의 Amazon 노트북 사용을](#) 참조하십시오.

노트북을 완료한 후 상호작용 데이터를 가져오고 나면 추천 또는 사용자 지정 리소스를 생성할 준비가 된 것입니다. 또는 항목 데이터셋 또는 사용자 데이터셋을 사용하여 프로세스를 반복할 수 있습니다. 추천 또는 사용자 지정 리소스 생성 방법에 대한 자세한 내용은 [3단계: 추천 또는 사용자 지정 리소스 생성](#)을 참조하세요.

Personalize 데이터세트로 데이터 바로 가져오기

데이터 세트를 생성했다면 대용량 CSV 파일의 대량 레코드를 Amazon Personalize 데이터 세트로 가져올 준비가 된 것입니다.

Personalize 데이터세트로 데이터를 바로 가져오려면 다음과 같이 하세요.

1. 데이터를 기반으로 스키마 JSON 파일을 생성합니다. 스키마 요구 사항 및 예제는 [스키마](#) 단원을 참조하세요.
2. 데이터 형식이 올바른지 확인하세요. 열 이름은 스키마와 일치해야만 합니다. 데이터는 CSV 파일로 되어 있어야만 합니다. 데이터 형식 지침은 [데이터 형식 지침](#) 단원을 참조하세요.
3. Simple Storage Service(S3) 버킷에 CSV 파일을 업로드하고 Personalize에 S3 리소스에 액세스할 수 있는 권한을 부여합니다.
4. 1단계의 JSON 파일을 사용하여 Personalize 스키마를 생성합니다. 그리고 Personalize 데이터세트를 생성합니다.
5. S3 버킷의 데이터로 데이터세트를 채우는 가져오기 작업을 생성합니다. 상호작용 데이터세트에 대한 데이터세트 가져오기 작업을 생성하려면 CSV 파일에 1000개 이상의 상호작용 레코드가 있어야 합니다.

Personalize 데이터세트로 데이터를 가져온 후에는, 데이터를 분석하고, S3 버킷으로 내보내고, 업데이트하거나, 데이터세트를 삭제하여 데이터를 삭제할 수 있습니다. 자세한 내용은 [데이터셋의 훈련 데이터 관리](#) 단원을 참조하세요.

주제

- [S3 버킷에 업로드](#)
- [데이터세트 및 스키마 생성](#)
- [데이터세트 가져오기 작업으로 대량 레코드 가져오기](#)

S3 버킷에 업로드

과거 입력 데이터의 형식을 지정한 후에는([데이터 형식 지침](#) 참조) S3 버킷에 CSV 파일을 업로드하고 Personalize에 S3 리소스에 액세스할 수 있는 권한을 부여해야 합니다.

1. 아직 권한을 부여하지 않았다면 Personalize가 사용자를 대신하여 Personalize의 리소스에 액세스할 수 있도록 하는 [권한 설정](#)의 단계를 따라 합니다.
2. CSV 파일을 Simple Storage Service(S3) 버킷에 업로드합니다. Personalize는 이 위치에서 데이터를 가져옵니다. 자세한 내용은 Simple Storage Service 사용 설명서의 [드래그 앤 드롭을 사용하여 파일 및 폴더 업로드](#) 단원을 참조하세요.
3. S3 버킷과 Personalize 서비스 역할에 액세스 정책을 연결하여 S3 리소스에 대한 액세스 권한을 Personalize에 부여합니다. [Personalize에 S3 리소스에 대한 액세스 권한 부여](#) 단원을 참조하세요.

암호화에 AWS Key Management Service (AWS KMS) 를 사용하는 경우 Amazon Personalize 와 Amazon Personalize IAM 서비스 역할에 키를 사용할 권한을 부여해야 합니다. 자세한 정보는 [Amazon Personalize에 AWS KMS 키 사용 권한 부여](#)을 참조하세요.

Amazon S3 버킷에 데이터를 업로드하고 Amazon Personalize에 Amazon S3에 대한 액세스 권한을 부여하면 Amazon Personalize 스키마와 데이터셋을 생성할 준비가 된 것입니다. [데이터세트 및 스키마 생성](#) 섹션을 참조하세요.

데이터세트 및 스키마 생성

1단계: 데이터세트 그룹 생성 작업을 완료했으면 데이터셋을 생성할 준비가 된 것입니다. 데이터세트는 데이터를 위한 Personalize 컨테이너입니다. 데이터셋을 생성할 때 데이터세트에 대한 스키마도 생성합니다. 스키마는 Personalize에 데이터 구조를 알려주고 Personalize가 데이터를 파싱할 수 있도록 해 줍니다.

Amazon Personalize 콘솔 AWS Command Line Interface ,AWS CLI() 또는 SDK를 사용하여 데이터셋을 생성합니다. AWS 도메인 데이터세트 그룹에서는 액션 및 액션 인터랙션 데이터셋을 비롯한 차선책 액션 리소스를 생성할 수 없습니다. 다양한 유형의 데이터세트와 데이터세트 및 스키마 요구사항에 대한 자세한 설명은 [데이터세트 및 스키마](#) 단원을 참조하세요.

주제

- [데이터세트 및 스키마 생성\(콘솔\)](#)
- [데이터세트 및 스키마 생성\(AWS CLI\)](#)
- [데이터세트 및 스키마 \(SDK\) 생성AWS](#)

데이터세트 및 스키마 생성(콘솔)

데이터 세트 그룹의 첫 번째 데이터 세트인 경우, 첫 번째 데이터 세트 유형은 항목 상호 작용 데이터 세트일 것입니다. 콘솔에서 항목 상호 작용 데이터 세트를 생성하려면 데이터 세트 이름을 지정한 다음 [Avro 형식](#)의 JSON 스키마를 지정합니다. 이 데이터세트 그룹의 첫 번째 데이터세트가 아닌 경우, 데이터세트 유형을 선택한 다음 이름과 스키마를 지정합니다.

Personalize 데이터세트 및 스키마 요구사항에 대한 자세한 내용은 [데이터세트 및 스키마](#) 단원을 참조하세요.

Note

1단계: 데이터세트 그룹 생성을 막 완료하고 이미 데이터 세트를 만드는 중이라면 이 절차의 4 단계로 건너뛰세요.

데이터세트 및 스키마를 생성하려면

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터세트 그룹 페이지의, 1단계: 데이터세트 그룹 생성에서 데이터세트 그룹을 선택합니다.
3. 데이터 세트 설정에서 데이터 세트 생성을 선택하고 생성할 데이터 세트 유형을 선택합니다.
4. Personalize 데이터세트로 직접 데이터 가져오기를 선택하고 다음을 선택합니다.
5. 데이터세트 세부 정보에서 데이터세트 이름에 대해 데이터세트의 이름을 지정합니다.
6. 스키마 세부 정보에서 스키마 선택에 대해 기존 스키마를 선택하거나 새 스키마 생성을 선택합니다.
7. 새 스키마를 생성할 경우 스키마 정의를 위해 데이터와 일치하는 스키마 JSON을 붙여넣습니다. 스키마에 있는 예제를 가이드로 사용하세요. 스키마를 생성하고 나면 스키마를 변경할 수 없습니다.
8. 새 스키마 이름에서 새 스키마의 이름을 지정합니다.
9. 태그의 경우, 원하는 태그를 추가할 수 있습니다. Personalize 리소스 태그 지정에 대한 자세한 내용은 Personalize 리소스에 태그 지정 단원을 참조하세요.
10. 다음을 선택하고 대량 데이터 준비 및 가져오기의 지침에 따라 데이터를 가져옵니다.

데이터세트 및 스키마 생성(AWS CLI)

를 사용하여 데이터세트와 스키마를 생성하려면 먼저 Avro 형식으로 스키마를 정의하고 작업을 사용하여 Amazon Personalize에 추가합니다. AWS CLI CreateSchema 그런 다음 CreateDataset 작업을 사용하여 데이터세트를 생성합니다. Personalize 데이터세트 및 스키마 요구사항에 대한 자세한 내용은 데이터세트 및 스키마 단원을 참조하세요.

스키마 및 데이터세트를 생성하려면

1. 스키마 파일을 Avro 형식으로 생성하고 JSON 파일로 저장합니다. 이 파일은 생성하고 있는 상호 작용과 같은 데이터세트의 유형을 기반으로 해야 합니다.

스키마는 데이터의 열과 일치해야만 하며, 스키마 name은 Amazon Personalize에서 인식하는 데이터 세트 유형 중 하나와 일치해야만 합니다. 다음은 최소 항목 상호 작용 데이터 세트 스키마를 사용한 예제입니다. 더 많은 예제는 [스키마](#) 단원을 참조하세요.

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "TIMESTAMP",
      "type": "long"
    }
  ],
  "version": "1.0"
}
```

2. 다음 명령을 실행하여 Personalize에서 스키마를 생성합니다. 스키마를 생성하고 나면 스키마를 변경할 수 없습니다. `schemaName`을 스키마의 이름으로 대체하고 `file://SchemaName.json`을 이전 단계에서 생성한 JSON 파일의 위치와 바꿉니다. 이 예제에서는 이 파일을 현재 폴더에 속한 것으로 표시합니다.

도메인 데이터 세트 그룹의 데이터 세트에 대한 스키마를 생성하는 경우 `domain` 파라미터를 추가하고 `ECOMMERCE` 또는 `VIDEO_ON_DEMAND`로 설정합니다. API에 대한 자세한 내용은 [CreateSchema](#) 단원을 참조하세요.

```
aws personalize create-schema \
  --name SchemaName \
  --schema file://SchemaName.json
```

스키마의 리소스 이름(ARN)이 다음 예제와 같이 표시됩니다.

```
{
  "schemaArn": "arn:aws:personalize:us-west-2:acct-id:schema/SchemaName"
}
```

- 다음 명령을 실행하여 빈 데이터세트를 생성합니다. [데이터세트 그룹 생성\(AWS CLI\)](#)에서 가져온 데이터세트 그룹 리소스 이름(ARN)과 이전 단계에서 가져온 스키마 ARN을 제공합니다. 데이터세트 유형 값은 Interactions, Users, Items, Actions 또는 Action_Interactions일 수 있습니다. API에 대한 자세한 내용은 [CreateDataset](#) 단원을 참조하세요.

```
aws personalize create-dataset \
  --name Dataset Name \
  --dataset-group-arn Dataset Group ARN \
  --dataset-type Dataset Type \
  --schema-arn Schema Arn
```

데이터세트 ARN은 다음 예제와 같이 표시됩니다.

```
{
  "datasetArn": "arn:aws:personalize:us-west-2:acct-id:dataset/DatasetName/
INTERACTIONS"
}
```

- 추후 사용할 수 있도록 데이터세트 ARN을 기록합니다. 데이터 세트를 생성하고 나면 교육 데이터를 가져올 준비가 됩니다. [대량 데이터 준비 및 가져오기](#) 섹션을 참조하십시오.

데이터세트 및 스키마 (SDK) 생성AWS

AWS SDK를 사용하여 데이터세트와 스키마를 생성하려면 먼저 [Avro 형식으로](#) 스키마를 정의하고 작업을 사용하여 Amazon Personalize에 추가합니다. [CreateSchema](#) 스키마를 생성하고 나면 스키마를 변경할 수 없습니다. 그런 다음 [CreateDataset](#) 작업을 사용하여 데이터세트를 생성합니다. Personalize 데이터세트 및 스키마 요구사항에 대한 자세한 내용은 [데이터세트 및 스키마](#) 단원을 참조하세요.

스키마 및 데이터세트를 생성하려면

- 스키마 파일을 Avro 형식으로 생성하고 작업 디렉토리에 JSON 파일로 저장합니다.

스키마는 데이터의 열과 일치해야만 하며, 스키마 name은 Amazon Personalize에서 인식하는 데이터세트 유형 중 하나와 일치해야만 합니다. 다음은 최소 항목 상호 작용 데이터 세트 스키마를 사용한 예제입니다. 더 많은 예제는 [스키마](#) 단원을 참조하세요.

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "TIMESTAMP",
      "type": "long"
    }
  ],
  "version": "1.0"
}
```

2. 다음 코드를 사용하여 스키마를 생성합니다. 스키마의 이름과 스키마 JSON 파일의 파일 경로를 지정합니다.

도메인 데이터 세트 그룹의 데이터 세트에 대한 스키마를 생성하는 경우 domain 파라미터를 추가하고 ECOMMERCE 또는 VIDEO_ON_DEMAND로 설정합니다. API에 대한 자세한 내용은 [CreateSchema](#) 단원을 참조하세요.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

with open('schemaFile.json') as f:
    createSchemaResponse = personalize.create_schema(
        name = 'schema name',
        schema = f.read()
    )

schema_arn = createSchemaResponse['schemaArn']
```

```
print('Schema ARN:' + schema_arn )
```

SDK for Java 2.x

```
public static String createSchema(PersonalizeClient personalizeClient, String
    schemaName, String filePath) {

    String schema = null;

    try {
        schema = new String(Files.readAllBytes(Paths.get(filePath)));
    } catch (IOException e) {
        System.out.println(e.getMessage());
    }

    try {
        CreateSchemaRequest createSchemaRequest = CreateSchemaRequest.builder()
            .name(schemaName)
            .schema(schema)
            .build();

        String schemaArn =
personalizeClient.createSchema(createSchemaRequest).schemaArn();
        System.out.println("Schema arn: " + schemaArn);

        return schemaArn;

    } catch(PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { CreateSchemaCommand } from
    "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});
```

```

import fs from 'fs';

let schemaFilePath = "SCHEMA_PATH";
let mySchema = "";

try {
  mySchema = fs.readFileSync(schemaFilePath).toString();
} catch (err) {
  mySchema = 'TEST' // For unit tests.
}

// Set the schema parameters.
export const createSchemaParam = {
  name: 'NAME', /* required */
  schema: mySchema /* required */
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
    CreateSchemaCommand(createSchemaParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};

run();

```

Personalize는 새 스키마의 ARN을 반환합니다. 이 정보는 다음 단계에서 사용해야 하므로 기록해 둡니다.

3. [CreateDataset](#) 작업을 사용하여 데이터세트를 생성합니다. 다음 코드는 데이터세트를 생성하는 방법을 보여줍니다. 데이터 세트 그룹의 Amazon 리소스 이름(ARN)과 이전 단계의 스키마 ARN을 지정하고 데이터 세트 유형을 지정합니다. 데이터 세트 유형 값은 Interactions, Users, Items, Actions 또는 Action_Interactions일 수 있습니다. 데이터세트의 다양한 유형에 대한 자세한 내용은 [데이터세트 및 스키마](#) 단원을 참조하세요.

SDK for Python (Boto3)

```
import boto3
```

```

personalize = boto3.client('personalize')

response = personalize.create_dataset(
    name = 'dataset_name',
    schemaArn = 'schema_arn',
    datasetGroupArn = 'dataset_group_arn',
    datasetType = 'dataset_type'
)

print ('Dataset Arn: ' + response['datasetArn'])

```

SDK for Java 2.x

```

public static String createDataset(PersonalizeClient personalizeClient,
                                   String datasetName,
                                   String datasetGroupArn,
                                   String datasetType,
                                   String schemaArn) {
    try {
        CreateDatasetRequest request = CreateDatasetRequest.builder()
            .name(datasetName)
            .datasetGroupArn(datasetGroupArn)
            .datasetType(datasetType)
            .schemaArn(schemaArn).build();

        String datasetArn =
personalizeClient.createDataset(request).datasetArn();
        System.out.println("Dataset " + datasetName + " created. Dataset ARN: "
+ datasetArn);

        return datasetArn;

    } catch(PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}

```

SDK for JavaScript v3

```

// Get service clients module and commands using ES6 syntax.
import { CreateDatasetCommand } from

```



```

    "@aws-sdk/client-personalize";
import { personalizeClient } from "../libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the dataset's parameters.
export const createDatasetParam = {
  datasetGroupArn: 'DATASET_GROUP_ARN', /* required */
  datasetType: 'DATASET_TYPE', /* required */
  name: 'NAME', /* required */
  schemaArn: 'SCHEMA_ARN' /* required */
}

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
CreateDatasetCommand(createDatasetParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();

```

데이터 세트를 생성하고 나면 교육 데이터를 가져올 준비가 됩니다. [대량 데이터 준비 및 가져오기](#) 섹션을 참조하세요.

데이터세트 가져오기 작업으로 대량 레코드 가져오기

입력 데이터의 형식을 지정하고([데이터 형식 지침](#) 참조) Simple Storage Service(S3) 버킷에 업로드하고([S3 버킷에 업로드](#) 참조), [데이터세트 및 스키마 생성](#) 작업을 완료한 후에, 데이터세트 가져오기 작업을 생성하여 대량 레코드를 데이터세트로 가져옵니다.

데이터세트 가져오기 작업은 S3 버킷의 데이터로 데이터세트를 채우는 대량 가져오기 도구입니다. Amazon Personalize 콘솔 AWS Command Line Interface (AWS CLI) 또는 SDK를 사용하여 데이터세트 가져오기 작업을 생성할 수 있습니다. AWS

이전에 데이터세트에 대한 데이터세트 가져오기 작업을 생성한 경우, 새 데이터세트 가져오기 작업을 사용하여 기존 대량 데이터에 추가하거나 기존 대량 데이터를 대체할 수 있습니다. 자세한 내용은 [기존 대량 데이터 업데이트](#) 단원을 참조하세요.

주제

- [가져오기 모드](#)
- [대량 레코드 가져오기\(콘솔\)](#)
- [대량 레코드 가져오기\(AWS CLI\)](#)
- [대량 레코드 가져오기\(AWS SDK\)](#)

가져오기 모드

데이터세트에 대한 가져오기 작업을 이미 생성한 경우 Amazon Personalize에서 새 레코드를 추가하는 방법을 구성할 수 있습니다. 이렇게 하려면 데이터세트 가져오기 작업의 가져오기 모드를 지정해야 합니다. 대량 레코드를 가져오지 않은 경우 콘솔에서 가져오기 모드 필드를 사용할 수 없으며 CreateDatasetImportJob API 작업에서만 지정할 FULL 수 있습니다. 기본값은 전체 교체입니다.

- 데이터세트의 기존 대량 데이터를 모두 덮어쓰려면 Personalize 콘솔에서 기존 데이터 바꾸기를 선택하거나 [CreateDatasetImportJob](#) API 작업에서 FULL를 지정합니다. 이렇게 해도 실시간으로 기록된 이벤트를 포함하여 개별적으로 가져온 데이터는 교체되지 않습니다.
- 데이터 세트의 기존 데이터에 레코드를 추가하려면 기존 데이터에 추가를 선택하거나 CreateDatasetImportJob API 작업에서 INCREMENTAL을 지정합니다. Personalize는 동일한 ID를 가진 모든 레코드를 새 레코드로 대체합니다.

Note

데이터 세트 가져오기 작업과 함께 항목 상호 작용 데이터 세트 또는 작업 상호 작용 데이터 세트에 데이터를 추가하려면, 새 항목 상호 작용 또는 작업 상호 작용 레코드가 1,000개 이상 있어야 합니다.

대량 레코드 가져오기(콘솔)

Important

기본 설정으로 데이터세트 가져오기 작업은 대량으로 가져온 데이터세트의 기존 데이터를 대체합니다. 벌크 데이터를 이미 가져온 경우 작업의 [가져오기 모드](#)를 변경하여 데이터를 추가할 수 있습니다.

Personalize 콘솔을 사용하여 대량 레코드를 데이터세트로 가져오려면 이름, IAM 서비스 역할, 데이터 위치를 사용하여 데이터세트 가져오기 작업을 생성합니다.

[데이터세트 및 스키마 생성](#)에서 데이터세트를 방금 생성한 경우, 5단계로 건너뛰세요.

대량 레코드를 가져오려면(콘솔)

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터세트 그룹 페이지에서 데이터세트 그룹을 선택합니다. 데이터세트 그룹 개요가 표시됩니다.
3. 탐색 창에서 데이터 세트를 선택한 다음 대량 데이터를 가져올 데이터 세트를 선택합니다.
4. 데이터세트 가져오기 작업에서 데이터세트 가져오기 작업 생성을 선택합니다.
5. 데이터세트 가져오기 작업이 처음인 경우 데이터 가져오기 소스에서 S3에서 데이터 가져오기를 선택합니다.
6. 데이터세트 가져오기 작업 이름의 경우, 가져오기 작업의 이름을 지정합니다.
7. 이미 대량 데이터를 가져온 경우 가져오기 모드에서 데이터세트 업데이트 방법을 선택합니다. 기존 데이터 바꾸기 또는 기존 데이터에 추가 선택합니다. 데이터세트에 대한 첫 번째 작업인 경우에는 이 옵션이 나타나지 않습니다. 자세한 내용은 [기존 대량 데이터 업데이트](#) 단원을 참조하세요.
8. 데이터 가져오기 원본에서 데이터 위치에 대해 Amazon S3의 데이터 파일 저장 위치를 지정합니다. 다음 구문을 사용합니다.

s3://<name of your S3 bucket>/<folder path>/<CSV filename>

CSV 파일이 S3 버킷의 폴더에 있고 하나의 데이터세트 가져오기 작업으로 여러 CSV 파일을 데이터세트에 업로드하려는 경우, 폴더 경로를 지정할 수 있습니다. Personalize는 폴더의 첫 번째 수준에 있는 파일만 사용하며, 모든 하위 폴더의 데이터는 사용하지 않습니다. 폴더 이름 뒤에 /가 있는 다음 구문을 사용하세요.

s3://<name of your S3 bucket>/<folder path>/

- IAM 역할에서 새 역할 생성 또는 기존 역할 사용 중 하나를 선택합니다. 필수 조건을 완료한 경우 기존 서비스 역할 사용을 선택하고 [Personalize에 대한 IAM 역할 생성](#)에서 생성한 역할을 지정합니다.
- 지표 속성을 생성하고 이 작업과 관련된 지표를 S3에 게시하려는 경우, S3에 이벤트 지표 게시에서 이 가져오기 작업에 대한 지표 게시를 선택합니다.

아직 생성하지 않은 상태에서 이 작업에 대한 지표를 게시하려는 경우, 지표 속성 생성하기를 선택하여 다른 탭에서 새 지표를 생성합니다. 지표 속성을 생성한 후에 이 화면으로 돌아와 가져오기 작업 생성하기를 완료할 수 있습니다.

지표 속성에 대한 자세한 정보는 [추천의 영향 측정](#)(를) 참조하세요.

- 태그의 경우, 원하는 태그를 추가할 수 있습니다. Personalize 리소스 태그 지정에 대한 자세한 내용은 [Personalize 리소스에 태그 지정](#) 단원을 참조하세요.
- Start import(가져오기 시작)를 선택합니다. 데이터 가져오기 작업이 시작되고 대시보드 개요 페이지가 표시됩니다. 상태가 ACTIVE로 표시되면 데이터세트 가져오기가 완료된 것이고, Personalize 데이터세트로 데이터를 가져온 후에는, 데이터를 분석하고, S3 버킷으로 내보내고, 업데이트하거나, 데이터세트를 삭제하여 데이터를 삭제할 수 있습니다. 자세한 내용은 [데이터셋의 훈련 데이터 관리](#) 단원을 참조하세요.

데이터 가져오기를 마치면 솔루션을 생성할 준비가 된 것입니다. 자세한 내용은 [솔루션 및 솔루션 버전 생성](#) 단원을 참조하세요.

대량 레코드 가져오기(AWS CLI)

Important

기본 설정으로 데이터세트 가져오기 작업은 대량으로 가져온 데이터세트의 기존 데이터를 대체합니다. 이미 대량 데이터를 가져온 경우 작업 [가져오기 모드](#)를 변경하여 데이터를 추가할 수 있습니다.

를 사용하여 대량 레코드를 가져오려면 AWS CLI 명령어를 사용하여 데이터세트 가져오기 작업을 만드세요. [CreateDatasetImportJob](#) 이전에 데이터세트에 대한 데이터세트 가져오기 작업을 생성한 경우, 가져오기 모드 파라미터를 사용하여 새 데이터를 추가하는 방법을 지정할 수 있습니다. 기존 대량 데이터 업데이트에 대한 자세한 내용은 [기존 대량 데이터 업데이트](#) 단원을 참조하세요.

대량 레코드 가져오기(AWS CLI)

1. 다음 명령을 실행하여 데이터세트 가져오기 작업을 생성합니다. 데이터세트의 리소스 이름(ARN)을 제공하고 교육 데이터를 저장한 S3 버킷의 경로를 지정합니다. 경로에 맞추어 다음 구문을 사용합니다.

s3://<name of your S3 bucket>/<folder path>/<CSV filename>

CSV 파일이 S3 버킷의 폴더에 있고 하나의 데이터세트 가져오기 작업으로 여러 CSV 파일을 데이터세트에 업로드하려는 경우, 폴더 경로를 지정할 수 있습니다. Personalize는 폴더의 첫 번째 수준에 있는 파일만 사용하며, 모든 하위 폴더의 데이터는 사용하지 않습니다. 폴더 이름 뒤에 /가 있는 다음 구문을 사용하세요.

s3://<name of your S3 bucket>/<folder path>/

에서 생성한 AWS Identity and Access Management (IAM) 역할 Amazon 리소스 이름 (ARN)을 입력합니다. [Personalize에 대한 IAM 역할 생성](#) 기본 설정 import-mode는 FULL입니다. 자세한 내용은 [기존 대량 데이터 업데이트](#) 단원을 참조하세요. 이 작업에 대한 자세한 내용은 [CreateDatasetImportJob](#) 단원을 참조하세요.

```
aws personalize create-dataset-import-job \
  --job-name dataset import job name \
  --dataset-arn dataset arn \
  --data-source dataLocation=s3://bucketname/filename \
  --role-arn roleArn \
  --import-mode FULL
```

데이터세트 가져오기 작업 ARN은 다음 예제와 같이 표시됩니다.

```
{
  "datasetImportJobArn": "arn:aws:personalize:us-west-2:acct-id:dataset-import-job/
DatasetImportJobName"
}
```

2. describe-dataset-import-job 명령을 사용하여 상태를 확인합니다. 이전 단계에서 반환된 데이터세트 가져오기 작업 ARN을 제공합니다. 이 작업에 대한 자세한 내용은 [DescribeDatasetImportJob](#) 단원을 참조하세요.

```
aws personalize describe-dataset-import-job \
  --dataset-import-job-arn dataset import job arn
```

상태를 포함하여 데이터세트 가져오기 작업의 속성이 표시됩니다. 처음에 status는 CREATE PENDING으로 표시됩니다.

```
{
  "datasetImportJob": {
    "jobName": "Dataset Import job name",
    "datasetImportJobArn": "arn:aws:personalize:us-west-2:acct-id:dataset-import-job/DatasetImportJobArn",
    "datasetArn": "arn:aws:personalize:us-west-2:acct-id:dataset/DatasetGroupName/INTERACTIONS",
    "dataSource": {
      "dataLocation": "s3://<bucketname>/ratings.csv"
    },
    "importMode": "FULL",
    "roleArn": "role-arn",
    "status": "CREATE PENDING",
    "creationDateTime": 1542392161.837,
    "lastUpdatedDateTime": 1542393013.377
  }
}
```

상태가 ACTIVE로 표시되면 데이터세트 가져오기가 완료된 것이고, Personalize 데이터세트로 데이터를 가져온 후에는, 데이터를 분석하고, S3 버킷으로 내보내고, 업데이트하거나, 데이터세트를 삭제하여 데이터를 삭제할 수 있습니다. 자세한 내용은 [데이터셋의 훈련 데이터 관리단원](#)을 참조하세요.

데이터를 데이터세트 그룹의 관련 데이터세트로 가져온 후, 솔루션 버전(학습된 모델)을 생성할 수 있습니다. 자세한 내용은 [솔루션 및 솔루션 버전 생성](#) 단원을 참조하세요.

대량 레코드 가져오기(AWS SDK)

Important

기본 설정으로 데이터세트 가져오기 작업은 대량으로 가져온 데이터세트의 기존 데이터를 대체합니다. [이미 대량 데이터를 가져온 경우 작업의 가져오기 모드를 변경하여 데이터를 추가할 수 있습니다.](#)

데이터를 가져오려면 [CreateDatasetImportJob](#) 작업과 함께 데이터세트 가져오기 작업을 생성하세요. 다음 코드 예는 데이터세트 가져오기 작업을 생성하는 방법을 보여줍니다.

작업 이름을 지정하고, 데이터세트의 리소스 이름(ARN) `datasetArn`을 설정하고, `dataLocation`를 교육 데이터를 저장한 S3 버킷의 경로로 설정합니다. 경로에 맞추어 다음 구문을 사용합니다.

s3://<name of your S3 bucket>/<folder path>/<CSV filename>.csv

CSV 파일이 S3 버킷의 폴더에 있고 하나의 데이터세트 가져오기 작업으로 여러 CSV 파일을 데이터세트에 업로드하려는 경우, 폴더 경로를 지정할 수 있습니다. Personalize는 폴더의 첫 번째 수준에 있는 파일만 사용하며, 모든 하위 폴더의 데이터는 사용하지 않습니다. 폴더 이름 뒤에 /가 있는 다음 구문을 사용하세요.

s3://<name of your S3 bucket>/<folder path>/

의 `roleArn` 경우 Amazon Personalize에 S3 버킷에 액세스할 수 있는 권한을 부여하는 AWS Identity and Access Management (IAM) 역할을 지정하십시오. [Personalize에 대한 IAM 역할 생성](#) 단원을 참조하세요. 기본 설정 `importMode`는 FULL입니다. 이렇게 하면 데이터세트의 모든 대량 데이터가 대체됩니다. 데이터를 추가하려면 `ro` 설정합니다. INCREMENTAL 기존 대량 데이터 업데이트에 대한 자세한 내용은 [기존 대량 데이터 업데이트](#) 단원을 참조하세요.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_dataset_import_job(
    jobName = 'YourImportJob',
    datasetArn = 'dataset_arn',
    dataSource = {'dataLocation': 's3://bucket/file.csv'},
    roleArn = 'role_arn',
    importMode = 'FULL'
)

dsij_arn = response['datasetImportJobArn']

print ('Dataset Import Job arn: ' + dsij_arn)

description = personalize.describe_dataset_import_job(
    datasetImportJobArn = dsij_arn)['datasetImportJob']
```

```
print('Name: ' + description['jobName'])
print('ARN: ' + description['datasetImportJobArn'])
print('Status: ' + description['status'])
```

SDK for Java 2.x

```
public static String createPersonalizeDatasetImportJob(PersonalizeClient
personalizeClient,

                                                    String jobName,
                                                    String datasetArn,
                                                    String s3BucketPath,
                                                    String roleArn,
                                                    ImportMode importMode) {

    long waitInMilliseconds = 60 * 1000;
    String status;
    String datasetImportJobArn;

    try {
        DataSource importDataSource = DataSource.builder()
            .dataLocation(s3BucketPath)
            .build();

        CreateDatasetImportJobRequest createDatasetImportJobRequest =
        CreateDatasetImportJobRequest.builder()
            .datasetArn(datasetArn)
            .dataSource(importDataSource)
            .jobName(jobName)
            .roleArn(roleArn)
            .importMode(importMode)
            .build();

        datasetImportJobArn =
        personalizeClient.createDatasetImportJob(createDatasetImportJobRequest)
            .datasetImportJobArn();

        DescribeDatasetImportJobRequest describeDatasetImportJobRequest =
        DescribeDatasetImportJobRequest.builder()
            .datasetImportJobArn(datasetImportJobArn)
            .build();

        long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;
```



```

while (Instant.now().getEpochSecond() < maxTime) {

    DatasetImportJob datasetImportJob = personalizeClient
        .describeDatasetImportJob(describeDatasetImportJobRequest)
        .datasetImportJob();

    status = datasetImportJob.status();
    System.out.println("Dataset import job status: " + status);

    if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
        break;
    }
    try {
        Thread.sleep(waitInMilliseconds);
    } catch (InterruptedException e) {
        System.out.println(e.getMessage());
    }
}
return datasetImportJobArn;

} catch (PersonalizeException e) {
    System.out.println(e.awsErrorDetails().errorMessage());
}
return "";
}

```

SDK for JavaScript v3

```

// Get service clients and commands using ES6 syntax.
import { CreateDatasetImportJobCommand, PersonalizeClient } from
    "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
    region: "REGION"
});

// Set the dataset import job parameters.
export const datasetImportJobParam = {
    datasetArn: 'DATASET_ARN', /* required */
    dataSource: {
        dataLocation: 's3://<name of your S3 bucket>/<folderName>/<CSVfilename>.csv' /*
        required */
    }
}

```

```

    },
    jobName: 'NAME',           /* required */
    roleArn: 'ROLE_ARN',     /* required */
    importMode: "FULL"       /* optional, default is FULL */
  };

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
    CreateDatasetImportJobCommand(datasetImportJobParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};

run();

```

[DescribeDatasetImportJob](#) 작업의 응답에는 작업의 상태가 포함됩니다.

데이터를 사용하여 모델을 교육할 수 있기 전에 상태가 ACTIVE로 변경될 때까지 기다려야 합니다.

상태가 ACTIVE로 표시되면 데이터세트 가져오기가 완료된 것이고, Personalize 데이터세트로 데이터를 가져온 후에는, 데이터를 분석하고, S3 버킷으로 내보내고, 업데이트하거나, 데이터세트를 삭제하여 데이터를 삭제할 수 있습니다. 자세한 내용은 [데이터셋의 훈련 데이터 관리](#) 단원을 참조하세요.

데이터를 데이터세트 그룹의 관련 데이터세트로 가져온 후, 솔루션 버전(학습된 모델)을 생성할 수 있습니다. 자세한 내용은 [솔루션 및 솔루션 버전 생성](#) 단원을 참조하세요.

개별 레코드 가져오기

[데이터세트 및 스키마 생성](#) 작업을 완료한 후에는 항목 상호 작용, 사용자, 항목, 작업 또는 작업 상호 작용을 포함한 개별 레코드를 기존 데이터 세트로 가져올 수 있습니다. 데이터를 개별적으로 가져오면 카탈로그가 커짐에 따라 Amazon Personalize 데이터 세트에 소량의 레코드를 추가할 수 있습니다. 개별 가져오기 작업당 최대 10개의 레코드를 가져올 수 있습니다.

Apache Kafka를 사용하는 경우 Amazon Personalize용 Kafka 커넥터를 사용하여 Amazon Personalize로 데이터를 실시간으로 스트리밍할 수 있습니다. 자세한 내용은 [personalize-kafka-connector Github 리포지토리의 Amazon Personalize용 Kafka 커넥터](#)를 참조하세요.

다량의 과거 레코드가 있는 경우 먼저 데이터를 대량으로 가져온 다음 필요에 따라 데이터를 개별적으로 가져오는 것이 좋습니다. [Personalize 데이터세트로 데이터 바로 가져오기](#) 단원을 참조하세요.

개별 레코드 가져오기에 대한 필터 업데이트

Amazon Personalize는 마지막 개별 가져오기 후 20분 이내에 데이터 세트 그룹에서 만든 모든 필터를 새 상호작용, 항목 및 사용자 데이터로 업데이트합니다. 이 업데이트를 통해 캠페인에서 사용자에게 대한 추천을 필터링할 때 최신 데이터를 사용할 수 있습니다.

이미 추천자를 생성했거나 캠페인과 함께 사용자 지정 솔루션 버전을 배포한 경우, 새 개별 레코드가 추천에 영향을 미치는 방식은 도메인 사용 사례 또는 사용하는 레시피에 따라 달라집니다. 자세한 내용은 [새 데이터가 실시간 추천에 영향을 미치는 방식](#) 단원을 참조하세요.

주제

- [개별적으로 상호작용 가져오기](#)
- [개별적으로 사용자 가져오기](#)
- [개별적으로 항목 가져오기](#)
- [개별적으로 작업 가져오기](#)

개별적으로 상호작용 가져오기

[데이터세트 및 스키마 생성](#) 작업을 완료하여 항목 상호 작용 데이터 세트를 생성한 후에는 새 이벤트를 하나 이상 데이터 세트로 개별적으로 가져올 수 있습니다. 상호 작용 [이벤트](#)를 개별적으로 가져오려면 [이벤트 트래커](#)를 만든 다음 하나 이상의 이벤트를 항목 상호 작용 데이터 세트로 가져오면 됩니다. Amazon Personalize 콘솔을 사용하여 과거 개별 상호작용 이벤트를 가져오거나 AWS Command Line Interface(AWS CLI) 또는 AWS SDK를 사용하여 과거 또는 실시간 이벤트를 가져올 수 있습니다.

이 단원에는 Amazon Personalize 콘솔을 사용하여 이벤트를 가져오는 방법에 대한 정보가 포함되어 있습니다. Amazon Personalize 콘솔을 사용하는 경우 과거 이벤트만 가져오는 것이 좋습니다. AWS CLI 또는 AWS SDK를 사용하여 실시간으로 이벤트를 기록하는 방법에 대한 자세한 내용은 [이벤트 기록](#) 단원을 참조하세요.

Amazon Personalize가 새 레코드에 대한 필터를 업데이트하는 방법 및 새 레코드가 추천에 미치는 영향에 대한 자세한 내용은 [개별 레코드 가져오기](#) 단원을 참조하세요.

주제

- [이벤트 트래커 생성\(콘솔\)](#)
- [개별적으로 이벤트 가져오기\(콘솔\)](#)

이벤트 트래커 생성(콘솔)

Note

이벤트 트래커를 이미 만든 경우 [개별적으로 이벤트 가져오기\(콘솔\)](#)로 건너뛰어도 됩니다.

이벤트를 상호작용 데이터 세트로 가져오려면 먼저 데이터 세트 그룹에 대한 [이벤트 트래커](#)를 만들어야 합니다.

이벤트 트래커를 생성하려면(콘솔)

1. <https://console.aws.amazon.com/personalize/home> 에서 Amazon Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터 세트 그룹 페이지에서 이벤트를 가져오려는 항목 상호 작용 데이터 세트가 포함된 데이터 세트 그룹을 선택합니다.
3. 데이터 세트 그룹의 대시보드에서, 이벤트 통합 SDK 설치에서 시작을 선택합니다.
4. 트래커 구성 페이지의 트래커 구성에서 트래커 이름에 이벤트 트래커의 이름을 입력하고 다음을 선택합니다.
5. SDK 설치 페이지에는 새 이벤트 트래커의 추적 ID와 AWS Amplify 또는 AWS Lambda를 사용한 이벤트 데이터의 스트리밍 지침이 표시됩니다.

Amazon Personalize 콘솔을 사용하여 이벤트 데이터를 업로드하므로 이 정보는 무시해도 됩니다. 향후에 AWS Amplify 또는 AWS Lambda를 사용하여 이벤트 데이터를 스트리밍하려는 경우 이벤트 트래커 페이지에서 이벤트 트래커를 선택하면 이 정보를 볼 수 있습니다.

6. 마침을 클릭합니다. 이제 콘솔에서 이벤트를 가져오거나([개별적으로 이벤트 가져오기\(콘솔\)](#) 참조) PutEvents 작업을 사용하여 실시간으로 이벤트를 기록([이벤트 기록](#) 참조)할 수 있습니다.

개별적으로 이벤트 가져오기(콘솔)

이벤트 트래커를 만든 후 이벤트를 개별적으로 항목 상호 작용 데이터 세트로 가져올 수 있습니다. 이 절차에서는 항목 상호 작용 데이터 세트를 이미 생성했다고 가정합니다. 데이터 세트 생성에 대한 정보는 [데이터세트 및 스키마 생성](#) 단원을 참조하세요.

이벤트를 개별적으로 가져오려면(콘솔)

1. <https://console.aws.amazon.com/personalize/home> 에서 Amazon Personalize 콘솔을 열고 계정에 로그인합니다.

2. 데이터 세트 그룹 페이지에서 이벤트를 가져오려는 항목 상호 작용 데이터 세트가 포함된 데이터 세트 그룹을 선택합니다.
3. 탐색 창에서 데이터 세트를 선택합니다.
4. 데이터 세트 페이지에서, 상호작용 데이터 세트를 선택합니다.
5. 데이터 세트 세부 정보 페이지의 오른쪽 상단에서 데이터 세트 수정을 선택하고 레코드 생성을 선택합니다.
6. 사용자-항목 상호작용 레코드 생성 페이지에서 레코드 입력에 이벤트 세부 정보를 JSON 형식으로 입력합니다. 이벤트의 필드 이름 및 값은 항목 상호 작용 데이터 세트를 생성할 때 사용한 스키마와 일치해야 합니다. Amazon Personalize는 이 스키마의 필드 이름 및 데이터 유형이 포함된 JSON 템플릿을 제공합니다. 한 번에 최대 10개의 이벤트를 가져올 수 있습니다.
7. 레코드 생성을 선택합니다. 응답에서 가져오기 결과가 나열되고 성공 또는 실패 메시지가 표시됩니다.

개별적으로 사용자 가져오기

[데이터세트 및 스키마 생성](#)을 완료하여 사용자 데이터 세트를 생성한 후에는 한 명 이상의 새 사용자를 데이터 세트로 개별적으로 가져올 수 있습니다. 사용자를 개별적으로 가져오면 카탈로그가 확장됨에 따라 소량의 배치 가져오기를 통해 사용자 데이터 세트를 최신 상태로 유지할 수 있습니다. 한 번에 최대 10명까지 사용자를 가져올 수 있습니다. 신규 사용자가 많은 경우 먼저 데이터를 대량으로 가져온 다음 필요에 따라 사용자 데이터를 개별적으로 가져오는 것이 좋습니다. [Personalize 데이터세트로 데이터 바로 가져오기](#) 단원을 참조하세요.

Amazon Personalize 콘솔, AWS Command Line Interface (AWS CLI) 또는 AWS SDK를 사용하여 사용자를 가져올 수 있습니다. 사용자 데이터 세트에 이미 있는 사용자와 동일한 `userId`를 가져오는 경우 Amazon Personalize는 새 사용자로 대체합니다. 한 번에 최대 10명까지 사용자를 가져올 수 있습니다.

Amazon Personalize가 새 레코드에 대한 필터를 업데이트하는 방법 및 새 레코드가 추천에 미치는 영향에 대한 자세한 내용은 [개별 레코드 가져오기](#) 단원을 참조하세요.

주제

- [개별적으로 사용자 가져오기\(콘솔\)](#)
- [개별적으로 사용자 가져오기\(AWS CLI\)](#)
- [사용자를 개별적으로 가져오기 \(SDK\) AWS](#)

개별적으로 사용자 가져오기(콘솔)

한 번에 최대 10명까지 사용자를 가져올 수 있습니다. 이 절차에서는 사용자 데이터 세트를 이미 생성했다고 가정합니다. 데이터 세트 생성에 대한 정보는 [데이터세트 및 스키마 생성](#) 단원을 참조하세요.

사용자를 개별적으로 가져오려면(콘솔)

1. <https://console.aws.amazon.com/personalize/home> 에서 Amazon Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터 세트 그룹 페이지에서 사용자를 가져오려는 사용자 데이터 세트가 포함된 데이터 세트 그룹을 선택합니다.
3. 탐색 창에서 데이터 세트를 선택합니다.
4. 데이터 세트 페이지에서, 사용자 데이터 세트를 선택합니다.
5. 데이터 세트 세부 정보 페이지의 오른쪽 상단에서 데이터 세트 수정을 선택하고 레코드 생성을 선택합니다.
6. 사용자 레코드 생성 페이지에서 레코드 입력을 위해 사용자 세부 정보를 JSON 형식으로 입력합니다. 사용자의 필드 이름과 값은 사용자 데이터 세트를 생성할 때 사용한 스키마와 일치해야 합니다. Amazon Personalize는 이 스키마의 필드 이름 및 데이터 유형이 포함된 JSON 템플릿을 제공합니다.
7. 레코드 생성을 선택합니다. 응답에서 가져오기 결과가 나열되고 성공 또는 실패 메시지가 표시됩니다.

개별적으로 사용자 가져오기(AWS CLI)

[PutUsers](#) 작업을 사용하여 사용자 데이터 세트에 사용자를 한 명 이상 추가합니다. 한 번의 PutUsers 호출로 최대 10명까지 사용자를 가져올 수 있습니다. 이 단원에서는 사용자 데이터 세트를 이미 생성했다고 가정합니다. 데이터 세트 생성에 대한 정보는 [데이터세트 및 스키마 생성](#) 단원을 참조하세요.

다음 `put-users` 명령을 사용하여 AWS CLI로 사용자를 한 명 이상 추가할 수 있습니다. `dataset arn`을 데이터 세트의 Amazon 리소스 이름(ARN)으로 그리고 `user Id`를 사용자 ID로 바꿉니다. 동일한 `userId`가 이미 사용자 데이터 세트에 있는 경우 Amazon Personalize는 새 사용자로 대체합니다.

예를 들어 `properties`의 경우 사용자 데이터 세트의 각 필드에 대해 `propertyName`를 스키마의 필드 이름(낙타 대문자 형식)으로 바꿉니다. 예를 들어 `GENDER`는 `gender`이 되고, `MEMBERSHIP_TYPE`은 `membershipType`가 될 것입니다. `user data`을 해당 사용자에게 대한 데이터로 바꿉니다. 범주형 문자열 데이터의 경우 단일 속성에 여러 범주를 포함시키려면 파이프(`|`)로 각 범주를 구분합니다. 예를 들어 `"Premium Class|Legacy Member"`입니다.

```
aws personalize-events put-users \
  --dataset-arn dataset arn \
  --users '[{
    "userId": "user Id",
    "properties": "{\"propertyName\": \"user data\"}"
  },
  {
    "userId": "user Id",
    "properties": "{\"propertyName\": \"user data\"}"
  }]'
```

사용자를 개별적으로 가져오기 (SDK) AWS

[PutUsers](#) 작업을 사용하여 사용자 데이터 세트에 사용자를 한 명 이상 추가합니다. 동일한 `userId`가 이미 사용자 데이터 세트에 있는 경우 Amazon Personalize는 새 사용자로 대체합니다. 한 번의 `PutUsers` 호출로 최대 10명까지 사용자를 가져올 수 있습니다. 이 단원에서는 사용자 데이터 세트를 이미 생성했다고 가정합니다. 데이터 세트 생성에 대한 정보는 [데이터세트 및 스키마 생성](#) 단원을 참조하세요.

다음 코드는 사용자 데이터 세트에 사용자를 한 명 이상 추가하는 방법을 보여줍니다. 각 속성 이름 파라미터에 대해 스키마의 필드 이름을 낙타 대문자로 표기하여 전달합니다. 예를 들어 GENDER는 `gender`이 되고, MEMBERSHIP_TYPE은 `membershipType`가 될 것입니다. 각 속성 값 파라미터에 대해 해당 사용자에 대한 데이터를 전달합니다.

범주형 문자열 데이터의 경우 단일 속성에 여러 범주를 포함시키려면 파이프(|)로 각 범주를 구분합니다. 예를 들어 "Premium class|Legacy Member"입니다.

SDK for Python (Boto3)

```
import boto3

personalize_events = boto3.client(service_name='personalize-events')

personalize_events.put_users(
    datasetArn = 'dataset arn',
    users = [{
        'userId': 'user ID',
        'properties': "{\"propertyName\": \"user data\"}"
    },
    {
        'userId': 'user ID',
```

```
    'properties': "{\"propertyName\": \"user data\"}"  
  ]  
)
```

SDK for Java 2.x

```
public static int putUsers(PersonalizeEventsClient personalizeEventsClient,  
    String datasetArn,  
    String user1Id,  
    String user1PropertyName,  
    String user1PropertyValue,  
    String user2Id,  
    String user2PropertyName,  
    String user2PropertyValue) {  
  
    int responseCode = 0;  
    ArrayList<User> users = new ArrayList<>();  
  
    try {  
        User user1 = User.builder()  
            .userId(user1Id)  
            .properties(String.format("{\"%1$s\": \"%2$s\"}", user1PropertyName,  
user1PropertyValue))  
            .build();  
  
        users.add(user1);  
  
        User user2 = User.builder()  
            .userId(user2Id)  
            .properties(String.format("{\"%1$s\": \"%2$s\"}", user2PropertyName,  
user2PropertyValue))  
            .build();  
  
        users.add(user2);  
  
        PutUsersRequest putUsersRequest = PutUsersRequest.builder()  
            .datasetArn(datasetArn)  
            .build();  
  
        responseCode =  
personalizeEventsClient.putUsers(putUsersRequest).sdkHttpResponse().statusCode();  
        System.out.println("Response code: " + responseCode);  
        return responseCode;  
    }  
}
```



```
    } catch (PersonalizeEventsException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return responseCode;
}
```

SDK for JavaScript v3

```
import {
    PutUsersCommand,
    PersonalizeEventsClient,
} from "@aws-sdk/client-personalize-events";

const personalizeEventsClient = new PersonalizeEventsClient({
    region: "REGION",
});

// set the put users parameters
var putUsersParam = {
    datasetArn:
        "DATASET ARN",
    users: [
        {
            userId: "userId",
            properties: '{"column1Name": "value", "column2Name": "value"}',
        },
        {
            userId: "userId",
            properties: '{"column1Name": "value", "column2Name": "value"}',
        },
    ],
};

export const run = async () => {
    try {
        const response = await personalizeEventsClient.send(
            new PutUsersCommand(putUsersParam)
        );
        console.log("Success!", response);
        return response; // For unit tests.
    } catch (err) {
        console.log("Error", err);
    }
}
```

```
};
run();
```

개별적으로 항목 가져오기

[데이터세트 및 스키마 생성](#)을 완료하여 항목 데이터 세트를 생성한 후에는 하나 이상의 새 항목을 데이터 세트로 개별적으로 가져올 수 있습니다. 항목을 개별적으로 가져오면 카탈로그가 확장됨에 따라 소량의 배치 가져오기를 통해 항목 데이터 세트를 최신 상태로 유지할 수 있습니다. 한 번에 최대 10개의 항목을 가져올 수 있습니다. 다량의 새 항목이 있는 경우 먼저 데이터를 대량으로 가져온 다음 필요에 따라 항목 데이터를 개별적으로 가져오는 것이 좋습니다. [Personalize 데이터세트로 데이터 바로 가져오기](#) 섹션을 참조하십시오.

Amazon Personalize 콘솔, AWS Command Line Interface (AWS CLI) 또는 AWS SDK를 사용하여 항목을 가져올 수 있습니다. 항목 데이터 세트에 이미 있는 항목과 동일한 itemId을 가져오는 경우, Amazon Personalize는 새 항목으로 대체합니다.

Amazon Personalize가 새 레코드에 대한 필터를 업데이트하는 방법 및 새 레코드가 추천에 미치는 영향에 대한 자세한 내용은 [개별 레코드 가져오기](#) 단원을 참조하세요.

주제

- [개별적으로 항목 가져오기\(콘솔\)](#)
- [개별적으로 항목 가져오기\(AWS CLI\)](#)
- [항목을 개별적으로 가져오기 \(SDK\) AWS](#)

개별적으로 항목 가져오기(콘솔)

한 번에 최대 10개의 항목을 항목 데이터 세트로 가져올 수 있습니다. 이 절차에서는 항목 데이터 세트를 이미 생성했다고 가정합니다. 데이터 세트 생성에 대한 정보는 [데이터세트 및 스키마 생성](#) 단원을 참조하세요.

항목을 개별적으로 가져오려면(콘솔)

1. <https://console.aws.amazon.com/personalize/home> 에서 Amazon Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터 세트 그룹 페이지에서 항목을 가져오려는 항목 데이터 세트가 포함된 데이터 세트 그룹을 선택합니다.
3. 탐색 창에서 데이터 세트를 선택합니다.

4. 데이터 세트 페이지에서, 항목 데이터 세트를 선택합니다.
5. 데이터 세트 세부 정보 페이지의 오른쪽 상단에서 데이터 세트 수정을 선택한 후 레코드 생성을 선택합니다.
6. 항목 레코드 생성 페이지에서 레코드 입력에 항목 세부 정보를 JSON 형식으로 입력합니다. 항목의 필드 이름과 값은 항목 데이터 세트를 생성할 때 사용한 스키마와 일치해야 합니다. Amazon Personalize는 이 스키마의 필드 이름 및 데이터 유형이 포함된 JSON 템플릿을 제공합니다.
7. 레코드 생성을 선택합니다. 응답에서 가져오기 결과가 나열되고 성공 또는 실패 메시지가 표시됩니다.

개별적으로 항목 가져오기(AWS CLI)

[PutItems](#) 작업을 사용하여 항목 데이터 세트에 항목 하나 이상 추가합니다. 한 번의 PutItems 호출로 최대 10개의 항목을 가져올 수 있습니다. 이 단원에서는 항목 데이터 세트를 이미 생성했다고 가정합니다. 데이터 세트 생성에 대한 정보는 [데이터세트 및 스키마 생성](#) 단원을 참조하세요.

다음 `put-items` 명령을 사용하여 AWS CLI로 항목을 하나 이상 추가할 수 있습니다. `dataset arn`을 데이터 세트의 Amazon 리소스 이름(ARN)으로 그리고 `item Id`를 항목 ID로 바꿉니다. 동일한 `itemId`가 있는 항목이 항목 데이터 세트에 이미 있는 경우 Amazon Personalize는 새 항목으로 대체합니다.

예를 들어 `properties`의 경우 항목 데이터 세트의 각 필드에 대해 `propertyName`를 스키마의 필드 이름(낙타 대문자 형식)으로 바꿉니다. 예를 들어 GENRES는 `genres`이 되고 CREATION_TIMESTAMP는 `creationTimestamp`가 될 것입니다. `item data`을 해당 항목의 데이터로 대체합니다. CREATION_TIMESTAMP 데이터는 [Unix epoch 시간 형식](#)이고 초 단위여야 합니다. 범주형 문자열 데이터의 경우 단일 속성에 여러 범주를 포함시키려면 파이프(`|`)로 각 범주를 구분합니다. 예를 들어 `"Horror|Action"`입니다.

```
aws personalize-events put-items \
  --dataset-arn dataset arn \
  --items '[{
    "itemId": "item Id",
    "properties": "{\"propertyName\": \"\i>item data\i>\"}"
  },
  {
    "itemId": "item Id",
    "properties": "{\"propertyName\": \"\i>item data\i>\"}"
  }]'
```

항목을 개별적으로 가져오기 (SDK) AWS

[PutItems](#) 작업을 사용하여 항목 데이터 세트에 항목을 하나 이상 추가합니다. 한 번의 PutItems 호출로 최대 10개의 항목을 가져올 수 있습니다. 동일한 itemId이 있는 항목이 항목 데이터 세트에 이미 있는 경우 Amazon Personalize는 새 항목으로 대체합니다. 이 단원에서는 항목 데이터 세트를 이미 생성했다고 가정합니다. 데이터 세트 생성에 대한 정보는 [데이터세트 및 스키마 생성 단원](#)을 참조하세요.

다음 코드는 항목 데이터 세트에 항목을 하나 이상 추가하는 방법을 보여줍니다. 각 속성 이름 파라미터에 대해 스키마의 필드 이름을 낙타 대문자로 표기하여 전달합니다. 예를 들어 GENRES는 genres이 되고 CREATION_TIMESTAMP는 creationTimestamp가 될 것입니다. 각 속성 값 파라미터에 대해 항목의 데이터를 전달합니다. CREATION_TIMESTAMP 데이터는 [Unix epoch 시간 형식](#)이고 초 단위여야 합니다.

범주형 문자열 데이터의 경우 단일 속성에 여러 범주를 포함시키려면 파이프(|)로 각 범주를 구분합니다. 예를 들어 "Horror|Action"입니다.

SDK for Python (Boto3)

```
import boto3

personalize_events = boto3.client(service_name='personalize-events')

personalize_events.put_items(
    datasetArn = 'dataset arn',
    items = [{
        'itemId': 'item ID',
        'properties': "{\"propertyName\": \"item data\"}"
    },
    {
        'itemId': 'item ID',
        'properties': "{\"propertyName\": \"item data\"}"
    }
])
```

SDK for Java 2.x

```
public static int putItems(PersonalizeEventsClient personalizeEventsClient,
                          String datasetArn,
                          String item1Id,
                          String item1PropertyName,
                          String item1PropertyValue,
                          String item2Id,
```

```
        String item2PropertyName,
        String item2PropertyValue) {

    int responseCode = 0;
    ArrayList<Item> items = new ArrayList<>();

    try {
        Item item1 = Item.builder()
            .itemId(item1Id)
            .properties(String.format("{\\"%1$s\\": \\"%2$s\\"}",
                item1PropertyName, item1PropertyValue))
            .build();

        items.add(item1);

        Item item2 = Item.builder()
            .itemId(item2Id)
            .properties(String.format("{\\"%1$s\\": \\"%2$s\\"}",
                item2PropertyName, item2PropertyValue))
            .build();

        items.add(item2);

        PutItemsRequest putItemsRequest = PutItemsRequest.builder()
            .datasetArn(datasetArn)
            .items(items)
            .build();

        responseCode =
personalizeEventsClient.putItems(putItemsRequest).sdkHttpResponse().statusCode();
        System.out.println("Response code: " + responseCode);
        return responseCode;

    } catch (PersonalizeEventsException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return responseCode;
}
```

SDK for JavaScript v3

```
import {
    PutItemsCommand,
```

```
PersonalizeEventsClient,
} from "@aws-sdk/client-personalize-events";

const personalizeEventsClient = new PersonalizeEventsClient({
  region: "REGION",
});

// set the put items parameters
var putItemsParam = {
  datasetArn:
    "DATASET ARN",
  items: [
    {
      itemId: "itemId",
      properties: '{"column1Name": "value", "column2Name": "value"}',
    },
    {
      itemId: "itemId",
      properties: '{"column1Name": "value", "column2Name": "value"}',
    },
  ],
};

export const run = async () => {
  try {
    const response = await personalizeEventsClient.send(
      new PutItemsCommand(putItemsParam)
    );
    console.log("Success!", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};

run();
```

개별적으로 작업 가져오기

[데이터세트 및 스키마 생성](#) 작업을 완료하여 [작업 데이터 세트](#)를 생성한 후에는 새 작업을 하나 이상 데이터 세트로 개별적으로 가져올 수 있습니다. 작업을 개별적으로 가져오면 카탈로그가 확장되어 소

량의 배치 가져오기를 통해 작업 데이터 세트를 최신 상태로 유지할 수 있습니다. 한 번에 최대 10개 까지 작업을 가져올 수 있습니다. 신규 작업이 많은 경우 먼저 데이터를 대량으로 가져온 후 필요에 따라 작업 데이터를 개별적으로 가져오는 것이 좋습니다. [Personalize 데이터세트로 데이터 바로 가져오기](#)를 참조하세요.

Amazon Personalize 콘솔, AWS Command Line Interface(AWS CLI) 또는 AWS SDK를 사용하여 작업을 가져올 수 있습니다. 작업 데이터 세트에 이미 있는 작업과 동일한 actionId가 있는 작업을 가져오는 경우, Amazon Personalize는 새 작업으로 대체합니다.

새 레코드가 추천에 영향을 미치는 방식에 대한 자세한 내용은 [새 데이터가 실시간 추천에 영향을 미치는 방식](#) 섹션을 참조하세요.

주제

- [개별적으로 작업 가져오기\(콘솔\)](#)
- [개별적으로 작업 가져오기\(AWS CLI\)](#)
- [개별적으로 작업 가져오기\(AWS SDK\)](#)

개별적으로 작업 가져오기(콘솔)

한 번에 최대 10개의 작업을 작업 데이터 세트로 가져올 수 있습니다. 이 섹션에서는 작업 데이터 세트를 이미 생성했다고 가정합니다. 데이터 세트 생성에 대한 정보는 [데이터세트 및 스키마 생성](#) 단원을 참조하세요.

작업을 개별적으로 가져오려면(콘솔)

1. <https://console.aws.amazon.com/personalize/home> 에서 Amazon Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터 세트 그룹 페이지에서 추가하려는 작업 데이터 세트가 포함된 데이터 세트 그룹을 선택합니다.
3. 탐색 창에서 데이터 세트를 선택합니다.
4. 데이터 세트 페이지에서 작업 데이터 세트를 선택합니다.
5. 데이터 세트 세부 정보 페이지의 오른쪽 상단에서 데이터 세트 수정을 선택한 후 레코드 생성을 선택합니다.
6. 작업 레코드 생성 페이지에서 레코드 입력에 작업 세부 정보를 JSON 형식으로 입력합니다. 작업의 필드 이름과 값은 작업 데이터 세트를 생성할 때 사용한 스키마와 일치해야 합니다. Amazon Personalize는 이 스키마의 필드 이름 및 데이터 유형이 포함된 JSON 템플릿을 제공합니다.

7. 레코드 생성을 선택합니다. 응답에서 가져오기 결과가 나열되고 성공 또는 실패 메시지가 표시됩니다.

개별적으로 작업 가져오기(AWS CLI)

PutActions API 작업을 사용하여 작업 데이터 세트에 작업을 하나 이상 추가합니다. 한 번에 최대 10개까지 작업을 가져올 수 있습니다. 이 섹션에서는 작업 데이터 세트를 이미 생성했다고 가정합니다. 데이터 세트 생성에 대한 정보는 [데이터세트 및 스키마 생성](#) 단원을 참조하세요.

다음 put-actions 명령을 사용하여 AWS CLI로 작업을 하나 이상 추가할 수 있습니다. dataset arn을 데이터 세트의 Amazon 리소스 이름(ARN)으로, actionId를 작업 ID로 바꿉니다. 동일한 actionId의 작업이 이미 작업 데이터 세트에 있는 경우 Amazon Personalize는 새 작업으로 대체합니다.

properties의 경우 작업 데이터 세트의 각 필드에 대해 propertyName을 스키마의 필드 이름(낙타 대문자 형식)으로 바꿉니다. 예를 들어, ACTION_EXPIRATION_TIMESTAMP는 actionExpirationTimestamp이 되고 CREATION_TIMESTAMP는 creationTimestamp가 됩니다. property data을 해당 속성에 대한 데이터로 바꿉니다.

```
aws personalize-events put-actions \
  --dataset-arn dataset arn \
  --actions '[{
    "actionId": "actionId",
    "properties": "{\"propertyName\": \"\i>property data\"}"
  },
  {
    "actionId": "actionId",
    "properties": "{\"propertyName\": \"\i>property data\"}"
  }
]'
```

개별적으로 작업 가져오기(AWS SDK)

PutActions 작업을 사용하여 작업 데이터 세트에 작업을 하나 이상 추가합니다. 한 번의 PutActions 호출로 최대 10개까지 작업을 가져올 수 있습니다. 동일한 actionId의 작업이 이미 작업 데이터 세트에 있는 경우 Amazon Personalize는 새 작업으로 대체합니다. 이 섹션에서는 작업 데이터 세트를 이미 생성했다고 가정합니다. 데이터 세트 생성에 대한 정보는 [데이터세트 및 스키마 생성](#) 단원을 참조하세요.

다음 코드는 작업 데이터 세트에 작업을 하나 이상 추가하는 방법을 보여줍니다. 각 작업에 대해 actionId를 지정합니다. 동일한 actionId의 작업이 이미 작업 데이터 세트에 있는 경우

Amazon Personalize는 새 작업으로 대체합니다. `properties`의 경우 작업 데이터 세트의 각 추가 필드에 대해 `propertyName`를 스키마의 필드 이름(낙타 대문자 형식)으로 바꿉니다. 예를 들어, `ACTION_EXPIRATION_TIMESTAMP`는 `actionExpirationTimestamp`이 되고 `CREATION_TIMESTAMP`는 `creationTimestamp`가 됩니다. `property data`을 해당 속성에 대한 데이터로 바꿉니다.

```
import boto3

personalize_events = boto3.client(service_name='personalize-events')

personalize_events.put_actions(
    datasetArn = 'dataset arn',
    actions = [{
        'actionId': 'actionId',
        'properties': "{\"propertyName\": \"property value\"}"
    },
    {
        'actionId': 'actionId',
        'properties': "{\"propertyName\": \"property value\"}"
    }
])
```

3단계: 추천 또는 사용자 지정 리소스 생성

데이터를 가져오고 나면 권장 사항이나 사용자 지정 리소스를 생성할 준비가 된 것입니다. 다음 리소스를 사용하여 권장 사항을 받으세요. 생성하는 리소스는 데이터 세트 그룹 유형에 따라 달라집니다.

- 도메인 데이터 세트 그룹의 경우, 도메인을 기반으로 사전 정의된 사용 사례에 대한 추천을 생성합니다. 추천을 사용하여 권장 사항을 받을 수 있습니다. 제공된 사용 사례에 대한 자세한 내용은 [사용 사례 선택](#) 단원을 참조하세요. 도메인 데이터 세트 그룹에 사용자 지정 리소스를 추가할 수도 있습니다. 여기에는 사용자 지정 사용 사례에 맞추어 교육된 솔루션 및 솔루션 버전이 포함됩니다.
- 사용자 지정 데이터 세트 그룹의 경우, 레시피를 사용하여 솔루션을 구성합니다. 그런 다음 솔루션 버전을 생성합니다(모델 교육). 사용 가능한 레시피에 대한 자세한 내용은 [레시피 선택](#) 단원을 참조하세요.

실시간 권장 사항의 경우, 캠페인에 솔루션 버전을 배포합니다. 배치 권장 사항 및 사용자 세그먼트의 경우, 캠페인이 필요하지 않습니다.

주제

- [도메인 추천 생성](#)
- [사용자 지정 리소스 생성](#)

도메인 추천 생성

데이터를 가져오고 나면 도메인 데이터 세트 그룹에서 추천을 생성하고, 평가하고, 관리할 준비가 된 것입니다. 추천은 권장 사항을 생성하는 도메인 데이터 세트 그룹 리소스입니다. 애플리케이션에서 이를 사용하여 [GetRecommendations](#) 작업과 함께 실시간 권장 사항을 받을 수 있습니다.

주제

- [추천 생성](#)
- [추천자 평가](#)
- [추천자 관리](#)

추천 생성

도메인 데이터세트 그룹을 생성하고 데이터를 가져오고 나면, 도메인 사용 사례에 맞는 추천을 생성할 수 있습니다. 추천은 추천을 생성하는 도메인 데이터세트 그룹 리소스입니다. 애플리케이션에서 추천을 사용하여 [GetRecommendations](#) 작업과 함께 실시간 추천을 받을 수 있습니다.

추천을 생성하고 나면 사용 사례를 지정하고 Personalize에서 사용 사례에 가장 적합한 구성으로 추천자를 지원하는 모델을 교육합니다. 사용 사례마다 추천을 받기 위한 API 요구사항이 다릅니다. 도메인 별 추천 사용 사례 목록은 [사용 사례 선택](#) 단원을 참조하세요. 리전당 최대 15개의 추천을 생성할 수 있습니다.

Personalize는 7일마다 추천자를 지원하는 모델을 자동으로 재학습합니다. 이는 데이터세트의 전체 데이터를 기반으로 완전히 새로운 모델을 생성하는 전체 재교육입니다. 가장 적합한 추천 제품 및 사용자를 위한 추천 제품 사용 사례를 사용하여 Personalize는 2시간마다 기존 모델을 업데이트하여 탐색과 함께 추천에 새 항목을 포함시킵니다.

추천자를 생성하면 추천에서 항목 메타데이터를 활성화할 수 있습니다. 자세한 정보는 [추천에서 메타데이터 활성화](#)를 참조하세요.

Amazon Personalize 콘솔 AWS Command Line Interface (AWS CLI) 또는 SDK를 사용하여 추천자를 생성할 수 있습니다. AWS

추천 상태

추천자는 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

추천 상태를 확인하려면 Personalize 콘솔의 추천 페이지로 이동하거나 [DescribeRecommender](#) 작업을 사용합니다.

주제

- [초당 최소 추천 요청 수 및 Auto Scaling](#)
- [추천에서 메타데이터 활성화](#)
- [교육 시 사용되는 열 구성](#)
- [추천 생성\(콘솔\)](#)
- [추천 생성\(AWS CLI\)](#)
- [추천인 \(SDK\)AWS 만들기](#)

초당 최소 추천 요청 수 및 Auto Scaling

Important

`minRecommendationRequestsPerSecond`가 높으면 요금이 인상됩니다. `minRecommendationRequestsPerSecond`(기본값)의 경우 1부터 시작하는 것이 좋습니다. Amazon CloudWatch 지표를 사용하여 사용량을 추적하고 `minRecommendationRequestsPerSecond` 필요에 따라 사용량을 늘리십시오.

추천을 생성할 때 추천의 초당 최소 추천 요청을 구성할 수 있습니다. 초당 최소 추천 요청(`minRecommendationRequestsPerSecond`)은 Personalize에서 프로비저닝하는 기본 추천 요청 처리량을 지정합니다. 기본값은 `minRecommendationRequestsPerSecond`입니다¹. 추천 요청은 단일한 `GetRecommendations` 작업입니다. 요청 처리량은 초당 요청 수로 측정되며 Personalize는 초당 요청을 사용하여 시간당 요청 및 추천 사용 요금을 도출합니다.

초당 요청의 수가 `minRecommendationRequestsPerSecond` 이상으로 증가할 경우, Personalize는 프로비저닝된 용량을 자동으로 확장 및 축소하지만 `minRecommendationRequestsPerSecond` 미만으로는 절대 그렇게 하지 않습니다. 용량이 증가 중에 잠시 동안의 지연으로 요청의 손실이 발생할 수 있습니다.

요금은 시간당 최소 요청 수 (기준 `minRecommendationRequestsPerSecond`) 와 실제 요청 수 중 더 큰 금액입니다. 실제 요청 처리량은 1시간 이내의 초당 평균 요청 수로 계산됩니다. `minRecommendationRequestsPerSecond` 기본값부터 시작하여 Amazon CloudWatch 메트릭을 사용하여 사용량을 추적한 다음 `minRecommendationRequestsPerSecond` 필요에 따라 늘리는 것이 좋습니다.

추천에서 메타데이터 활성화

Important

추천에서 메타데이터를 활성화하면 추가 비용이 발생합니다. 자세한 내용은 [Amazon Personalize 요금](#)을 참조하세요.

추천자를 생성할 때 추천 결과에 항목 데이터 세트의 항목 메타데이터를 포함하는 옵션을 활성화할 수 있습니다. 활성화한 경우 추천 요청에 항목 데이터 세트의 열을 지정할 수 있습니다. Amazon Personalize는 추천 응답의 각 항목에 대해 이 데이터를 반환합니다.

메타데이터를 통해 사용자 인터페이스에서 추천을 강화할 수 있습니다(예: 캐러셀에 영화 장르 추가). 또는 이를 사용하여 추천 품질을 시각적으로 평가할 수도 있습니다. 앱에서 생성형 AI를 사용하는 경우 메타데이터를 AI 프롬프트에 연결하여 관련성이 더 높은 콘텐츠를 생성할 수 있습니다. Amazon Personalize를 생성형 AI와 함께 사용하는 방법에 대한 자세한 내용은 [Amazon Personalize 및 생성형 AI](#) 섹션을 참조하세요.

- Amazon Personalize 콘솔에서 메타데이터를 활성화하려면 추천자를 생성할 때 고급 구성의 추천 결과에서 항목 메타데이터 반환을 선택합니다. 자세한 정보는 [추천 생성\(콘솔\)](#)을 참조하세요.
- AWS SDK를 `enableMetadataWithRecommendations` 사용하여 `true` 메타데이터를 활성화하거나 AWS CLI, `recommenderConfig` 설정된 상태에서 `CreateRecommender` API 작업을 사용하십시오. 자세한 내용은 [추천 생성\(AWS CLI\)](#) 또는 [추천인 \(SDK\)AWS 만들기](#)을 참조하세요.

추천에 메타데이터를 추가하려면 메타데이터 열이 포함된 항목 데이터 세트가 있어야 합니다. 교육 시 메타데이터를 사용할 필요는 없습니다. 데이터 세트 생성에 대한 내용은 [데이터세트 및 스키마 생성](#) 섹션을 참조하세요. 데이터 관리 및 업데이트에 대한 내용은 [데이터셋의 훈련 데이터 관리](#) 섹션을 참조하세요.

교육 시 사용되는 열 구성

추천자를 생성할 때 Amazon Personalize가 추천자를 지원하는 모델을 교육할 때 고려하는 열을 수정할 수 있습니다.

이를 통해 다양한 교육 데이터 조합을 실험해 볼 수 있습니다. 또는 중요한 데이터가 없는 열을 제외할 수도 있습니다. 예를 들어, 추천을 필터링하는 데만 사용하려는 열이 있을 수 있습니다. 이 열을 교육에서 제외할 수 있으며, Personalize는 필터링할 때만 이 열을 고려합니다.

EVENT_TYPE 열은 제외할 수 없습니다. 기본 설정으로 Personalize는 교육 시 사용할 수 있는 모든 열을 사용합니다. 다음 데이터는 항상 교육에서 제외됩니다.

- 부울 데이터 유형의 열
- [임프레션 데이터](#)
- 범주형 또는 텍스트형이 아닌 사용자 지정 문자열 필드

교육에 노출 데이터를 포함할 수는 없지만 사용 사례 또는 레시피에서 사용하는 경우 Amazon Personalize는 권장 사항을 받을 때 노출 데이터를 사용하여 탐색을 안내합니다.

- Amazon Personalize 콘솔을 사용하여 학습할 때 사용되는 열을 구성하려면 추천자를 생성할 때 고급 구성 페이지에서 사용할 열을 선택합니다. 자세한 정보는 [추천 생성\(콘솔\)](#)을 참조하세요.
- AWS SDK를 사용하여 학습할 때 사용되는 열을 구성하려면 [CreateRecommender](#) API 작업을 사용하고 `recommenderConfig` 다음을 지정하십시오. AWS CLI `excludeDatasetColumns trainingDataConfig` 코드 샘플은 [훈련 시 사용되는 열 구성 \(AWS CLI\)](#) 또는 [훈련 시 사용되는 열 구성 \(AWS SDK\)](#) 을 참조하십시오.

추천 생성(콘솔)

Important

`minRecommendationRequestsPerSecond`가 높으면 요금이 인상됩니다. `minRecommendationRequestsPerSecond`(기본값)의 경우 1부터 시작하는 것이 좋습니다. Amazon CloudWatch 지표를 사용하여 사용량을 추적하고 `minRecommendationRequestsPerSecond` 필요에 따라 사용량을 늘리십시오. 자세한 내용은 [초당 최소 추천 요청 수 및 Auto Scaling](#) 단원을 참조하세요.

Personalize 콘솔을 사용하여 각 사용 사례에 대한 추천을 생성합니다. 방금 도메인 데이터세트 그룹을 생성했고 이미 개요 페이지에 있는 경우, 3단계로 건너뛰세요.

추천을 생성하려면

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터세트 그룹 페이지에서 도메인 데이터세트 그룹을 선택합니다.
3. 3단계에서 <도메인 이름> 추천자 사용을 선택하고 추천자 생성을 선택합니다.
4. 사용 사례 선택 페이지에서 추천을 생성하려는 사용 사례를 선택하고 각 사례에 추천 이름을 지정합니다. Personalize는 선택한 각 사용 사례에 대한 추천을 생성합니다. 사용 가능한 사용 사례는 도메인에 따라 다릅니다. 사용 사례 선택에 대한 자세한 내용은 [사용 사례 선택](#) 단원을 참조하세요.
5. 다음을 선택합니다.
6. 고급 구성 페이지에서 비즈니스 니즈에 따라 각 추천을 구성하세요.

- 추천 사용 사례에서 사용하는 각 데이터세트에 대해 Personalize가 추천을 지원하는 모델을 학습시킬 때 고려하는 열을 선택할 수 있습니다. 기본 설정으로 Personalize는 교육 시 사용할 수 있는 모든 열을 사용합니다. 자세한 정보는 [교육 시 사용되는 열 구성](#)을 참조하세요.
- 초당 최소 추천 요청을 수정하여 추천에 대한 새로운 최소 요청 용량을 지정할 수 있습니다. `minRecommendationRequestsPerSecond`가 높으면 요금이 인상됩니다. (기본값)의 경우 1부터 시작하는 것이 좋습니다. Amazon CloudWatch 지표를 사용하여 사용량을 추적하고 `minRecommendationRequestsPerSecond` 필요에 따라 사용량을 늘리십시오. 자세한 내용은 [초당 최소 추천 요청 수 및 Auto Scaling](#) 단원을 참조하세요.
- 추천과 함께 항목 데이터 세트 메타데이터를 포함할 수 있도록 하려면 추천 결과에서 항목 메타데이터 반환을 선택합니다. 활성화한 경우 추천 또는 개인 맞춤형 순위 요청에 항목 데이터 세트의 열을 지정할 수 있습니다. Amazon Personalize는 추천 응답의 각 항목에 대해 이 데이터를 반환합니다.

메타데이터를 활성화하려면 메타데이터 열이 있는 항목 데이터 세트가 있어야 합니다.

- `Top picks for your` 또는 `Recommended for you` 사용 사례의 경우, 원하는 대로 탐색 구성을 변경합니다. 탐색에는 상호작용 데이터가 거의 없는 항목에 대한 사용자의 반응을 알아보기 위해 다양한 항목 추천을 테스트하는 작업이 포함됩니다. 다음 필드를 사용하여 탐색을 구성하세요.
 - 관련성이 낮은 항목 탐색에 주목(탐색 가중치) — 탐색 범위를 설정합니다. 0과 1 사이에서 소수 값을 지정합니다. 기본값은 0.3입니다. 값이 1에 가까울수록 탐색이 더 많아집니다. 더 많이 탐색할수록 항목 상호 작용 데이터가 적거나 이전 행동을 기반으로 한 관련성이 낮은 항목이 더 많이 추천에 포함됩니다. 0이면 탐색이 수행되지 않으며 최신 데이터(관련성)를 기반으로 추천이 제공됩니다.

- 탐색 항목 기간 제한 – 항목 상호 작용 데이터 세트의 모든 항목에서 최근 상호 작용이 발생한 이후 최대 항목 사용 기간(일)을 지정합니다. 이렇게 하면 항목 기간에 따른 항목 탐색 범위가 정의됩니다. Amazon Personalize는 생성 타임스탬프 또는 생성 타임스탬프 데이터가 누락된 경우 항목 상호 작용 데이터를 기반으로 항목 기간을 결정합니다. Personalize가 항목 기간을 결정하는 방법에 대한 자세한 내용은 [생성 타임스탬프 데이터](#) 단원을 참조하세요.

Personalize가 탐색 중에 고려하는 항목을 늘리려면 더 큰 값을 입력합니다. 최소값은 1일이고, 기본값은 30일입니다. 지정한 항목 보관 기간 한도보다 오래된 항목이 추천에 포함될 수 있습니다. 이는 이러한 항목이 사용자와 관련이 있고 탐색에서 해당 항목을 식별할 수 없었기 때문입니다.

- 태그의 경우, 원하는 태그를 추가할 수 있습니다. Personalize 리소스 태그 지정에 대한 자세한 내용은 [Personalize 리소스에 태그 지정](#) 단원을 참조하세요.

7. 추천자 생성을 선택하여 각 사용 사례에 적합한 추천자를 생성합니다.

추천자 페이지에서 각 추천자의 상태를 모니터링할 수 있습니다. 추천 상태가 활성화되면 신청서에서 이를 사용하여 추천을 받을 수 있습니다.

추천 생성(AWS CLI)

도메인 데이터세트 그룹을 생성하고 데이터를 가져오고 나면, 도메인 사용 사례에 맞는 추천을 생성할 수 있습니다. 추천은 추천을 생성하는 도메인 데이터세트 그룹 리소스입니다.

Top picks for your 또는 Recommended for you 사용 사례에 대해, Personalize는 항목을 추천할 때 탐색을 사용합니다. 자세한 내용은 [탐색 구성](#) 단원을 참조하세요.

주제

- [추천 생성](#)
- [교육 시 사용되는 열 구성](#)
- [탐색 구성](#)
- [추천에서 메타데이터 활성화](#)

추천 생성

다음 AWS CLI 코드를 사용하여 도메인 사용 사례에 맞는 추천자를 만드세요. 각 도메인 사용 사례에 대해 이 코드를 실행합니다. recipeArn의 경우, 사용 사례를 위해 리소스 이름(ARN)을 제공합니다. 사용 가능한 사용 사례는 도메인에 따라 다릅니다. 사용 사례 및 ARN 목록은 [사용 사례 선택](#) 단원을 참조하세요.

```
aws personalize create-recommender \
--name recommender name \
--dataset-group-arn dataset group ARN \
--recipe-arn recipe ARN
```

교육 시 사용되는 열 구성

학습에서 열을 제외하려면 추천 구성의 일부로 `trainingDataConfig`의 `excludedDatasetColumns` 객체를 제공하세요. 객체의 각 키에 대해 데이터세트 유형을 제공하세요. 각 값에 대해 제외할 열 목록을 제공하세요. 자세한 정보는 [교육 시 사용되는 열 구성](#)을 참조하세요.

```
aws personalize create-recommender \
--name recommender name \
--dataset-group-arn dataset group ARN \
--recipe-arn recipe ARN \
--recommender-config "{\"trainingDataConfig\": {\"excludedDatasetColumns\": { \"datasetType\": [ \"column1Name\", \"column2Name\"]}}}"
```

탐색 구성

Top picks for your 또는 Recommended for you 사용 사례에 대해, Personalize는 항목을 추천할 때 탐색을 사용합니다. 탐색에는 상호작용 데이터가 거의 없는 항목에 대한 사용자의 반응을 알아보기 위해 다양한 항목 추천을 테스트하는 작업이 포함됩니다. 다음과 같이 탐색을 구성할 수 있습니다.

- 관련성이 낮은 항목 탐색에 주목(탐색 가중치) — 탐색 범위를 설정합니다. 0과 1 사이에서 소수 값을 지정합니다. 기본값은 0.3입니다. 값이 1에 가까울수록 탐색이 더 많아집니다. 더 많이 탐색할수록 항목 상호 작용 데이터가 적거나 이전 행동을 기반으로 한 관련성이 낮은 항목이 더 많이 추천에 포함됩니다. 0이면 탐색이 수행되지 않으며 최신 데이터(관련성)를 기반으로 추천이 제공됩니다.
- 탐색 항목 기간 제한 – 항목 상호 작용 데이터 세트의 모든 항목에서 최근 상호 작용이 발생한 이후 최대 항목 사용 기간(일)을 지정합니다. 이렇게 하면 항목 기간에 따른 항목 탐색 범위가 정의됩니다. Amazon Personalize는 생성 타임스탬프 또는 생성 타임스탬프 데이터가 누락된 경우 항목 상호 작용 데이터를 기반으로 항목 기간을 결정합니다. Personalize가 항목 기간을 결정하는 방법에 대한 자세한 내용은 [생성 타임스탬프 데이터](#) 단원을 참조하세요.

Personalize가 탐색 중에 고려하는 항목을 늘리려면 더 큰 값을 입력합니다. 최소값은 1일이고, 기본값은 30일입니다. 지정한 항목 보관 기간 한도보다 오래된 항목이 추천에 포함될 수 있습니다. 이는 이러한 항목이 사용자와 관련이 있고 탐색에서 해당 항목을 식별할 수 없었기 때문입니다.

다음 코드는 Top picks for you 사용 사례에 맞는 추천을 생성할 때 탐색을 구성하는 방법을 보여 줍니다. 이 예제는 기본 값을 사용합니다.

항목 데이터 세트가 있고 추천을 받을 때 메타데이터를 포함하는 옵션을 원하는 경우 recommender-config를 업데이트하여 enableMetadataWithRecommendations 필드를 추가하고 true로 설정하세요.

```
aws personalize create-recommender \
--name recommender name \
--dataset-group-arn dataset group ARN \
--recipe-arn arn:aws:personalize:::recipe/aws-vod-top-picks \
--recommender-config "{\"itemExplorationConfig\":{\"explorationWeight\":\"0.3\",
\"explorationItemAgeCutOff\":\"30\"}}"
```

추천에서 메타데이터 활성화

항목 데이터 세트가 있고 추천을 받을 때 메타데이터를 포함하는 옵션을 원하는 경우 recommender-config에서 enableMetadataWithRecommendations를 true로 설정합니다.

```
aws personalize create-recommender \
--name recommender name \
--dataset-group-arn dataset group \
--recipe-arn recipe ARN \
--recommender-config "{\"enableMetadataWithRecommendations\": \"true\"}"
```

추천인 (SDK)AWS 만들기

도메인 데이터세트 그룹을 생성하고 데이터를 가져오고 나면, 도메인 사용 사례에 맞는 추천을 생성할 수 있습니다. 추천은 추천을 생성하는 도메인 데이터세트 그룹 리소스입니다.

모든 사용 사례에 대해 교육 시 사용되는 열을 구성할 수 있습니다. 자세한 내용은 [교육 시 사용되는 열 구성](#) 단원을 참조하세요. Top picks for your 또는 Recommended for you 사용 사례에 대해, Personalize는 항목을 추천할 때 탐색을 사용합니다. 자세한 내용은 [탐색 구성](#) 단원을 참조하세요.

주제

- [추천 생성](#)
- [탐색 구성](#)
- [교육 시 사용되는 열 구성](#)
- [메타데이터 활성화](#)

추천 생성

다음 코드로 도메인 사용 사례에 맞는 추천을 생성합니다. 추천에 이름을 지정하고 도메인 데이터세트 그룹의 리소스 이름(ARN)을 제공합니다. `recipeArn`의 경우, 사용 사례에 맞는 ARN을 제공합니다. 각 도메인 사용 사례에 대해 이 코드를 실행합니다. 사용 가능한 사용 사례는 도메인에 따라 다릅니다. 사용 사례, ARN 및 요구 사항 목록은 [사용 사례 선택](#) 단원을 참조하세요.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

create_recommender_response = personalize.create_recommender(
    name = 'recommender name',
    recipeArn = 'recipe ARN',
    datasetGroupArn = 'dataset group ARN'
)

recommender_arn = create_recommender_response['recommenderArn']

print('Recommender ARN:' + recommender_arn)
```

SDK for Java 2.x

```
public static String createRecommender(PersonalizeClient personalizeClient,
    String name,
    String datasetGroupArn,
    String recipeArn) {

    long maxTime = 0;
    long waitInMilliseconds = 30 * 1000; // 30 seconds
    String recommenderStatus = "";

    try {
        CreateRecommenderRequest createRecommenderRequest =
        CreateRecommenderRequest.builder()
            .datasetGroupArn(datasetGroupArn)
            .name(name)
            .recipeArn(recipeArn)
            .build();

        CreateRecommenderResponse recommenderResponse = personalizeClient
```

```

        .createRecommender(createRecommenderRequest);
    String recommenderArn = recommenderResponse.recommenderArn();
    System.out.println("The recommender ARN is " + recommenderArn);

    DescribeRecommenderRequest describeRecommenderRequest =
DescribeRecommenderRequest.builder()
        .recommenderArn(recommenderArn)
        .build();

    maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

    while (Instant.now().getEpochSecond() < maxTime) {

        recommenderStatus =
personalizeClient.describeRecommender(describeRecommenderRequest).recommender()
            .status();
        System.out.println("Recommender status: " + recommenderStatus);

        if (recommenderStatus.equals("ACTIVE") ||
recommenderStatus.equals("CREATE FAILED")) {
            break;
        }
        try {
            Thread.sleep(waitInMilliseconds);
        } catch (InterruptedException e) {
            System.out.println(e.getMessage());
        }
    }
    return recommenderArn;

} catch (PersonalizeException e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
return "";
}
}

```

SDK for JavaScript v3

```

// Get service clients and commands using ES6 syntax.
import { CreateRecommenderCommand, PersonalizeClient } from
"@aws-sdk/client-personalize";

```

```

// create personalizeClient
const personalizeClient = new PersonalizeClient({
  region: "REGION"
});

// set the recommender's parameters
export const createRecommenderParam = {
  name: "RECOMMENDER_NAME",          /* required */
  recipeArn: "RECIPE_ARN",           /* required */
  datasetGroupArn: "DATASET_GROUP_ARN" /* required */
}

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
    CreateRecommenderCommand(createRecommenderParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();

```

탐색 구성

Top picks for your 또는 Recommended for you 사용 사례에 대해, Personalize는 항목을 추천할 때 탐색을 사용합니다. 탐색에는 상호작용 데이터가 거의 없는 항목에 대한 사용자의 반응을 알아보기 위해 다양한 항목 추천을 테스트하는 작업이 포함됩니다. 다음과 같이 탐색을 구성할 수 있습니다.

- 관련성이 낮은 항목 탐색에 주목(탐색 가중치) — 탐색 범위를 설정합니다. 0과 1 사이에서 소수 값을 지정합니다. 기본값은 0.3입니다. 값이 1에 가까울수록 탐색이 더 많아집니다. 더 많이 탐색할수록 항목 상호 작용 데이터가 적거나 이전 행동을 기반으로 한 관련성이 낮은 항목이 더 많이 추천에 포함됩니다. 0이면 탐색이 수행되지 않으며 최신 데이터(관련성)를 기반으로 추천이 제공됩니다.
- 탐색 항목 기간 제한 – 항목 상호 작용 데이터 세트의 모든 항목에서 최근 상호 작용이 발생한 이후 최대 항목 사용 기간(일)을 지정합니다. 이렇게 하면 항목 기간에 따른 항목 탐색 범위가 정의됩니다. Amazon Personalize는 생성 타임스탬프 또는 생성 타임스탬프 데이터가 누락된 경우 항목 상호 작용 데이터를 기반으로 항목 기간을 결정합니다. Personalize가 항목 기간을 결정하는 방법에 대한 자세한 내용은 [생성 타임스탬프 데이터](#) 단원을 참조하세요.

Personalize가 탐색 중에 고려하는 항목을 늘리려면 더 큰 값을 입력합니다. 최소값은 1일이고, 기본값은 30일입니다. 지정한 항목 보관 기간 한도보다 오래된 항목이 추천에 포함될 수 있습니다. 이는 이러한 항목이 사용자와 관련이 있고 탐색에서 해당 항목을 식별할 수 없었기 때문입니다.

다음 코드는 추천을 생성할 때 탐색을 구성하는 방법을 보여줍니다. 이 예제는 기본 값을 사용합니다.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

create_recommender_response = personalize.create_recommender(
    name = 'recommender name',
    recipeArn = 'arn:aws:personalize::recipe/aws-vod-top-picks',
    datasetGroupArn = 'dataset group ARN',
    recommenderConfig = {"itemExplorationConfig": {"explorationWeight": "0.3",
"explorationItemAgeCutOff": "30"}}
)

recommender_arn = create_recommender_response['recommenderArn']

print('Recommender ARN:' + recommender_arn)
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { CreateRecommenderCommand, PersonalizeClient } from
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
  region: "REGION"
});

// set the recommender's parameters
export const createRecommenderParam = {
  name: "RECOMMENDER_NAME",           /* required */
  recipeArn: "RECIPE_ARN",           /* required */
  datasetGroupArn: "DATASET_GROUP_ARN", /* required */
  recommenderConfig: {
```

```

    itemExplorationConfig: {
      explorationWeight: "0.3",
      explorationItemAgeCutOff: "30"
    }
  }
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
    CreateRecommenderCommand(createRecommenderParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();

```

교육 시 사용되는 열 구성

학습에서 열을 제외하려면 추천 구성의 일부로 `trainingDataConfig`의 `excludedDatasetColumns` 객체를 제공하세요. 각 키에 대해 데이터세트 유형을 제공하세요. 각 값에 대해 제외할 열 목록을 제공하세요. 다음 코드는 추천을 생성할 때 교육에서 열을 제외하는 방법을 보여줍니다. 자세한 정보는 [교육 시 사용되는 열 구성](#)을 참조하세요.

SDK for Python (Boto3)

```

import boto3

personalize = boto3.client('personalize')

create_recommender_response = personalize.create_recommender(
    name = 'recommender name',
    recipeArn = 'recipe name',
    datasetGroupArn = 'dataset group ARN',
    recommenderConfig = {
      "trainingDataConfig": {
        "excludedDatasetColumns": {
          "datasetType": ["COLUMN_A", "COLUMN_B"]
        }
      }
    }
)

```

```
    }  
  )  
  
  recommender_arn = create_recommender_response['recommenderArn']  
  
  print('Recommender ARN:' + recommender_arn)
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.  
import { CreateRecommenderCommand, PersonalizeClient } from  
  "@aws-sdk/client-personalize";  
  
// create personalizeClient  
const personalizeClient = new PersonalizeClient({  
  region: "REGION"  
});  
  
// set the recommender's parameters  
export const createRecommenderParam = {  
  name: "RECOMMENDER_NAME",          /* required */  
  recipeArn: "RECIPE_ARN",          /* required */  
  datasetGroupArn: "DATASET_GROUP_ARN", /* required */  
  recommenderConfig: {  
    trainingDataConfig: {  
      excludedDatasetColumns: {  
        "DATASET_TYPE": ["COLUMN_A", "COLUMN_B"]  
      }  
    }  
  }  
};  
  
export const run = async () => {  
  try {  
    const response = await personalizeClient.send(new  
      CreateRecommenderCommand(createRecommenderParam));  
    console.log("Success", response);  
    return response; // For unit tests.  
  } catch (err) {  
    console.log("Error", err);  
  }  
};  
run();
```

메타데이터 활성화

항목 데이터 세트가 있고 추천을 받을 때 메타데이터를 포함하는 옵션을 원하는 경우 `recommender-config`에서 `enableMetadataWithRecommendations`를 `true`로 설정합니다.

```
import boto3

personalize = boto3.client('personalize')

create_recommender_response = personalize.create_recommender(
    name = 'recommender name',
    recipeArn = 'recipe name',
    datasetGroupArn = 'dataset group ARN',
    recommenderConfig = {"enableMetadataWithRecommendations": True}
)

recommender_arn = create_recommender_response['recommenderArn']

print('Recommender ARN:' + recommender_arn)
```

추천자 평가

오프라인 및 온라인 지표를 통해 추천자의 성능을 평가할 수 있습니다. 온라인 지표는 실시간 추천과의 사용자 상호작용에서 보게 되는 경험적 결과입니다. 예를 들어 카탈로그를 탐색하는 사용자의 클릭률을 기록할 수 있습니다. 모든 온라인 지표를 생성하고 기록하는 것은 귀하의 책임입니다.

오프라인 지표는 추천자를 생성할 때 Amazon Personalize가 생성하는 지표입니다. 오프라인 지표를 사용하면 추천자 기본 모델의 성능을 평가할 수 있습니다. 오프라인 지표를 사용하면 모델을 동일한 데이터로 학습된 다른 모델과 비교할 수 있습니다. 이 단원의 나머지 부분에서 지표란 용어는 오프라인 지표를 가리킵니다.

성능 지표를 얻기 위해 Amazon Personalize는 입력 상호작용 데이터를 학습 세트와 테스트 세트로 분리합니다. 학습 세트는 90%의 사용자와 이들의 상호작용 데이터로 구성됩니다. 테스트 세트는 나머지 10%의 사용자와 이들의 상호작용 데이터로 구성됩니다.

그러면 Amazon Personalize는 학습 세트를 사용하여 추천자를 생성합니다. 학습이 완료되면 Amazon Personalize는 새 추천자에 테스트 세트의 각 사용자 데이터 중 가장 오래된 90%를 입력으로 제공합니다. 그런 다음 Amazon Personalize는 해당 추천자가 생성한 추천을 테스트 세트에 있는 각 사용자 데이터의 최신 10%에서 얻은 실제 상호작용과 비교하여 지표를 계산합니다.

주제

- [지표 검색](#)
- [지표 정의](#)
- [예제](#)
- [추가 리소스](#)

지표 검색

추천자가 활성화되면 Amazon Personalize 콘솔에서 해당 추천자에 대한 지표를 보거나 [DescribeRecommender](#) 작업을 호출하여 지표를 검색할 수 있습니다.

주제

- [지표 보기\(콘솔\)](#)
- [지표 검색\(AWS CLI\)](#)
- [지표 검색\(AWS SDK\)](#)

지표 보기(콘솔)

콘솔에서 추천자 지표를 보려면 해당 추천자의 세부 정보 페이지로 이동합니다.

1. <https://console.aws.amazon.com/personalize/home> 에서 Amazon Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터 세트 그룹 페이지에서 도메인 데이터 세트 그룹을 선택합니다.
3. 탐색 창에서 추천자를 선택합니다.
4. 추천자 목록에서 해당 지표를 확인할 추천자를 선택합니다.

지표 검색(AWS CLI)

다음 코드는 AWS CLI를 사용하여 추천자의 지표를 가져오는 방법을 보여줍니다.

```
aws personalize describe-recommender \
--recommender-arn recommender arn
```

다음은 VIDEO_ON_DEMAND 도메인의 가장 적합한 추천 제품 사용 사례에 대해 생성된 추천자의 지표 출력 예제입니다.

```
{
```

```

    "recommender": {
      "recommenderArn": "arn:aws:personalize:region:acct-id:recommender/
recommenderName",
      "datasetGroupArn": "arn:aws:personalize:region:acct-id:dataset-group/
dsGroupName",
      "name": "name123",
      "recipeArn": "arn:aws:personalize:::recipe/aws-vod-top-picks",
      "modelMetrics": {
        "coverage": 0.27,
        "mean_reciprocal_rank_at_25": 0.0379,
        "normalized_discounted_cumulative_gain_at_5": 0.0405,
        "normalized_discounted_cumulative_gain_at_10": 0.0513,
        "normalized_discounted_cumulative_gain_at_25": 0.0828,
        "precision_at_5": 0.0136,
        "precision_at_10": 0.0102,
        "precision_at_25": 0.0091,
      }
      "recommenderConfig": {},
      "creationDateTime": "2022-05-06T10:11:24.589000-07:00",
      "lastUpdatedDateTime": "2022-05-06T10:34:33.270000-07:00",
      "status": "ACTIVE",
    }
  }
}

```

지표 검색(AWS SDK)

다음 코드는 Python용 SDK(Boto3)를 사용하여 추천자에 대한 지표를 가져오는 방법을 보여줍니다.

```

import boto3

personalize = boto3.client('personalize')

response = personalize.describe_recommender(
    recommenderArn = 'recommender_arn'
)
print(response['recommender']['modelMetrics'])

```

다음은 VIDEO_ON_DEMAND 도메인의 가장 적합한 추천 제품 사용 사례에 대해 생성된 추천자의 지표 출력 예제입니다.

```

{
  "recommender": {

```

```

    "recommenderArn": "arn:aws:personalize:region:acct-id:recommender/
recommenderName",
    "datasetGroupArn": "arn:aws:personalize:region:acct-id:dataset-group/
dsGroupName",
    "name": "name123",
    "recipeArn": "arn:aws:personalize:::recipe/aws-vod-top-picks",
    "modelMetrics": {
        "coverage": 0.27,
        "mean_reciprocal_rank_at_25": 0.0379,
        "normalized_discounted_cumulative_gain_at_5": 0.0405,
        "normalized_discounted_cumulative_gain_at_10": 0.0513,
        "normalized_discounted_cumulative_gain_at_25": 0.0828,
        "precision_at_5": 0.0136,
        "precision_at_10": 0.0102,
        "precision_at_25": 0.0091,
    }
    "recommenderConfig": {},
    "creationDateTime": "2022-05-06T10:11:24.589000-07:00",
    "lastUpdatedDateTime": "2022-05-06T10:34:33.270000-07:00",
    "status": "ACTIVE",
}
}
}

```

지표 정의

Amazon Personalize가 추천자에 대해 생성하는 지표는 다음과 같은 용어를 사용하여 아래에 설명되어 있습니다.

- 관련 추천은 사용자가 실제로 상호작용한 항목에 대한 추천입니다. 이러한 항목은 테스트 세트의 각 사용자 상호작용 데이터 중 최신 10%에서 가져온 것입니다.
- 순위는 추천 목록 내에서 추천 항목의 위치를 말합니다. 위치 1(목록의 맨 위)이 사용자와 가장 관련 있는 것으로 간주됩니다.

각 지표에 대해 숫자가 높을수록(1에 가까울수록) 더 좋습니다. 더 자세히 알아보려면 [추가 리소스](#)에 나열되어 있는 리소스를 참조하세요.

비율

비율 값은 상호작용 및 항목 데이터 세트의 고유 항목 총 수 중에서 Amazon Personalize가 추천할 수 있는 고유 항목의 비율을 나타냅니다. 비율 점수가 높을수록 Amazon Personalize는 여러 사용자에게 동일한 몇 가지 항목을 반복해서 추천하는 대신 해당 항목을 더 많이 추천합니다. 가장 적합한

추천 제품(VIDEO_ON_DEMAND) 및 추천 제품(ECOMMERCE) 등 특성 항목 탐색이 포함된 사용 사례가 그렇지 않은 경우보다 비율이 더 높습니다.

평균 역수 순위 25

이 지표는 관련 추천을 상위 순위에서 생성하는 모델의 능력을 보여줍니다. 사용자에게 관련성이 높은 검색 결과를 생성하고 있지만 사용자가 목록에서 더 아래 쪽의 항목을 선택하지 않을 것으로 예상한다면 평균 역수 순위 25가 높은 모델을 선택할 수 있습니다. 예를 들어 사용자는 검색 결과에서 첫 번째 요리 레시피를 선택하는 경우가 많습니다.

Amazon Personalize는 추천 요청에 대해 평균 역수 순위 점수를 사용하여 이 지표를 계산합니다. 각 역수 순위 점수는 $1 / \text{the rank of the highest item interacted with by the user}$ 로 계산되는데, 여기서 가능한 총 순위는 25개입니다. 해당 사용자가 상호작용하는 더 낮은 순위의 다른 항목은 무시됩니다. 사용자가 첫 번째 항목을 선택한 경우 점수는 1입니다. 항목을 선택하지 않은 경우 점수는 0입니다.

예를 들어 세 명의 사용자에게 각각 25개의 추천을 표시할 수 있습니다.

- 사용자 1이 4 순위의 항목과 10 순위의 항목을 클릭한 경우 역수 순위 점수는 1/4이 됩니다.
- 사용자 2가 2 순위의 항목, 4 순위의 항목, 12 순위의 항목을 클릭한 경우 역수 순위 점수는 1/20이 됩니다.
- 사용자 3이 6 순위의 단일 항목을 클릭한 경우 역수 순위 점수는 1/6이 됩니다.

모든 추천 요청의 평균 역수 순위(이 경우 3)는 $(1/4 + 1/2 + 1/6) / 3 = .3056$ 로 계산됩니다.

정규화된 할인 누적 이득(NDCG) K(5, 10 또는 25)

이 지표는 모델이 추천의 순위를 얼마나 잘 매겼는지 알려줍니다. 여기서 K는 5, 10 또는 25개의 추천으로 구성된 표본 크기입니다. 이 지표는 순위가 가장 높은 항목 이외의 추천 순위에 가장 관심이 있는 경우에 유용합니다(자세한 내용은 mean reciprocal rank at 25 단원 참조). 예를 들어 한 번에 최대 10가지의 영화를 캐루셀에 표시하는 애플리케이션이 있는 경우 NDCG at 10의 점수가 유용할 수 있습니다.

Amazon Personalize는 테스트 세트의 각 사용자에게 대한 순위 위치를 기반으로 추천에 가중치를 할당하여 NDCG를 계산합니다. 각 추천은 위치에 따라 계수만큼 할인됩니다(더 낮은 가중치가 부여됨). 최종 지표는 테스트 세트에 있는 모든 사용자의 평균입니다. 정규화된 할인 누적 이득 K는 목록에서 더 아래 쪽에 있는 추천이 목록에서 더 위 쪽에 있는 추천보다 관련성이 낮다고 가정합니다.

Amazon Personalize에서는 $1/\log(1 + \text{position})$ 의 가중 계수를 사용하며 목록의 맨 위가 위치 1입니다.

정밀도 K

이 지표는 K(5, 10 또는 25)개 추천의 표본 크기를 기반으로 했을 때 모델의 추천이 얼마나 관련성이 있는지를 알려줍니다.

Amazon Personalize는 테스트 세트의 각 사용자에게 대한 상위 K개 추천 중 관련성이 있는 추천 수를 K로 나눈 값을 기반으로 이 지표를 계산합니다. 여기서 K는 5, 10 또는 25입니다. 최종 지표는 테스트 세트에 속한 모든 사용자의 평균입니다.

예를 들어, 사용자에게 10개의 항목을 추천하는 데 이 사용자가 그 중 3개와 상호작용하는 경우 정밀도 K는 올바르게 예측된 항목 3개를 총 10개의 추천 항목으로 나눈 값 $3 / 10 = .30$ 입니다.

이 지표는 관련 항목의 정확한 추천을 보상합니다. 점수가 1에 가까울수록 모델의 정밀도가 높아집니다.

예제

다음은 특정 사용자를 위한 추천 목록을 생성하는 추천자의 간단한 예입니다. 두 번째 추천과 다섯 번째 추천은 이 사용자에게 대한 테스트 데이터의 레코드와 일치합니다. 이들 두 가지는 관련된 추천입니다. K를 5로 설정하면 사용자에게 대해 다음 지표가 생성됩니다.

reciprocal_rank

계산: $1/2$

결과: 0.5000

normalized_discounted_cumulative_gain_at_5

계산: $(1/\log(1 + 2) + 1/\log(1 + 5)) / (1/\log(1 + 1) + 1/\log(1 + 2))$

결과: 0.6241

precision_at_5

계산: $2/5$

결과: 0.4000

추가 리소스

추천자 시스템을 위한 다양한 유형의 지표를 자세히 알아보려면 다음 외부 리소스를 참조하세요.

- [MRR 대 MAP 대 NDCG: 순위 인식 평가 지표 및 사용 시기](#)
- [할인된 누적 이득: 알아야 할 순위 지표](#)
- [추천자 시스템의 k 재현율 및 정밀도](#)
- [추천자 시스템의 순위 평가 지표](#)

추천자 관리

추천자를 지원하는 모델을 사용자가 따로 관리할 것은 없습니다. Amazon Personalize가 7일마다 자동으로 재학습합니다. 이는 데이터 세트의 전체 데이터를 기반으로 완전히 새로운 모델을 생성하는 전체 재학습입니다. 가장 적합한 추천 제품 및 사용자를 위한 추천 제품의 경우 Amazon Personalize는 2시간마다 기존 모델을 업데이트하여 탐색을 통해 추천에 새 항목을 포함시킵니다. 자세한 설명은 [자동 업데이트](#) 섹션을 참조하세요.

추천자 관리에는 다음 작업이 포함됩니다.

- 추천자 중지 및 시작 - 활성 추천자에 대한 청구를 일시 중지하려면 추천자를 중지했다가 나중에 다시 시작할 수 있습니다. 자세한 설명은 [추천자 중지 및 시작](#) 섹션을 참조하세요.
- 추천자 구성 업데이트 — 추천자가 학습에서 사용하는 열을 업데이트하고 추천자의 요청 용량을 업데이트할 수 있습니다. 자세한 설명은 [추천자 업데이트](#) 섹션을 참조하세요.
- 추천자 삭제 - [DeleteRecommender](#) 작업을 사용하여 추천자를 삭제할 수 있습니다. 또는 Amazon Personalize 콘솔의 추천자 세부 정보 페이지에서 추천자를 삭제할 수도 있습니다.

주제

- [추천자 업데이트](#)
- [추천자 중지 및 시작](#)

추천자 업데이트

추천자를 만든 후 추천자의 구성을 업데이트할 수 있습니다.

- 추천자가 학습에서 사용하는 열을 업데이트할 수 있습니다. 학습에 사용된 열을 수정하면 Amazon Personalize가 추천자를 지원하는 모델의 전체 재학습을 자동으로 시작합니다. 업데이트가 완료 되더라도 추천자로부터 추천을 받을 수 있습니다. 추천자는 업데이트가 완료될 때까지 이전 구성을 사용합니다. 이 업데이트의 상태를 추적하려면 [DescribeRecommender](#) 작업에서 반환된 `latestRecommenderUpdate`을 사용합니다. 추천자를 만들 때 제공한 것과 동일한 열을 제공하면 업데이트가 발생하지 않습니다.

- 추천자의 초당 최소 추천 요청을 업데이트할 수 있습니다. 이는 Amazon Personalize에서 프로비저닝한 기본 추천 요청 처리량을 지정합니다. 값이 높으면 요금이 인상됩니다. 1부터 시작하는 것이 좋습니다. Amazon CloudWatch 지표를 사용하여 사용량을 추적하고 필요에 따라 사용량을 늘립니다. 자세한 내용은 [초당 최소 추천 요청 수 및 Auto Scaling](#) 단원을 참조하세요.
- 가장 적합한 추천 제품 및 추천 제품 사용 사례의 경우 관련 항목 탐색에 중점을 두고 탐색 항목 기간 커트라인을 조정하여 탐색 구성을 업데이트할 수 있습니다. 탐색에 대한 자세한 내용은 [사용 사례 선택](#)의 사용 사례 단원을 참조하세요.

Amazon Personalize 콘솔, AWS Command Line Interface(AWS CLI) 또는 AWS SDK를 사용하여 추천자를 업데이트할 수 있습니다.

주제

- [추천자 업데이트\(Amazon Personalize 콘솔\)](#)
- [추천자 업데이트\(AWS CLI\)](#)
- [추천자 업데이트\(AWS SDK\)](#)

추천자 업데이트(Amazon Personalize 콘솔)

추천을 생성한 후에는 업데이트할 수 있습니다. 추천자가 학습에서 사용하는 열과 추천자의 초당 최소 추천 요청량을 업데이트할 수 있습니다. 가장 적합한 추천 제품과 추천 제품 사용 사례의 경우 탐색 구성을 업데이트할 수 있습니다. 콘솔을 사용하여 추천자를 업데이트하려면 다음과 같이 합니다.

추천자 구성을 업데이트하려면(콘솔)

1. <https://console.aws.amazon.com/personalize/home> 에서 Amazon Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터 세트 그룹 페이지에서 도메인 데이터 세트 그룹을 선택합니다.
3. 탐색 창에서 추천자를 선택합니다.
4. 추천자 페이지에서 업데이트하려는 추천자를 선택합니다.
5. 추천자 구성에서 편집을 선택합니다.
6. 추천자의 구성을 변경하고 업데이트를 선택합니다. 다른 구성 옵션에 대한 자세한 내용은 [추천 생성\(콘솔\)](#) 단원을 참조하세요.

추천자 업데이트(AWS CLI)

AWS CLI에서 추천자를 업데이트하려면 `update-recommender` 명령을 사용합니다. 추천자의 Amazon 리소스 이름(ARN)과 업데이트된 구성을 제공합니다. 다음 코드는 추천자가 학습에 사용하는 열을 업데이트하는 방법을 보여줍니다.

```
aws personalize update-recommender \
--dataset-group-arn dataset group ARN \
--recommender-config "{\"trainingDataConfig\": {\"excludedDatasetColumns\":
{ \"datasetType\" : [ \"column1Name\", \"column2Name\" ]}}}
```

학습에 사용된 열을 수정하면 Amazon Personalize이 추천자를 지원하는 모델의 전체 재학습을 자동으로 시작합니다. 업데이트가 완료되더라도 추천자로부터 추천을 받을 수 있습니다. 추천자는 업데이트가 완료될 때까지 이전 구성을 사용합니다. 이 업데이트의 상태를 추적하려면 [DescribeRecommender](#) 작업에서 반환된 `latestRecommenderUpdate`를 사용합니다.

변경할 수 있는 다른 옵션에 대한 자세한 내용은 [RecommenderConfig](#) 단원을 참조하세요.

추천자 업데이트(AWS SDK)

AWS에서 추천자를 업데이트하려면 [UpdateRecommender](#) 작업을 사용합니다. 추천자의 Amazon 리소스 이름(ARN)을 제공하고 새 구성을 지정합니다. 다음 코드는 추천자가 학습에 사용하는 열을 업데이트하는 방법을 보여줍니다.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

update_recommender_response = personalize.update_recommender(
    recommenderArn = 'dataset group ARN',
    recommenderConfig = {
        "trainingDataConfig": {
            "excludedDatasetColumns": {
                "datasetType": ["COLUMN_A", "COLUMN_B"]
            }
        }
    }
)
```


SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { UpdateRecommenderCommand, PersonalizeClient } from
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
  region: "REGION"
});

// set the request's parameters
export const updateRecommenderParam = {
  recommenderArn: "RECOMMENDER_ARN", /* required */
  recommenderConfig: {
    trainingDataConfig: {
      excludedDatasetColumns: {
        "DATASET_TYPE": ["COLUMN_A", "COLUMN_B"]
      }
    }
  }
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
    UpdateRecommenderCommand(updateRecommenderParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

`recommenderConfig`의 `excludedDatasetColumns`에서 학습에 사용된 열을 수정하면 Amazon Personalize이 추천자를 지원하는 모델의 전체 재학습을 자동으로 시작합니다. 업데이트가 완료 되더라도 추천자로부터 추천을 받을 수 있습니다. 추천자는 업데이트가 완료될 때까지 이전 구성을 사용합니다. 이 업데이트의 상태를 추적하려면 [DescribeRecommender](#) 작업에서 반환된 `latestRecommenderUpdate`을 사용합니다.

변경할 수 있는 다른 옵션에 대한 자세한 내용은 [RecommenderConfig](#) 단원을 참조하세요.

추천자 중지 및 시작

추천자를 활성화한 후에는 추천자를 중지하고 나중에 시작할 수 있습니다. 이렇게 하면 추천자 청구를 일시 중지시키고 사용할 때만 비용을 지불할 수 있습니다. 예를 들어 특정 요일에만 추천을 받아야 할 수 있습니다. 필요하지 않은 날에는 추천자를 중지하고 필요한 날에는 추천자를 시작할 수 있습니다.

추천자를 중지한 후에는 추천을 받는 데 사용할 수 없습니다. 추천자를 중지하면 추천자 청구 및 재학습이 중지됩니다. 하지만 추천자를 중지해도 추천자가 삭제되지는 않습니다. 언제든지 다시 시작하고 추천 받기를 재개할 수 있습니다. 추천자를 시작한다고 해서 데이터가 포함된 새 추천자가 만들어지지 않습니다. 대신 7일마다 추천자 청구 및 재학습을 재개합니다.

Amazon Personalize 콘솔 AWS Command Line Interface ,AWS CLI() AWS , SDK 또는 [StartRecommender](#) 및 API 작업을 사용하여 추천자를 중지하고 시작할 수 있습니다.

[StopRecommender](#)

추천자 상태

추천자를 중지하면 다음과 같은 순서로 추천자 상태가 ACTIVE에서 INACTIVE로 변경됩니다.

ACTIVE > STOP PENDING > STOP IN PROGRESS > INACTIVE

추천자를 시작하면 다음과 같은 순서로 추천자 상태가 INACTIVE에서 ACTIVE로 변경됩니다.

INACTIVE > START PENDING > START IN PROGRESS > ACTIVE

주제

- [추천자 중지 및 시작\(콘솔\)](#)
- [추천자 중지 및 재시작\(AWS CLI\)](#)
- [추천자 \(SDK\) 중지 및 재시작AWS](#)

추천자 중지 및 시작(콘솔)

Personalize를 사용하여 추천자를 중지하고 다시 시작할 수 있습니다.

주제

- [추천자 중지\(콘솔\)](#)
- [추천자 시작\(콘솔\)](#)

추천자 중지(콘솔)

Personalize 콘솔을 사용하여 다음과 같이 활성 추천자를 중지할 수 있습니다.

추천을 중지하려면

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터세트 그룹 페이지에서 도메인 데이터세트 그룹을 선택합니다.
3. 탐색 창에서 추천자를 선택합니다.
4. 추천자 페이지에서 중지하려는 추천자를 선택합니다.
5. 오른쪽 상단에서 추천자 중지를 선택하고 표시되는 창에서 확인합니다.

추천자 상태가 비활성 상태이면 추천자가 중지된 것입니다. 이렇게 하면 모든 추천자 청구 및 재학습이 중단됩니다. 시작할 때까지 추천자를 사용할 수 없습니다.

추천자 시작(콘솔)

Personalize 콘솔을 사용하여 다음과 같이 비활성 추천자를 시작할 수 있습니다.

추천자를 시작하려면

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터세트 그룹 페이지에서 도메인 데이터세트 그룹을 선택합니다.
3. 탐색 창에서 추천자를 선택합니다.
4. 추천자 페이지에서 시작하려는 추천자를 선택합니다.
5. 오른쪽 상단에서 추천자 시작을 선택하고 표시되는 창에서 추천자를 시작할지 확인합니다.

추천자 상태가 활성화되면 그로부터 추천을 다시 받을 수 있습니다. 추천자 청구 및 자동 재학습도 재개됩니다.

추천자 중지 및 재시작(AWS CLI)

를 사용하여 활성 추천자를 중지하려면 `stop-recommender` 명령을 사용하고 다음과 같이 추천자의 Amazon 리소스 이름 (ARN) 을 제공하십시오. AWS CLI

```
aws personalize stop-recommender --recommender-arn "recommender arn"
```

를 사용하여 비활성 추천자를 시작하려면 `start-recommender` 명령을 사용하고 중지된 추천자의 ARN을 다음과 같이 제공하십시오. AWS CLI

```
aws personalize start-recommender --recommender-arn "recommender arn"
```

[StartRecommender](#) API 작업에 대한 자세한 내용은 [StopRecommender](#) 단원을 참조하세요.

추천자 (SDK) 중지 및 재시작AWS

AWS SDK를 사용하여 활성 추천자를 시작하거나 비활성 추천자를 중지할 수 있습니다.

[StartRecommender](#) API 작업에 대한 자세한 내용은 [StopRecommender](#) 단원을 참조하세요.

주제

- [추천인 중지 \(SDK\)AWS](#)
- [추천자 \(SDK\) 시작AWS](#)

추천인 중지 (SDK)AWS

다음 코드는 SDK를 사용하여 활성 추천자를 중지하는 방법을 보여줍니다. AWS 중지하면 모든 추천자 청구 및 자동 재학습이 중지됩니다. 다시 시작할 때까지 추천자를 사용할 수 없습니다.

SDK for Python (Boto3)

Python용 SDK(Boto3)를 사용하여 활성 추천자를 중지하려면 `stop_recommender` 메서드를 사용하고 다음과 같이 추천자의 Amazon 리소스 이름(ARN)을 제공합니다.

```
import boto3
personalize = boto3.client('personalize')

stop_recommender_response = personalize.stop_recommender(
    recommenderArn = "recommenderARN"
)
print(stop_recommender_response)
```

SDK for Java 2.x

Java 2.x용 SDK를 사용하여 활성 추천자를 중지하려면 다음과 같이 `stopRecommender` 메서드를 사용하고 추천자의 ARN을 제공합니다.

```
public static void stopRecommender(PersonalizeClient personalizeClient,
                                   String datasetGroupArn) {

    try {

        StopRecommenderRequest stopRecommenderRequest =
        StopRecommenderRequest.builder()
            .recommenderArn(recommenderArn)
            .build();
        personalizeClient.stopRecommender(stopRecommenderRequest);
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { StopRecommenderCommand, PersonalizeClient } from
    "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
    region: "REGION"
});

// set the request params
export const stopRecommenderParam = {
    recommenderArn: "RECOMMENDER_ARN" /* required */
};

export const run = async () => {
    try {
        const response = await personalizeClient.send(
            new StopRecommenderCommand(stopRecommenderParam)
        );
        console.log("Success", response);
        return response; // For unit tests.
    } catch (err) {
        console.log("Error", err);
    }
};
```

```
run();
```

추천자 (SDK) 시작AWS

다음 코드는 SDK를 사용하여 비활성 추천자를 시작하는 방법을 보여줍니다. AWS 추천자 상태가 활성화되면 그로부터 추천을 다시 받을 수 있습니다. 동시에 추천자 청구 및 자동 재학습도 재개됩니다.

SDK for Python (Boto3)

Python용 SDK(Boto3)를 사용하여 비활성 추천자를 시작하려면 `start_recommender` 메서드를 사용하고 다음과 같이 추천자의 Amazon 리소스 이름(ARN)을 제공합니다.

```
import boto3
personalize = boto3.client('personalize')

start_recommender_response = personalize.start_recommender(
    recommenderArn = "recommenderARN"
)
print(start_recommender_response)
```

SDK for Java 2.x

Java 2.x용 SDK로 비활성 추천자를 시작하려면 다음과 같이 `startRecommender` 메서드를 사용하고 추천자용 ARN을 제공합니다.

```
public static void startRecommender(PersonalizeClient personalizeClient,
                                    String datasetGroupArn) {

    try {

        StartRecommenderRequest startRecommenderRequest =
        StartRecommenderRequest.builder()
            .recommenderArn(recommenderArn)
            .build();
        personalizeClient.startRecommender(startRecommenderRequest);
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { StartRecommenderCommand, PersonalizeClient } from
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
  region: "REGION"
});

// set the request params
export const startRecommenderParam = {
  recommenderArn: "RECOMMENDER_ARN" /* required */
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new StartRecommenderCommand(startRecommenderParam)
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

사용자 지정 리소스 생성

데이터를 가져오고 나면 추천을 받는 데 사용할 사용자 지정 리소스를 생성할 준비가 된 것입니다. 추천을 생성하는 사용자 지정 리소스를 생성하려면 다음과 같이 해야 합니다.

1. 솔루션 생성 및 구성: 모델이 특정 비즈니스 니즈를 충족하도록 솔루션 파라미터 및 레시피별 하이퍼파라미터를 사용자 지정합니다. 기본적으로 새 솔루션 버전은 자동 교육을 사용하여 구성 가능한 빈도로 솔루션 버전을 생성합니다. 기본 빈도는 7일마다입니다.
2. 솔루션 버전 생성 (모델 학습): 자동 학습을 사용하는 솔루션의 경우 솔루션이 활성화된 후 솔루션 버전 생성이 자동으로 시작됩니다. 수동 교육을 사용하는 솔루션의 경우 솔루션 버전을 수동으로 생성합니다. 솔루션 버전은 Personalize 추천 또는 사용자 세그먼트를 생성합니다.

3. 캠페인과 함께 솔루션 버전 배포(실시간 추천만 해당): 캠페인을 생성하여 솔루션 버전을 배포합니다. 실시간 추천을 요청할 때 캠페인을 사용합니다. 배치 추천을 받는 경우, 캠페인을 생성할 필요가 없습니다.

주제

- [솔루션 및 솔루션 버전 생성](#)
- [캠페인 생성](#)

솔루션 및 솔루션 버전 생성

데이터 가져오기를 마치면 솔루션을 생성할 준비가 된 것입니다. 솔루션이란 Personalize 레시피, 사용자 지정 파라미터 및 하나 이상의 솔루션 버전(학습된 모델)의 조합을 말합니다.

Personalize에서 솔루션을 생성하려면 다음과 같이 합니다.

1. 솔루션 생성 — 모델이 특정 비즈니스 요구 사항을 충족하도록 솔루션 파라미터 및 레시피별 하이퍼 파라미터를 사용자 지정합니다. [솔루션 생성 및 구성](#) 단원을 참조하세요. 사용 가능한 레시피 목록은 [레시피 선택](#) 단원을 참조하세요.

기본적으로 새 솔루션 버전은 자동 교육을 사용하여 7일마다 솔루션 버전을 생성합니다. 교육 빈도를 구성할 수 있습니다.

2. 솔루션 버전 생성 (모델 학습): 자동 학습을 사용하는 솔루션의 경우 솔루션이 활성화된 후 솔루션 버전 생성이 자동으로 시작됩니다. 여전히 솔루션 버전을 수동으로 생성할 수 있습니다. 자동 교육 기능이 꺼진 솔루션의 경우 솔루션 버전을 수동으로 생성합니다. [솔루션 버전 생성](#) 섹션을 참조하십시오.
3. 솔루션 버전 평가 — Personalize가 새 솔루션 버전에서 생성하는 지표를 사용하여 모델의 성능을 평가합니다. [지표를 사용하여 솔루션 버전 평가](#) 섹션을 참조하세요.

주제

- [솔루션 생성 및 구성](#)
- [솔루션 버전 생성](#)
- [지표를 사용하여 솔루션 버전 평가](#)

솔루션 생성 및 구성

데이터 가져오기를 마치면 솔루션을 생성할 준비가 된 것입니다. 솔루션이란 Amazon Personalize 레시피, 사용자 지정 교육 파라미터 및 하나 이상의 솔루션 버전의 조합을 말합니다. 솔루션 버전은 학습된 기계 학습 모델을 말합니다.

기본적으로 모든 새 솔루션은 자동 교육을 사용하여 7일마다 새 솔루션 버전을 생성합니다. 자동 훈련은 마지막 훈련 이후 대량 또는 실시간 상호 작용 데이터를 가져온 경우에만 발생합니다. 여기에는 항목 상호 작용이 포함되며, Next-Best-Action 레시피를 사용하는 솔루션의 경우 동작 상호 작용 데이터가 포함됩니다. 자동 훈련은 솔루션을 삭제할 때까지 계속됩니다. 자세한 정보는 [자동 교육 구성](#)을 참조하세요.

기존 솔루션이 있는 경우, Personalize 콘솔을 사용하여 솔루션을 복제할 수 있습니다. 솔루션을 복제할 때 기존 솔루션의 구성 (예: 레시피 및 하이퍼파라미터)을 시작점으로 사용하고 원하는 대로 변경할 수 있습니다. 자세한 정보는 [솔루션 복제\(콘솔\)](#)을 참조하세요.

콘솔, AWS Command Line Interface (AWS CLI) 또는 AWS SDK를 사용하여 솔루션을 만들고 구성할 수 있습니다. 솔루션을 생성한 후에는 Amazon Personalize 콘솔의 솔루션 세부 정보 페이지에서 또는 작업과 함께 구성 세부 정보를 볼 수 있습니다. [DescribeSolution](#)

솔루션을 생성한 후에는 구성을 변경할 수 없습니다. 기본적으로 모든 새 솔루션은 자동 교육을 사용합니다. 자동 교육을 사용하면 솔루션이 활성화되는 동안 교육 비용이 발생합니다. 불필요한 비용을 피하려면 완료 후 솔루션을 삭제해야 합니다. 교육 비용에 대한 자세한 내용은 [Amazon Personalize](#) 요금을 참조하십시오.

주제

- [솔루션 생성\(콘솔\)](#)
- [솔루션 생성\(AWS CLI\)](#)
- [솔루션 \(AWS SDK\) 만들기](#)
- [교육 구성](#)
- [솔루션 복제\(콘솔\)](#)

솔루션 생성(콘솔)

Important

솔루션을 만든 후에는 구성을 변경할 수 없습니다. 기본적으로 모든 새 솔루션은 자동 교육을 사용합니다. 자동 교육을 사용하면 솔루션이 활성화되는 동안 교육 비용이 발생합니다. 불

필요한 비용을 피하려면 완료 후 솔루션을 삭제해야 합니다. 교육 비용에 대한 자세한 내용은 [Amazon Personalize](#) 요금을 참조하십시오.

콘솔에서 솔루션을 생성하려면 데이터세트 그룹을 선택한 다음 솔루션 이름, 레시피 및 선택적 교육 구성을 지정합니다.

솔루션을 구성하려면(콘솔)

1. <https://console.aws.amazon.com/personalize/home> 에서 Amazon Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터세트 그룹 페이지에서 데이터세트 그룹을 선택합니다.
3. 개요 페이지의 3단계에서 다음 중 하나를 수행하십시오.
 - 도메인 데이터세트 그룹을 만든 경우 커스텀 리소스 사용을 선택하고 솔루션 생성을 선택합니다.
 - 사용자 지정 데이터 세트 그룹을 생성한 경우 솔루션 생성을 선택합니다.
4. 솔루션 이름의 경우 솔루션에 대한 이름을 지정합니다.
5. 솔루션 유형에서 만들려는 솔루션 유형을 선택합니다. 선택한 유형에 따라 사용 가능한 레시피가 달라집니다.
 - 항목 추천을 선택하여 사용자를 위한 항목 추천을 받습니다. 개인 맞춤형 영화 추천을 예로 들 수 있습니다.
 - 작업 추천을 선택하여 사용자를 위한 작업 추천을 받습니다. 예를 들어 앱 다운로드와 같이 사용자를 위한 차선책을 만들어 보세요.
 - 항목 데이터를 기반으로 사용자 세그먼트(사용자 그룹)를 가져오려면 사용자 세분화를 선택합니다.
6. 레시피의 경우 레시피를 선택합니다([레시피 선택](#) 참조).
7. 태그의 경우, 원하는 태그를 추가할 수 있습니다. Personalize 리소스 태그 지정에 대한 자세한 내용은 [Personalize 리소스에 태그 지정](#) 단원을 참조하세요.
8. 다음을 선택합니다.
9. 교육 구성 페이지에서 비즈니스 요구 사항에 맞게 솔루션을 사용자 지정하세요.
 - 자동 교육에서 솔루션이 자동 교육을 사용할지 여부를 선택합니다. 자동 학습을 사용하는 경우 변경할 수 있습니다 Automatic training frequency. 기본 훈련 빈도는 7일마다입니다.

자동 교육을 사용하는 것이 좋습니다. 이렇게 하면 권장 사항의 관련성을 더 쉽게 유지할 수 있습니다. 교육 빈도는 비즈니스 요구 사항, 사용하는 레시피, 데이터 가져오기 빈도에 따라 달라집니다. 자세한 정보는 [자동 교육 구성](#)을 참조하세요. 관련성 유지에 대한 자세한 내용은 [참조하십시오 추천 관련성 유지](#).

- 하이퍼파라미터 구성에서 레시피와 비즈니스 요구 사항에 따라 하이퍼파라미터 옵션을 구성하십시오. 레시피에 따라 다른 하이퍼파라미터를 사용합니다. 사용 가능한 하이퍼파라미터에 대한 자세한 내용은 [에서 개별 레시피를 참조하십시오. 레시피 선택](#)
 - 학습용 컬럼에서 레시피가 아이템 추천 또는 사용자 세그먼트를 생성하는 경우 Amazon Personalize가 솔루션 버전을 생성할 때 고려하는 열을 선택적으로 선택하십시오. 자세한 정보는 [교육 시 사용되는 열 구성](#)을 참조하세요.
 - 추가 구성에서 아이템 상호작용 데이터세트에 EVENT_TYPE이 있거나 EVENT_TYPE과 EVENT_VALUE 열이 모두 있는 경우 선택적으로 이벤트 유형 및 이벤트 값 임계값 필드를 사용하여 Amazon Personalize가 모델을 학습할 때 사용하는 아이템 상호 작용 데이터를 선택할 수 있습니다. 자세한 내용은 [교육에 사용할 항목 상호 작용 데이터 선택](#) 단원을 참조하세요.
 - [사용자-개인 맞춤 레시피](#) 또는 [개인 맞춤형-순위 레시피](#) 레시피를 사용하는 경우, 원하는 목표를 지정하고 목표 민감도를 선택하여 관련성 이외에도 목표에 맞추어 솔루션을 최적화합니다. 객관적 민감도는 Amazon Personalize가 상호 작용 데이터를 통한 관련성과 비교하여 목표에 따라 추천 항목의 균형을 맞추는 방법을 구성합니다. 자세한 정보는 [추가 목표에 맞게 솔루션 최적화](#)를 참조하세요.
10. 다음을 선택하고 솔루션 세부 정보를 검토하십시오. 솔루션을 만든 후에는 솔루션 구성을 변경할 수 없습니다.
11. 솔루션 생성을 선택합니다. 솔루션을 생성한 후 Amazon Personalize는 한 시간 내에 첫 번째 솔루션 버전 생성을 시작합니다. 교육이 시작되면 솔루션 세부 정보 페이지의 솔루션 버전 섹션에서 교육을 모니터링할 수 있습니다. 자동으로 생성된 솔루션 버전의 교육 유형은 AUTOMATIC입니다.

솔루션 버전이 활성 상태이면 해당 버전을 사용하여 권장 사항을 받을 수 있습니다. 액티브 솔루션 버전을 사용하는 방법은 권장 사항을 받는 방법에 따라 달라집니다.

- 실시간 추천을 받으려면 Amazon Personalize 캠페인과 함께 액티브 솔루션 버전을 배포하십시오. 캠페인을 사용하여 사용자를 위한 추천을 받을 수 있습니다. [캠페인 생성](#) 섹션을 참조하십시오.
- 일괄 추천의 경우 일괄 추천 작업 또는 배치 세그먼트 작업을 생성할 때 ACTIVE 솔루션 버전을 지정합니다. [배치 권장 사항 및 사용자 세그먼트\(사용자 지정 리소스\)](#) 섹션을 참조하십시오.

솔루션 생성(AWS CLI)

⚠ Important

솔루션을 만든 후에는 구성을 변경할 수 없습니다. 기본적으로 모든 새 솔루션은 자동 교육을 사용합니다. 자동 교육을 사용하면 솔루션이 활성화되는 동안 교육 비용이 발생합니다. 불필요한 비용을 피하려면 완료 후 솔루션을 삭제해야 합니다. 교육 비용에 대한 자세한 내용은 [Amazon Personalize 요금](#)을 참조하십시오.

를 사용하여 솔루션을 생성하려면 `create-solution` 명령을 사용하십시오. AWS CLI이 명령은 [CreateSolution](#) API 작업을 사용합니다. 다음 코드는 자동 학습을 사용하는 솔루션을 만드는 방법을 보여줍니다. 5일마다 새 솔루션 버전이 자동으로 생성됩니다.

코드를 사용하려면 솔루션에 이름을 지정하도록 업데이트하고, 데이터세트 그룹의 Amazon Resource Name (ARN) 을 지정하고, 선택적으로 훈련 빈도를 변경하고, 사용할 레시피의 ARN을 지정하십시오. 레시피에 대한 자세한 내용은 [레시피 선택](#) 단원을 참조하십시오.

```
aws personalize create-solution \
  --name solution name \
  --dataset-group-arn dataset group ARN \
  --recipe-arn recipe ARN \
  --perform-auto-training \
  --solution-config "{\"autoTrainingConfig\": {\"schedulingExpression\": \"rate(5 days)\"}\"}
```

- 자동 학습을 사용하는 것이 좋습니다. 이를 통해 권장 사항 관련성을 더 쉽게 유지하고 개선할 수 있습니다. 기본적으로 모든 새 솔루션은 자동 교육을 사용합니다. 기본 교육 빈도는 7일마다입니다. 교육 빈도는 비즈니스 요구 사항, 사용하는 레시피, 데이터 가져오기 빈도에 따라 달라집니다. 자세한 정보는 [자동 교육 구성](#)을 참조하세요.
- 레시피에 따라 코드를 수정하여 레시피별 속성 및 하이퍼파라미터를 구성하거나 (참조 [하이퍼파라미터 및 HPO](#)), 학습에 사용되는 열을 구성 (참조 [교육 시\(AWS CLI\) 사용되는 열 구성](#)) 하거나, 학습에 사용되는 항목 상호 작용 데이터를 필터링하도록 (참조 [교육에 사용할 항목 상호 작용 데이터 선택](#)) 할 수 있습니다.
- [사용자-개인 맞춤 레시피](#) 또는 [개인 맞춤형-순위 레시피](#) 레시피 중 하나를 사용하면 관련성 외에도 목표에 맞게 솔루션을 최적화할 수 있습니다. 자세한 정보는 [추가 목표에 맞게 솔루션 최적화](#)을 참조하세요.

솔루션을 생성한 후 나중에 사용할 수 있도록 솔루션 ARN을 기록해 두십시오. 자동 교육을 사용하면 솔루션이 활성화된 후 1년 이내에 솔루션 버전 생성이 시작됩니다. 한 시간 내에 솔루션 버전을 수동으로 만들면 솔루션은 첫 번째 자동 교육을 건너뛰게 됩니다. [교육이 시작되면 버전 API 작업을 통해 솔루션 버전의 Amazon 리소스 이름 \(ARN\) 을 ListSolution 가져올 수 있습니다.](#) 상태를 확인하려면 [DescribeSolution](#) 버전 API 작업을 사용하십시오.

솔루션 버전이 ACTIVE이면 해당 버전을 사용하여 권장 사항을 받을 수 있습니다. 액티브 솔루션 버전을 사용하는 방법은 권장 사항을 받는 방법에 따라 달라집니다.

- 실시간 추천을 받으려면 Amazon Personalize 캠페인과 함께 액티브 솔루션 버전을 배포하십시오. 캠페인을 사용하여 사용자를 위한 추천을 받을 수 있습니다. [캠페인 생성](#) 섹션을 참조하십시오.
- 일괄 추천의 경우 일괄 추론 작업 또는 배치 세그먼트 작업을 생성할 때 ACTIVE 솔루션 버전을 지정합니다. [배치 권장 사항 및 사용자 세그먼트\(사용자 지정 리소스\)](#) 섹션을 참조하십시오.

솔루션 (AWS SDK) 만들기

Important

솔루션을 만든 후에는 구성을 변경할 수 없습니다. 기본적으로 모든 새 솔루션은 자동 교육을 사용합니다. 자동 교육을 사용하면 솔루션이 활성화되는 동안 교육 비용이 발생합니다. 불필요한 비용을 피하려면 완료 후 솔루션을 삭제해야 합니다. 교육 비용에 대한 자세한 내용은 [Amazon Personalize](#) 요금을 참조하십시오.

AWS SDK로 솔루션을 생성하려면 [CreateSolution](#) API 작업을 사용하십시오. 다음 코드는 자동 학습을 사용하는 솔루션을 만드는 방법을 보여줍니다. 5일마다 새 솔루션 버전이 자동으로 생성됩니다.

코드를 사용하려면 솔루션에 이름을 지정하도록 업데이트하고, 데이터세트 그룹의 Amazon Resource Name (ARN) 을 지정하고, 선택적으로 훈련 빈도를 변경하고, 사용하려는 레시피의 ARN을 지정하십시오. 레시피에 대한 자세한 내용은 [레시피 선택](#) 단원을 참조하십시오.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

create_solution_response = personalize.create_solution(
    name = 'solution name',
```

```
recipeArn = 'recipe ARN',
datasetGroupArn = 'dataset group ARN',
performAutoTraining = True,
solutionConfig = {
  "autoTrainingConfig": {
    "schedulingExpression": "rate(5 days)"
  }
}
)
solution_arn = create_solution_response['solutionArn']
print('solution_arn: ', solution_arn)
```

SDK for JavaScript v3

```
import {
  CreateSolutionCommand,
  PersonalizeClient,
} from "@aws-sdk/client-personalize";

// create client
const personalizeClient = new PersonalizeClient({ region: "REGION" });

// set the solution parameters
export const solutionParam = {
  datasetGroupArn: "DATASET_GROUP_ARN" /* required */,
  recipeArn: "RECIPE_ARN" /* required */,
  name: "SOLUTION_NAME" /* required */,
  performAutoTraining: true /* optional, default is true */,
  solutionConfig: {
    autoTrainingConfig: {
      schedulingExpression:
        "rate(5 days)" /* optional, default is every 7 days */,
    },
  },
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateSolutionCommand(solutionParam)
    );
    console.log("Success", response);
  }
}
```

```

    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();

```

- 자동 학습을 사용하는 것이 좋습니다. 이를 통해 권장 사항 관련성을 더 쉽게 유지하고 개선할 수 있습니다. 기본적으로 모든 새 솔루션은 자동 교육을 사용합니다. 기본 교육 빈도는 7일마다입니다. 교육 빈도는 비즈니스 요구 사항, 사용하는 레시피, 데이터 가져오기 빈도에 따라 달라집니다. 자세한 정보는 [자동 교육 구성](#)을 참조하세요.
- 레시피에 따라 코드를 수정하여 레시피별 속성 및 하이퍼파라미터를 구성하거나 (참조 [하이퍼파라미터 및 HPO](#)), 학습에 사용되는 열을 구성 (참조 [교육 시 사용되는 열 구성 \(SDK\)AWS](#)) 하거나, 학습에 사용되는 항목 상호 작용 데이터를 필터링하도록 (참조 [교육에 사용할 항목 상호 작용 데이터 선택](#)) 할 수 있습니다.
- [사용자-개인 맞춤 레시피](#) 또는 [개인 맞춤형-순위 레시피](#) 레시피 중 하나를 사용하면 관련성 외에도 목표에 맞게 솔루션을 최적화할 수 있습니다. 자세한 정보는 [추가 목표에 맞게 솔루션 최적화](#)를 참조하세요.

솔루션을 생성한 후 나중에 사용할 수 있도록 솔루션 ARN을 기록해 두십시오. 자동 교육을 사용하면 솔루션이 활성화된 후 1년 이내에 솔루션 버전 생성이 시작됩니다. 한 시간 내에 솔루션 버전을 수동으로 만들면 솔루션은 첫 번째 자동 교육을 건너뛰게 됩니다. [교육이 시작되면 버전 API 작업을 통해 솔루션 버전의 Amazon 리소스 이름 \(ARN\)을 ListSolution 가져올 수 있습니다.](#) 상태를 확인하려면 [DescribeSolution버전](#) API 작업을 사용하십시오.

솔루션 버전이 ACTIVE이면 해당 버전을 사용하여 권장 사항을 받을 수 있습니다. 액티브 솔루션 버전을 사용하는 방법은 권장 사항을 받는 방법에 따라 달라집니다.

- 실시간 추천을 받으려면 Amazon Personalize 캠페인과 함께 액티브 솔루션 버전을 배포하십시오. 캠페인을 사용하여 사용자를 위한 추천을 받을 수 있습니다. [캠페인 생성](#) 섹션을 참조하십시오.
- 일괄 추천의 경우 일괄 추론 작업 또는 배치 세그먼트 작업을 생성할 때 ACTIVE 솔루션 버전을 지정합니다. [배치 권장 사항 및 사용자 세그먼트\(사용자 지정 리소스\)](#) 섹션을 참조하십시오.

교육 구성

솔루션을 만들 때 특정 비즈니스 요구 사항에 맞게 교육을 구성할 수 있습니다.

- 솔루션에서 자동 교육을 사용할지 여부를 구성할 수 있습니다. 그리고 훈련 빈도도 구성할 수 있습니다. 기본적으로 모든 솔루션은 자동 학습을 사용합니다. 자세한 정보는 [자동 교육 구성](#)을 참조하세요.
- 레시피가 아이템 추천 또는 사용자 세그먼트를 생성하는 경우 모델 교육 (솔루션 버전 생성) 시 Amazon Personalize가 고려하는 열을 수정할 수 있습니다. 자세한 정보는 [교육 시 사용되는 열 구성](#)을 참조하세요.
- 레시피와 비즈니스 니즈에 따라 모델을 최적화하도록 하이퍼파라미터를 구성할 수 있습니다. 레시피에 따라 다른 하이퍼파라미터를 사용합니다. 하이퍼파라미터 구성에 대한 자세한 내용은 을 참조하십시오. [하이퍼파라미터 및 HPO](#) 레시피에 사용할 수 있는 하이퍼파라미터는 [레시피 선택](#)의 레시피 페이지를 참조하세요.
- [사용자-개인 맞춤 레시피](#) 또는 [개인 맞춤형-순위 레시피](#) 레시피를 사용하면 관련성 외에도 목표에 맞추어 솔루션을 최적화할 수 있습니다. 자세한 내용은 [추가 목표에 맞게 솔루션 최적화](#) 단원을 참조하세요.
- 이벤트 유형 및 이벤트 값 데이터가 있는 경우, 이를 사용하여 Amazon Personalize가 교육 중에 고려하는 항목 상호 작용 레코드를 선택할 수 있습니다. 자세한 내용은 [교육에 사용할 항목 상호 작용 데이터 선택](#) 단원을 참조하세요.

주제

- [자동 교육 구성](#)
- [교육 시 사용되는 열 구성](#)
- [추가 목표에 맞게 솔루션 최적화](#)
- [하이퍼파라미터 및 HPO](#)
- [교육에 사용할 항목 상호 작용 데이터 선택](#)

자동 교육 구성

Important

솔루션을 만든 후에는 구성을 변경할 수 없습니다. 기본적으로 모든 새 솔루션은 자동 교육을 사용합니다. 자동 교육을 사용하면 솔루션이 활성화되는 동안 교육 비용이 발생합니다. 불필요한 비용을 피하려면 완료 후 솔루션을 삭제해야 합니다. 교육 비용에 대한 자세한 내용은 [Amazon Personalize](#) 요금을 참조하십시오.

솔루션을 생성할 때 솔루션이 자동 교육을 사용하는지 여부를 구성할 수 있습니다. 훈련 빈도도 구성할 수 있습니다. 예를 들어 5일마다 새 솔루션 버전을 생성하도록 솔루션을 구성할 수 있습니다.

기본적으로 모든 새 솔루션은 자동 교육을 사용하여 7일마다 새 솔루션 버전을 생성합니다. 자동 훈련은 마지막 훈련 이후 대량 또는 실시간 상호 작용 데이터를 가져온 경우에만 발생합니다. 여기에는 항목 상호 작용이 포함되며, Next-Best-Action 레시피를 사용하는 솔루션의 경우 동작 상호 작용 데이터가 포함됩니다. 자동 훈련은 솔루션을 삭제할 때까지 계속됩니다.

자동 교육을 사용하는 것이 좋습니다. 그러면 솔루션을 더 쉽게 유지 관리할 수 있습니다. 이를 통해 솔루션이 최신 데이터에서 학습하는 데 필요한 수동 교육이 필요하지 않습니다. 자동 교육을 사용하지 않으면 솔루션이 최신 데이터에서 학습할 수 있도록 새 솔루션 버전을 수동으로 만들어야 합니다. 이로 인해 추천이 유효하지 않고 전환율이 낮아질 수 있습니다. Amazon Personalize 권장 사항을 유지 관리하는 방법에 대한 자세한 내용은 [을 참조하십시오. 추천 관련성 유지](#)

Amazon Personalize 콘솔, AWS Command Line Interface (AWS CLI) 또는 AWS SDK를 사용하여 자동 교육을 구성할 수 있습니다. 콘솔을 사용한 자동 교육 구성 단계는 [을 참조하십시오. 솔루션 생성\(콘솔\)](#)

솔루션을 생성한 후 나중에 사용할 수 있도록 솔루션 ARN을 기록해 두십시오. 자동 교육을 사용하면 솔루션이 활성화된 후 1년 이내에 솔루션 버전 생성이 시작됩니다. 한 시간 내에 솔루션 버전을 수동으로 만들면 솔루션은 첫 번째 자동 교육을 건너뛰게 됩니다. [교육이 시작되면 버전 API 작업을 통해 솔루션 버전의 Amazon 리소스 이름 \(ARN\) 을 ListSolution 가져올 수 있습니다.](#) 상태를 확인하려면 [DescribeSolution버전 API](#) 작업을 사용하십시오.

솔루션 버전이 ACTIVE이면 해당 버전을 사용하여 권장 사항을 받을 수 있습니다. 액티브 솔루션 버전을 사용하는 방법은 권장 사항을 받는 방법에 따라 달라집니다.

- 실시간 추천을 받으려면 Amazon Personalize 캠페인과 함께 액티브 솔루션 버전을 배포하십시오. 캠페인을 사용하여 사용자를 위한 추천을 받을 수 있습니다. [캠페인 생성](#) 섹션을 참조하십시오.
- 일괄 추천의 경우 일괄 추론 작업 또는 배치 세그먼트 작업을 생성할 때 ACTIVE 솔루션 버전을 지정합니다. [배치 권장 사항 및 사용자 세그먼트\(사용자 지정 리소스\)](#) 섹션을 참조하십시오.

주제

- [지침 및 요구 사항](#)
- [자동 교육 구성 \(AWS CLI\)](#)
- [자동 교육 \(SDK\) 구성](#)

지침 및 요구 사항

다음은 자동 학습에 대한 지침 및 요구 사항입니다.

- 자동 훈련은 마지막 훈련 이후 대량 또는 실시간 상호 작용 데이터를 가져온 경우에만 발생합니다. 여기에는 항목 상호 작용이 포함되며, Next-Best-Action 레시피를 사용하는 솔루션의 경우 동작 상호 작용 데이터가 포함됩니다.
- 각 교육에서는 학습에 포함하는 데이터셋 그룹의 모든 데이터를 고려합니다. 학습에 사용되는 열을 구성하는 방법에 대한 자세한 내용은 [을 참조하십시오. 교육 시 사용되는 열 구성](#)
- 여전히 솔루션 버전을 수동으로 생성할 수 있습니다.
- 자동 교육은 솔루션이 활성화된 후 1시간 이내에 시작됩니다. 한 시간 내에 솔루션 버전을 수동으로 만들면 솔루션은 첫 번째 자동 교육을 건너뛰게 됩니다.
- 교육 일정은 교육 시작일을 기준으로 합니다. 예를 들어 첫 번째 솔루션 버전이 오후 7시에 교육을 시작하고 주간 교육을 사용하는 경우 다음 솔루션 버전은 일주일 후인 오후 7시에 교육을 시작합니다.
- 모든 레시피에는 최소 1주 동안 교육 빈도를 두는 것이 좋습니다. 훈련 빈도는 1일에서 30일 사이로 지정할 수 있습니다. 기본값은 7일마다입니다.
- 사용자 개인화-v2, 사용자 맞춤화 또는 Next-Best-Action을 사용하는 경우 솔루션이 자동으로 업데이트되어 권장 사항에 대한 새 항목이나 조치를 고려합니다. 자동 업데이트는 자동 학습과 다릅니다. 자동 업데이트는 완전히 새로운 솔루션 버전을 만들지 않으며, 모델은 최신 데이터를 통해 학습하지 않습니다. 솔루션을 유지 관리하려면 훈련 빈도가 여전히 1주 이상이어야 합니다. 추가 지침 및 요구 사항을 포함한 자동 업데이트에 대한 자세한 내용은 [을 참조하십시오. 자동 업데이트.](#)
- 지금-유행을 사용하는 경우 Personalize는 구성 가능한 시간 간격으로 상호작용 데이터에서 가장 인기 있는 항목을 자동으로 식별합니다. Trending-Now는 대량 또는 스트리밍 상호 작용 데이터를 통해 마지막 교육 이후 추가된 항목을 추천할 수 있습니다. 훈련 빈도는 여전히 1주 이상이어야 합니다. 자세한 정보는 [지금-유행 레시피](#)를 참조하세요.
- 자동 업데이트 기능이 있는 레시피 또는 Trending-Now 레시피를 사용하지 않는 경우 Amazon Personalize는 다음 교육 이후에만 새 항목을 추천 대상으로 간주합니다. 예를 들어, Similar-Items 레시피를 사용하고 매일 새 항목을 추가하는 경우 이러한 항목이 같은 날 권장 사항에 표시되려면 일일 자동 교육 빈도를 사용해야 합니다.

자동 교육 구성 (AWS CLI)

다음 코드는 5일마다 솔루션 버전을 자동으로 생성하는 솔루션을 만드는 방법을 보여줍니다. 자동 학습을 끄려면 `perform-auto-training` 로 설정하십시오 `false`.

훈련 빈도를 변경하려면 `schedulingExpression` 에서 을 수정할 수 `autoTrainingConfig` 있습니다. 표현식은 `rate(value unit)` 형식이 맞아야 합니다. 값에는 1에서 30 사이의 숫자를 지정하십시오. 단위의 경우 `day` 또는 을 지정합니다 `days`.

`create-solution` 명령에 대한 전체 설명은 을 참조하십시오 [솔루션 생성\(AWS CLI\)](#).

```
aws personalize create-solution \
--name solution name \
--dataset-group-arn dataset group ARN \
--recipe-arn recipe ARN \
--perform-auto-training \
--solution-config "{\"autoTrainingConfig\": {\"schedulingExpression\": \"rate(5 days)\\"}}"
```

자동 교육 (SDK) 구성

다음 코드는 AWS SDK를 사용하여 자동 학습이 포함된 솔루션을 만드는 방법을 보여줍니다. 솔루션은 5일마다 솔루션 버전을 자동으로 생성합니다. 자동 학습을 끄려면 `performAutoTraining` 로 설정합니다 `false`.

훈련 빈도를 변경하려면 `schedulingExpression` 에서 을 수정할 수 `autoTrainingConfig` 있습니다. 표현식은 `rate(value unit)` 형식이 맞아야 합니다. 값에는 1에서 30 사이의 숫자를 지정하십시오. 단위의 경우 `day` 또는 을 지정합니다 `days`.

`CreateSolution` API 작업에 대한 전체 설명은 을 참조하십시오 [솔루션 \(AWS SDK\) 만들기](#).

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

create_solution_response = personalize.create_solution(
    name = 'solution name',
    recipeArn = 'recipe ARN',
    datasetGroupArn = 'dataset group ARN',
    performAutoTraining = True,
    solutionConfig = {
        "autoTrainingConfig": {
            "schedulingExpression": "rate(5 days)"
        }
    }
)
```

```
    }  
  )  
  solution_arn = create_solution_response['solutionArn']  
  print('solution_arn: ', solution_arn)
```

SDK for JavaScript v3

```
import {  
  CreateSolutionCommand,  
  PersonalizeClient,  
} from "@aws-sdk/client-personalize";  
  
// create client  
const personalizeClient = new PersonalizeClient({ region: "REGION" });  
  
// set the solution parameters  
export const solutionParam = {  
  datasetGroupArn: "DATASET_GROUP_ARN" /* required */,  
  recipeArn: "RECIPE_ARN" /* required */,  
  name: "SOLUTION_NAME" /* required */,  
  performAutoTraining: true /* optional, default is true */,  
  solutionConfig: {  
    autoTrainingConfig: {  
      schedulingExpression:  
        "rate(5 days)" /* optional, default is every 7 days */,  
    },  
  },  
};  
  
export const run = async () => {  
  try {  
    const response = await personalizeClient.send(  
      new CreateSolutionCommand(solutionParam)  
    );  
    console.log("Success", response);  
    return response; // For unit tests.  
  } catch (err) {  
    console.log("Error", err);  
  }  
};  
run();
```

교육 시 사용되는 열 구성

Important

솔루션을 만든 후에는 구성을 변경할 수 없습니다. 기본적으로 모든 새 솔루션은 자동 교육을 사용합니다. 자동 교육을 사용하면 솔루션이 활성화되는 동안 교육 비용이 발생합니다. 불필요한 비용을 피하려면 완료 후 솔루션을 삭제해야 합니다. 교육 비용에 대한 자세한 내용은 [Amazon Personalize](#) 요금을 참조하십시오.

레시피가 항목 추천이나 사용자 세그먼트를 생성하는 경우 Amazon Personalize가 솔루션 버전을 생성(모델 교육)할 때 고려하는 열을 수정할 수 있습니다.

교육 시 사용되는 열을 변경하여 Personalize가 모델 교육(솔루션 버전 생성) 시 어떤 데이터를 사용할 것인지를 제어할 수 있습니다. 이를 통해 다양한 교육 데이터 조합을 실험해 볼 수 있습니다. 또는 중요한 데이터가 없는 열을 제외할 수도 있습니다. 예를 들어, 추천을 필터링하는 데만 사용하려는 열이 있을 수 있습니다. 이 열을 교육에서 제외할 수 있으며, Personalize는 필터링할 때만 이 열을 고려합니다.

EVENT_TYPE 열은 제외할 수 없습니다. 기본 설정으로 Personalize는 교육 시 사용할 수 있는 모든 열을 사용합니다. 다음 데이터는 항상 교육에서 제외됩니다.

- 부울 데이터 유형의 열
- [임프레션 데이터](#)
- 범주형 또는 텍스트형이 아닌 사용자 지정 문자열 필드

교육에 노출 데이터를 포함할 수는 없지만 사용 사례 또는 레시피에서 사용하는 경우 Amazon Personalize는 권장 사항을 받을 때 노출 데이터를 사용하여 탐색을 안내합니다.

솔루션을 이미 생성했고 교육 시 사용하는 열을 수정하려는 경우, 솔루션을 복제할 수 있습니다. 솔루션을 복제할 때 기존 솔루션의 구성(예: 레시피 및 하이퍼파라미터)을 시작점으로 사용하고 필요에 따라 변경할 수 있습니다. 자세한 정보는 [솔루션 복제\(콘솔\)](#)을 참조하세요.

Amazon Personalize 콘솔 AWS Command Line Interface (AWS CLI) 또는 SDK를 사용하여 학습할 때 Amazon Personalize가 사용하는 열을 구성할 수 있습니다. AWS Personalize 콘솔을 사용하여 열을 선택하는 방법에 대한 자세한 내용은 [솔루션 생성\(콘솔\)](#)의 고급 구성 단계를 참조하세요. 솔루션을 생성한 후에는 Personalize 콘솔의 솔루션 세부 정보 페이지에서 또는 [DescribeSolution](#)작업으로 솔루션에서 사용하는 열을 볼 수 있습니다.

주제

- [교육 시\(AWS CLI\) 사용되는 열 구성](#)
- [교육 시 사용되는 열 구성 \(SDK\)AWS](#)

교육 시(AWS CLI) 사용되는 열 구성

학습에서 열을 제외하려면 솔루션 구성의 일부로 `trainingDataConfig`의 `excludedDatasetColumns` 객체를 제공하세요. 각 키에 대해 데이터세트 유형을 제공하세요. 각 값에 대해 제외할 열 목록을 제공하세요. 다음 코드는 AWS CLI를 사용하여 솔루션을 만들 때 교육에서 열을 제외하는 방법을 보여줍니다.

```
aws personalize create-solution \
--name solution name \
--dataset-group-arn dataset group ARN \
--recipe-arn recipe ARN \
--solution-config "{\"trainingDataConfig\": {\"excludedDatasetColumns\": {
  { \"datasetType\" : [ \"column1Name\", \"column2Name\" ] } } }\"
```

교육 시 사용되는 열 구성 (SDK)AWS

학습에서 열을 제외하려면 솔루션 구성의 일부로 `trainingDataConfig`의 `excludedDatasetColumns` 객체를 제공하세요. 각 키에 대해 데이터세트 유형을 제공하세요. 각 값에 대해 제외할 열 목록을 제공하세요. 다음 코드는 SDK for Python(Boto3)를 사용하여 솔루션을 만들 때 교육에서 열을 제외하는 방법을 보여줍니다.

```
import boto3

personalize = boto3.client('personalize')

create_solution_response = personalize.create_solution(
    name = 'solution name',
    recipeArn = 'recipe ARN',
    datasetGroupArn = 'dataset group ARN',
    solutionConfig = {
        "trainingDataConfig": {
            "excludedDatasetColumns": {
                "datasetType": ["COLUMN_A", "COLUMN_B"]
            }
        }
    }
)

solution_arn = create_solution_response['solutionArn']
```

```
print('solution_arn: ', solution_arn)
```

추가 목표에 맞게 솔루션 최적화

Important

솔루션을 만든 후에는 구성을 변경할 수 없습니다. 기본적으로 모든 새 솔루션은 자동 교육을 사용합니다. 자동 교육을 사용하면 솔루션이 활성화되는 동안 교육 비용이 발생합니다. 불필요한 비용을 피하려면 완료 후 솔루션을 삭제해야 합니다. 교육 비용에 대한 자세한 내용은 [Amazon Personalize 요금을 참조하십시오](#).

사용자-개인 맞춤 레시피 또는 개인 맞춤형-순위 레시피를 사용하는 경우 교육 전에 수익 극대화처럼 관련성을 최대한 높이는 것 외에도 목표에 맞게 Amazon Personalize 솔루션을 최적화할 수 있습니다.

항목 추천 레시피를 사용하는 경우 Amazon Personalize의 주요 목표는 과거 및 실시간 항목 상호 작용 데이터를 기반으로 사용자에게 가장 관련성이 높은 항목을 예측하는 것입니다. 다음은 사용자가 가장 많이 상호작용할 가능성이 높은 항목입니다(예: 클릭할 가능성이 가장 높은 항목). 스트리밍 시간 최대화나 수익 증대와 같은 추가 목표가 있는 경우 관련성과 목표 두 가지 모두를 기반으로 추천을 생성하는 솔루션을 만들 수 있습니다.

추가 목표에 맞게 솔루션을 최적화하려면 사용자-개인 맞춤 레시피 또는 -순위 레시피로 새 솔루션을 생성하고 목표와 관련된 항목 데이터세트에서 숫자 메타데이터 열을 선택합니다. 추천을 생성할 때 Personalize는 이 데이터 열에 대해 값이 더 높은 항목에 더 많은 중요도를 부여합니다. 예를 들어 스트리밍 시간을 최대화하려면 VIDEO_LENGTH 열을 선택하고 수익을 극대화하기 위해 PRICE 열을 선택할 수 있습니다.

Amazon Personalize 콘솔, AWS Command Line Interface (AWS CLI) 또는 AWS SDK를 사용할 수 있습니다. Amazon Personalize 콘솔 사용에 대한 자세한 내용은 [을 참조하십시오](#). [솔루션 생성\(콘솔\)](#)

주제

- [지침 및 요구 사항](#)
- [목표 강조와 관련성의 균형 유지](#)
- [최적화 성능 측정](#)
- [솔루션 최적화\(AWS CLI\)](#)
- [솔루션 최적화 \(AWS SDK\)](#)
- [샘플 Jupyter Notebook](#)

지침 및 요구 사항

목표 요구 사항은 다음과 같습니다.

- 목표에 사용할 열을 하나만 선택할 수 있습니다.
- 스키마의 열은 숫자 유형을 가져야 합니다.
- 스키마의 열은 null유형을 가질 수 없습니다.

스키마와 데이터 유형에 대한 자세한 내용은 [스키마](#) 단원을 참조하세요.

목표 강조와 관련성의 균형 유지

관련성보다는 목표에 더 중점을 두고 항목을 추천하는 경우 절충점이 있을 수 있습니다. 예를 들어 추천을 통해 수익을 높이려는 경우 고가의 항목만 추천하면 항목이 사용자와 관련성이 떨어지고 사용자 참여도와 전환율이 낮아질 수 있습니다.

관련성과 목표 간의 균형을 맞추려면 솔루션을 만들 때 다음과 같은 목표 민감도 수준 중 하나를 선택 하십시오.

- **고기:** Amazon Personalize가 주로 항목 상호 작용 데이터를 사용하여 사용자와 가장 관련성이 높은 항목을 예측합니다.
- **낮음:** Personalize가 사용자의 목표에 덜 중점을 둡니다. 항목 상호 작용 데이터를 통한 관련성이 더 중요합니다.
- **보통:** Amazon Personalize가 항목 상호 작용 데이터를 통해 사용자의 목표와 관련성을 동등하게 강조합니다.
- **높음:** Personalize가 고객의 목표에 더 중점을 둡니다. 항목 상호 작용 데이터를 통한 관련성은 그다 지 중요하지 않습니다.

최적화 성능 측정

최적화 목표를 가진 솔루션에 대한 솔루션 버전을 생성(모델을 학습)하면 Personalize가 `average_rewards_at_k` 지표를 생성합니다. `average_rewards_at_k`의 점수는 목표 달성에 있어 솔루션 버전이 얼마나 잘 수행되는지를 나타냅니다. 이 지표를 계산하기 위해 Personalize는 다음과 같 이 각 사용자에게 대한 보상을 계산합니다.

```
rewards_per_user = total rewards from the user's interactions with their
top 25 reward generating recommendations / total rewards from the user's
interactions with recommendations
```


최종 `average_rewards_at_k`는 1보다 작거나 같고 0보다 큰 10진수 값으로 정규화된 모든 `rewards_per_user`의 평균입니다. 값이 1에 가까울수록 추천에서 기대할 수 있는 사용자당 평균 이득이 더 커집니다.

예를 들어 클릭으로 인한 수익을 극대화하는 것이 목표인 경우 Personalize는 가장 비용이 많이 드는 상위 25개 추천 항목 중에서 사용자가 클릭한 항목에서 발생한 총 수익을 사용자가 클릭한 모든 추천 항목의 수익으로 나누어 각 사용자 점수를 계산합니다. 그러면 Personalize는 모든 사용자 점수의 정규화된 평균을 반환합니다. `average_rewards_at_k`이 1에 가까울수록 추천을 통해 사용자당 얻을 수 있는 평균 수익이 더 커집니다.

지표 생성에 대한 자세한 내용은 [지표를 사용하여 솔루션 버전 평가](#) 단원을 참조하세요.

솔루션 최적화(AWS CLI)

사용자-개인 맞춤 또는 -순위 레시피를 통해서만 목표에 맞게 최적화할 수 있습니다. 를 사용하여 추가 목표에 맞게 솔루션을 최적화하려면 새 솔루션을 생성하고 `solutionConfig` 객체의 `optimizationObjective` 키를 사용하여 목표 세부 정보를 지정하십시오. AWS CLI `optimizationObjective`에는 다음 필드가 있습니다.

- `itemAttribute`: 목표와 관련된 항목 데이터세트의 수치 메타데이터 열의 이름을 지정합니다.
- `objectiveSensitivity`: 추천을 생성할 때 솔루션이 목표에 중점을 두는 정도를 지정합니다. 목표 민감도 수준은 Amazon Personalize가 항목 상호 작용 데이터 세트의 데이터를 통해 목표와 관련성을 기준으로 추천 항목의 균형을 맞추는 방법을 구성합니다. `objectiveSensitivity`은 OFF, 낮음, MEDIUM 또는 HIGH가 될 수 있습니다. 자세한 내용은 [목표 강조와 관련성의 균형 유지](#) 단원을 참조하세요.

다음은 `create-solution` AWS CLI 명령의 예입니다. `solution name`, `dataset group arn` 및 `recipe arn` 값을 사용자의 값으로 바꿉니다.

`optimizationObjective`의 경우, `COLUMN_NAME`를 해당 목표와 관련된 항목 데이터세트의 숫자 메타데이터 열 이름으로 바꿉니다. `objectiveSensitivity`의 경우 OFF, LOW, MEDIUM 또는 HIGH를 지정합니다.

```
aws personalize create-solution \
  --name solution name \
  --dataset-group-arn dataset group arn \
  --recipe-arn recipe arn \
  --solution-config "{\"optimizationObjective\":{\"itemAttribute\":\"COLUMN_NAME\",
  \"objectiveSensitivity\":\"MEDIUM\"}}"
```

솔루션이 준비되면 새 솔루션 버전을 생성합니다(예제 명령은 [솔루션 생성\(AWS CLI\)](#) 참조). 솔루션 버전을 만들었다면 솔루션 버전 지표로 최적화 성능을 볼 수 있습니다. [최적화 성능 측정](#) 섹션을 참조하십시오.

솔루션 최적화 (AWS SDK)

사용자-개인 맞춤 또는 -순위 레시피를 통해서만 목표에 맞게 최적화할 수 있습니다.

AWS SDK를 사용하여 추가 목표에 맞게 솔루션을 최적화하려면 새 솔루션을 만들고 솔루션 `solutionConfig` 개체의 `optimizationObjective` 키를 사용하여 목표 세부 정보를 지정하십시오. `optimizationObjective`에는 다음 필드가 있습니다.

- `itemAttribute`: 목표와 관련된 데이터세트 그룹의 항목 데이터세트의 수치 메타데이터 열의 이름을 지정합니다.
- `objectiveSensitivity`: 추천을 생성할 때 솔루션이 목표에 중점을 두는 정도를 지정합니다. 목표 민감도 수준은 Amazon Personalize가 항목 상호 작용 데이터 세트의 데이터를 통해 목표와 관련성을 기준으로 추천 항목의 균형을 맞추는 방법을 구성합니다. `objectiveSensitivity`은 OFF, LOW, MEDIUM 또는 HIGH일 수 있습니다. 자세한 정보는 [목표 강조와 관련성의 균형 유지](#)를 참조하십시오.

다음 코드를 사용하여 AWS SDK for Python (Boto3) 또는 AWS SDK for Java 2.x를 통해 추가 목표가 있는 솔루션을 생성합니다.

솔루션이 준비되면 새 솔루션 버전을 생성합니다(예: 코드는 [솔루션 버전 \(AWS SDK\) 생성](#) 참조). 솔루션 버전을 만들었다면 솔루션 버전 지표로 최적화 성능을 볼 수 있습니다. [최적화 성능 측정](#) 섹션을 참조하십시오.

SDK for Python (Boto3)

추가 목표에 최적화된 솔루션을 만들려면 다음 `create_solution` 메서드를 사용합니다. `solution name`, `dataset group arn` 및 `recipe arn` 값을 사용자의 값으로 바꿉니다.

`optimizationObjective`의 경우, `COLUMN_NAME`를 해당 목표와 관련된 항목 데이터세트의 숫자 메타데이터 열 이름으로 바꿉니다. `objectiveSensitivity`의 경우 OFF, LOW, MEDIUM 또는 HIGH를 지정합니다.

```
import boto3

personalize = boto3.client('personalize')
```

```

create_solution_response = personalize.create_solution(
    name= 'solution name',
    recipeArn = 'recipe arn',
    datasetGroupArn = 'dataset group arn',
    solutionConfig = {
        "optimizationObjective": {
            "itemAttribute": "COLUMN_NAME",
            "objectiveSensitivity": "MEDIUM"
        }
    }
)
solution_arn = create_solution_response['solutionArn']
print('solution_arn: ', solution_arn)

```

SDK for Java 2.x

추가 목표에 최적화된 솔루션을 생성하려면 `createPersonalizeSolution` 메서드를 사용하여 Personalize 서비스 클라이언트, 데이터세트 그룹의 Amazon 리소스 이름(ARN), 솔루션 이름, 레시피 ARN, 항목 속성, 목표 민감도 수준 등을 파라미터로 전달합니다.

```

public static String createPersonalizeSolution(PersonalizeClient personalizeClient,
                                             String datasetGroupArn,
                                             String solutionName,
                                             String recipeArn,
                                             String itemAttribute,
                                             String objectiveSensitivity) {

    try {
        OptimizationObjective optimizationObjective =
        OptimizationObjective.builder()
            .itemAttribute(itemAttribute)
            .objectiveSensitivity(objectiveSensitivity)
            .build();

        SolutionConfig solutionConfig = SolutionConfig.builder()
            .optimizationObjective(optimizationObjective)
            .build();

        CreateSolutionRequest solutionRequest = CreateSolutionRequest.builder()
            .name(solutionName)
            .datasetGroupArn(datasetGroupArn)
            .recipeArn(recipeArn)
            .solutionConfig(solutionConfig)

```

```

        .build();

        CreateSolutionResponse solutionResponse =
personalizeClient.createSolution(solutionRequest);

        return solutionResponse.solutionArn();

    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";

```

SDK for JavaScript v3

```

// Get service clients and commands using ES6 syntax.
import { CreateSolutionCommand, PersonalizeClient } from
"@aws-sdk/client-personalize";

// create the personalizeClient
const personalizeClient = new PersonalizeClient({ region: "REGION"});

// set the solution parameters.
export const createSolutionParam = {
    datasetGroupArn: 'DATASET_GROUP_ARN',           /* required */
    recipeArn: 'RECIPE_ARN',                       /* required */
    name: 'NAME',                                   /* required */
    solutionConfig: {
        optimizationObjective: {
            itemAttribute: "COLUMN_NAME",           /* specify the numerical column from
the Items dataset related to your objective */
            objectiveSensitivity: "MEDIUM"         /* specify OFF, LOW, MEDIUM, or HIGH
*/
        }
    }
};

export const run = async () => {
    try {
        const response = await personalizeClient.send(new
CreateSolutionCommand(createSolutionParam));
        console.log("Success", response);
        return response; // For unit tests.
    }

```

```

    } catch (err) {
      console.log("Error", err);
    }
  };
run();

```

샘플 Jupyter Notebook

[추가 목표 기반 항목 메타데이터에 최적화된 솔루션을 생성하는 방법을 보여주는 샘플 Jupyter 노트북](#)은 [Amazon Personalize 샘플 리포지토리의 objective_optimization 폴더를 참조하십시오](#). [GitHub](#)

하이퍼파라미터 및 HPO

⚠ Important

솔루션을 만든 후에는 구성을 변경할 수 없습니다. 기본적으로 모든 새 솔루션은 자동 교육을 사용합니다. 자동 교육을 사용하면 솔루션이 활성화되는 동안 교육 비용이 발생합니다. 불필요한 비용을 피하려면 완료 후 솔루션을 삭제해야 합니다. 교육 비용에 대한 자세한 내용은 [Amazon Personalize 요금을 참조하십시오](#).

교육 전에 하이퍼파라미터를 지정하여 학습된 모델을 특정 사용 사례에 맞추어 최적화합니다. 이 파라미터는 값이 교육 프로세스 중에 결정되는 모델 파라미터와 대조됩니다.

[CreateSolution](#) 작업에 전달되는 [SolutionConfig](#) 객체의 일부인 `algorithmHyperParameters` 키를 사용하여 하이퍼파라미터를 지정합니다.

`CreateSolution` 요청의 요약 버전은 다음과 같습니다. 예제에는 `solutionConfig` 객체가 포함되어 있습니다. `solutionConfig`를 사용하여 레시피의 기본 파라미터를 대체합니다.

```

{
  "name": "string",
  "recipeArn": "string",
  "eventType": "string",
  "solutionConfig": {
    "optimizationObjective": {
      "itemAttribute": "string",
      "objectiveSensitivity": "string"
    },
    "eventValueThreshold": "string",
    "featureTransformationParameters": {

```

```

        "string" : "string"
    },
    "algorithmHyperParameters": {
        "string" : "string"
    },
    "hpoConfig": {
        "algorithmHyperParameterRanges": {
            ...
        },
        "hpoResourceConfig": {
            "maxNumberOfTrainingJobs": "string",
            "maxParallelTrainingJobs": "string"
        }
    },
},
}

```

레시피에 따라 다른 하이퍼파라미터를 사용합니다. 사용 가능한 하이퍼파라미터는 [레시피 선택](#)의 개별 레시피를 참조하세요.

하이퍼파라미터 최적화 활성화

하이퍼파라미터 최적화(HPO) 또는 튜닝은 특정 학습 목표에 맞는 최적 하이퍼파라미터를 선택하는 작업입니다. 지정된 가능성 범위 내의 서로 다른 값으로 다양한 교육 작업을 실행하여 최적의 하이퍼파라미터를 결정합니다.

[사용자 개인화-v2 및 개인화 순위-v2를 사용하는 경우, 자동 교육을 켜면 Amazon Personalize는](#) 90일마다 HPO를 자동으로 수행합니다. 자동 교육이 없으면 HPO가 발생하지 않습니다. 다른 모든 레시피의 경우 HPO를 활성화해야 합니다. HPO를 사용하려면 `performHPO`를 `true`로 설정하고 `hpoConfig`객체를 포함합니다.

하이퍼파라미터에는 범주형, 연속 또는 정수 값이 지정될 수 있습니다. `hpoConfig` 객체에는 하이퍼파라미터와 범위를 지정하는 각 유형에 해당하는 키가 있습니다. 요청에 각 유형을 입력해야 하지만 레시피에 특정 유형의 파라미터가 없는 경우에는 비워 둘 수 있습니다. 예를 들어, 사용자-개인화에는 연속 유형의 조정 가능한 하이퍼파라미터가 없습니다. 따라서 `continuousHyperParameterRange`의 경우 빈 배열을 전달합니다.

다음 코드는 SDK for Python(Boto3)를 사용하여 활성화된 HPO로 솔루션을 생성하는 방법을 보여줍니다. 이 예제의 솔루션은 [사용자-개인 맞춤 레시피](#) 레시피를 사용하며 HPO를 `true`로 설정합니다. 코드는 `hidden_dimension`에 대한 값과 `categoricalHyperParameterRanges` 및 `integerHyperParameterRanges`을 제공합니다. `continuousHyperParameterRange`는 비어 있

고 `hpoResourceConfig`는 `maxNumberOfTrainingJobs` 및 `maxParallelTrainingJobs`을 설정합니다.

```
import boto3

personalize = boto3.client('personalize')

create_solution_response = personalize.create_solution(
    name = "solution name",
    datasetGroupArn = 'arn:aws:personalize:region:accountId:dataset-group/
datasetGroupName',
    recipeArn = 'arn:aws:personalize:::recipe/aws-user-personalization',
    performHPO = True,
    solutionConfig = {
        "algorithmHyperParameters": {
            "hidden_dimension": "55"
        },
        "hpoConfig": {
            "algorithmHyperParameterRanges": {
                "categoricalHyperParameterRanges": [
                    {
                        "name": "recency_mask",
                        "values": [ "true", "false" ]
                    }
                ],
                "integerHyperParameterRanges": [
                    {
                        "name": "bptt",
                        "minValue": 2,
                        "maxValue": 22
                    }
                ],
                "continuousHyperParameterRanges": [
                    ]
            },
            "hpoResourceConfig": {
                "maxNumberOfTrainingJobs": "4",
                "maxParallelTrainingJobs": "2"
            }
        }
    }
)
```

HPO에 대한 자세한 내용은 [자동 모델 튜닝](#)을 참조하세요.

하이퍼파라미터 보기

[DescribeSolution](#) 작업을 호출하여 솔루션의 하이퍼파라미터를 볼 수 있습니다. 다음 예제는 DescribeSolution출력을 보여줍니다. 솔루션 버전을 생성하고 나면(모델 교육), [DescribeSolutionVersion](#)작업과 함께 하이퍼파라미터를 볼 수도 있습니다.

```
{
  "solution": {
    "name": "hpo_config_solution",
    "solutionArn": "arn:aws:personalize:region:accountId:solution/solutionName",
    "performHPO": true,
    "performAutoML": false,
    "recipeArn": "arn:aws:personalize:::recipe/aws-user-personalization",
    "datasetGroupArn": "arn:aws:personalize:region:accountId:dataset-group/
datasetGroupName",
    "eventType": "click",
    "solutionConfig": {
      "hpoConfig": {
        "hpoResourceConfig": {
          "maxNumberOfTrainingJobs": "4",
          "maxParallelTrainingJobs": "2"
        },
        "algorithmHyperParameterRanges": {
          "integerHyperParameterRanges": [
            {
              "name": "training.bptt",
              "minValue": 2,
              "maxValue": 22
            }
          ],
          "continuousHyperParameterRanges": [],
          "categoricalHyperParameterRanges": [
            {
              "name": "data.recency_mask",
              "values": [
                "true",
                "false"
              ]
            }
          ]
        }
      }
    }
  }
}
```



```

    },
    "algorithmHyperParameters": {
      "hidden_dimension": "55"
    }
  },
  "status": "ACTIVE",
  "creationDateTime": "2022-07-08T12:12:48.565000-07:00",
  "lastUpdatedDateTime": "2022-07-08T12:12:48.565000-07:00"
}
}

```

교육에 사용할 항목 상호 작용 데이터 선택

Important

솔루션을 만든 후에는 구성을 변경할 수 없습니다. 기본적으로 모든 새 솔루션은 자동 교육을 사용합니다. 자동 교육을 사용하면 솔루션이 활성화되는 동안 교육 비용이 발생합니다. 불필요한 비용을 피하려면 완료 후 솔루션을 삭제해야 합니다. 교육 비용에 대한 자세한 내용은 [Amazon Personalize](#) 요금을 참조하십시오.

솔루션 버전을 생성(모델 교육)할 때 Amazon Personalize가 사용하는 항목 상호 작용 데이터 세트의 이벤트를 선택할 수 있습니다. 교육 전에 항목 상호 작용 데이터를 선택하면 관련 데이터 하위 집합만 교육에 사용하거나 노이즈를 제거하여 더 최적화된 모델을 교육할 수 있습니다. 항목 상호 작용 데이터 세트에 대한 자세한 내용은 [스키마 및 항목 상호 작용 데이터 세트](#) 섹션을 참조하세요.

Note

사용자 개인화-v2 또는 개인화 순위-v2를 사용하는 경우, 교육 비용은 이벤트 유형 또는 값을 기준으로 필터링하기 전에 항목 상호 작용 데이터를 기반으로 합니다. 요금에 대한 자세한 내용은 [Amazon Personalize](#) 요금을 참조하십시오.

항목 상호 작용 데이터는 다음과 같이 선택할 수 있습니다.

- 유형을 기준으로 레코드 선택 - 솔루션을 구성할 때 항목 상호 작용 데이터 세트의 EVENT_TYPE 열에 이벤트 유형이 포함되어 있는 경우 교육에 사용할 이벤트 유형을 필요에 따라 지정할 수 있습니다. 예를 들어, 항목 상호 작용 데이터 세트에 구매, 클릭 및 시청 이벤트 유형이 포함되어 있고 Amazon Personalize가 시청 이벤트만 사용하여 모델을 교육하도록 하려는 경우, 솔루션을 구성할 때 Amazon Personalize가 교육에 사용하는 event type과 동일한 시청을 제공하면 됩니다.

항목 상호 작용 데이터 세트의 EVENT_TYPE 열에 여러 이벤트 유형이 있고 솔루션을 구성할 때 이벤트 유형을 제공하지 않은 경우, Amazon Personalize는 유형에 관계없이 모든 항목 상호 작용 데이터를 동일한 가중치로 학습하는 데 사용합니다.

- 유형 및 값을 기준으로 레코드 선택 - 솔루션을 구성할 때 항목 상호 작용 데이터 세트에 EVENT_TYPE 및 EVENT_VALUE 필드가 포함되어 있는 경우 특정 값을 임계값으로 설정하여 교육에서 레코드를 제외할 수 있습니다. 예를 들어, 시청의 EVENT_TYPE이 포함된 이벤트의 EVENT_VALUE 데이터가 사용자가 시청한 동영상의 비율이고, 이벤트 값 임계값을 0.5로, 이벤트 유형을 시청으로 설정하면, Personalize는 EVENT_VALUE가 0.5 이상인 시청 상호작용 이벤트만 사용하여 모델을 교육합니다.

다음 코드는 Python용 SDK (Boto3) 를 사용하여 사용자가 동영상의 절반 이상을 시청한 이벤트만 watch 사용하는 솔루션을 만드는 방법을 보여줍니다.

```
import boto3

personalize = boto3.client('personalize')

create_solution_response = personalize.create_solution(
    name = 'solution name',
    datasetGroupArn = 'arn:aws:personalize:region:accountId:dataset-group/
datasetGroupName',
    recipeArn = 'arn:aws:personalize:::recipe/aws-user-personalization-v2',
    eventType = 'watch',
    solutionConfig = {
        "eventValueThreshold": "0.5"
    }
)

# Store the solution ARN
solution_arn = create_solution_response['solutionArn']

# Use the solution ARN to get the solution status
solution_description = personalize.describe_solution(solutionArn = solution_arn)
['solution']
print('Solution status: ' + solution_description['status'])
```

솔루션 복제(콘솔)

새 솔루션을 생성할 때 Amazon Personalize 콘솔을 사용하여 솔루션을 복제할 수 있습니다. 솔루션을 복제할 때 기존 솔루션의 구성(예: 레시피 및 하이퍼파라미터)을 시작점으로 사용하고 필요에 따라 변경할 수 있습니다. 이 방법은 솔루션을 한 번만 변경하되 다른 모든 속성은 변경하지 않으려는 경우에 유용합니다. 예를 들어, 데이터 세트에 교육 데이터의 새 열 추가하기. 이 경우, 솔루션을 복제하고 솔루션에 이름을 지정하고 교육 시 사용되는 열을 변경하고 다른 모든 속성은 변경하지 않고 그대로 둡니다.

솔루션 복제

솔루션을 복제하려면 기존 솔루션을 선택하고 솔루션 복제 옵션을 선택합니다. 그런 다음 새 솔루션에 이름을 지정하고 관련 필드를 수정합니다.

솔루션을 복제하려면

1. <https://console.aws.amazon.com/acm-pca/home>에서 Amazon Personalize 콘솔을 열고 계정에 로그인하세요.
2. 데이터 세트 그룹 페이지에서 데이터 세트 그룹을 선택합니다.
3. 사용자 지정 리소스를 선택하고 솔루션을 선택합니다.
4. 복제하려는 솔루션을 선택합니다.
5. 작업을 선택하고 솔루션 복제를 선택합니다.
6. 새 솔루션에 이름을 지정합니다.
7. 솔루션 세부 정보 및 고급 구성을 변경합니다. Amazon Personalize는 이러한 필드를 기존 솔루션의 값으로 미리 채웁니다. 각 필드에 대한 자세한 내용은 [솔루션 생성 및 구성](#) 단원을 참조하세요.

솔루션 버전 생성

교육을 [솔루션 생성 및 구성](#) 마치면 교육을 시작할 준비가 된 것입니다.

- 솔루션에서 자동 학습을 사용하는 경우 솔루션은 사용자가 지정한 훈련 빈도에 따라 솔루션 버전을 생성합니다. 기본적으로 모든 새 솔루션은 자동 교육을 사용하여 7일마다 새 솔루션 버전을 생성합니다. 여전히 솔루션 버전을 수동으로 만들 수 있습니다. 자세한 정보는 [자동 교육 구성](#)을 참조하세요.
- 솔루션에 대한 자동 교육을 끄거나 수동으로 학습시키려는 경우 솔루션 버전을 수동으로 만들 수 있습니다. 솔루션 버전은 학습된 기계 학습 모델을 말합니다. 콘솔, AWS Command Line

Interface (AWS CLI) 또는 AWS SDK를 사용하여 솔루션 버전을 생성할 수 있습니다. 솔루션 버전이 CREATE_PENDING 또는 CREATE_IN_PROGRESS 상태인 경우, [the section called “StopSolutionVersionCreation”](#) 작업을 사용하여 솔루션 버전 생성 프로세스를 중지할 수 있습니다. [솔루션 버전 생성 중지](#) 단원을 참조하세요.

주제

- [솔루션 버전 생성\(콘솔\)](#)
- [솔루션 버전 생성\(AWS CLI\)](#)
- [솔루션 버전 \(AWS SDK\) 생성](#)
- [솔루션 버전 생성 중지](#)

솔루션 버전 생성(콘솔)

Personalize 콘솔로 솔루션을 처음 생성할 때는 솔루션 버전도 생성합니다. 솔루션 세부 정보 페이지의 솔루션 버전 단원에서 교육 진행 상황을 추적할 수 있습니다. 교육이 완료되면 상태가 활성으로 표시되고 준비가 된 것입니다. [캠페인 생성](#) 또는 [배치 권장 사항 및 사용자 세그먼트\(사용자 지정 리소스\)](#)을 참조하세요.

기존 솔루션을 위한 추가 솔루션 버전을 생성하려면 다음과 같이 솔루션 개요 페이지에서 새 솔루션 버전을 생성합니다.

솔루션 버전을 생성하려면

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터세트 그룹 페이지로 이동하여 새 솔루션이 포함된 데이터세트 그룹을 선택합니다.
3. 탐색 창의 사용자 지정 리소스에서 솔루션 및 레시피를 선택합니다.
4. 솔루션 및 레시피 페이지에서 솔루션 버전을 생성하려는 솔루션을 선택합니다.
5. 솔루션 개요 페이지에서 솔루션 버전 생성을 선택하여 새 모델 학습을 시작합니다.

솔루션 세부 정보 페이지의 솔루션 버전 단원에서 교육 진행 상황을 추적할 수 있습니다. 교육이 완료되면 상태가 활성으로 표시되며, Personalize에서 제공하는 지표를 사용하여 교육을 평가할 수 있습니다. 자세한 내용은 [지표를 사용하여 솔루션 버전 평가](#) 단원을 참조하세요.

오류로 인해 교육이 완료되지 않은 경우, 교육 비용은 청구되지 않습니다. 솔루션 버전이 CREATE_PENDING 또는 CREATE_IN_PROGRESS 상태인 경우, 솔루션 버전 생성 프로세스를 중지

할 수 있습니다. 솔루션 버전 생성을 중지하려면 솔루션 버전 세부 정보 페이지로 이동한 다음 중지를 선택합니다. [솔루션 버전 생성 중지](#) 섹션을 참조하십시오.

액티브 솔루션 버전을 사용하는 방법은 권장 사항을 받는 방법에 따라 달라집니다.

- 실시간 추천을 받으려면 Amazon Personalize 캠페인과 함께 액티브 솔루션 버전을 배포하십시오. 캠페인을 사용하여 사용자를 위한 추천을 받을 수 있습니다. [캠페인 생성](#) 섹션을 참조하십시오.
- 일괄 추천의 경우 일괄 추론 작업 또는 배치 세그먼트 작업을 생성할 때 ACTIVE 솔루션 버전을 지정합니다. [배치 권장 사항 및 사용자 세그먼트\(사용자 지정 리소스\)](#) 섹션을 참조하십시오.

솔루션 버전 생성(AWS CLI)

솔루션이 ACTIVE인 경우, 다음 명령을 실행하여 모델을 교육시킵니다. solution arn을 [솔루션 생성 및 구성](#)에서 제공하는 솔루션 리소스 이름(ARN)으로 대체하세요.

```
aws personalize create-solution-version \
  --solution-arn solution arn
```

솔루션 버전 ARN이 예를 들어, 다음과 같이 표시됩니다.

```
{
  "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/SolutionName/
<version-id>"
}
```

describe-solution-version 명령을 사용하여 솔루션 버전의 교육 상태를 확인합니다. 이전 단계에서 반환된 솔루션 버전 ARN을 입력합니다. API에 대한 자세한 내용은 [DescribeSolutionVersion](#) 단원을 참조하세요.

```
aws personalize describe-solution-version \
  --solution-version-arn solution version arn
```

솔루션 버전의 속성과 학습 status가 표시됩니다. 처음에 상태는 CREATE PENDING으로 표시됩니다. 예를 들면 다음과 같이 표시됩니다.

```
{
  "solutionVersion": {
    "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/
solutionName/<version-id>",
```

```

    ...,
    "status": "CREATE PENDING"
  }
}

```

status이 ACTIVE이면 교육이 완료된 것이므로, Personalize에서 제공하는 지표를 사용하여 교육을 평가할 수 있습니다. 자세한 내용은 [지표를 사용하여 솔루션 버전 평가](#) 단원을 참조하세요. 오류로 인해 교육이 완료되지 않은 경우, 교육 비용은 청구되지 않습니다.

솔루션 버전이 CREATE_PENDING 또는 CREATE_IN_PROGRESS 상태인 경우, [StopSolutionVersionCreation](#) 작업을 사용하여 솔루션 버전 생성 프로세스를 중지할 수 있습니다. [솔루션 버전 생성 중지](#) 단원을 참조하세요.

액티브 솔루션 버전을 사용하는 방법은 권장 사항을 받는 방법에 따라 달라집니다.

- 실시간 추천을 받으려면 Amazon Personalize 캠페인과 함께 액티브 솔루션 버전을 배포하십시오. 캠페인을 사용하여 사용자를 위한 추천을 받을 수 있습니다. [캠페인 생성](#) 섹션을 참조하십시오.
- 일괄 추천의 경우 일괄 추천 작업 또는 배치 세그먼트 작업을 생성할 때 ACTIVE 솔루션 버전을 지정합니다. [배치 권장 사항 및 사용자 세그먼트\(사용자 지정 리소스\)](#) 섹션을 참조하십시오.

솔루션 버전 (AWS SDK) 생성

솔루션이 활성 상태이면 다음 코드를 사용하여 솔루션 버전을 생성하세요. [솔루션 생성 및 구성](#)에서 리소스 이름(ARN)을 지정합니다. [DescribeSolutionVersion](#) 작업을 사용하여 솔루션 버전의 상태를 검색합니다.

SDK for Python (Boto3)

```

import boto3

personalize = boto3.client('personalize')
# Store the solution ARN
solution_arn = 'solution_arn'

# Use the solution ARN to get the solution status.
solution_description = personalize.describe_solution(solutionArn = 'solution_arn')
['solution']
print('Solution status: ' + solution_description['status'])

# Use the solution ARN to create a solution version.
print ('Creating solution version')

```

```
response = personalize.create_solution_version(solutionArn = solution_arn)
solution_version_arn = response['solutionVersionArn']
print('Solution version ARN: ' + solution_version_arn)

# Use the solution version ARN to get the solution version status.
solution_version_description = personalize.describe_solution_version(
    solutionVersionArn = solution_version_arn)['solutionVersion']
print('Solution version status: ' + solution_version_description['status'])
```

SDK for Java 2.x

```
public static String createPersonalizeSolutionVersion(PersonalizeClient
personalizeClient, String solutionArn) {
    long maxTime = 0;
    long waitInMilliseconds = 30 * 1000; // 30 seconds
    String solutionStatus = "";
    String solutionVersionStatus = "";
    String solutionVersionArn = "";

    try {
        DescribeSolutionRequest describeSolutionRequest =
DescribeSolutionRequest.builder()
            .solutionArn(solutionArn)
            .build();

        maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

        // Wait until solution is active.
        while (Instant.now().getEpochSecond() < maxTime) {

            solutionStatus =
personalizeClient.describeSolution(describeSolutionRequest).solution().status();
            System.out.println("Solution status: " + solutionStatus);

            if (solutionStatus.equals("ACTIVE") || solutionStatus.equals("CREATE
FAILED")) {
                break;
            }
            try {
                Thread.sleep(waitInMilliseconds);
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
    }
}
```

```
    }

    // Once the solution is active, start creating a solution version.

    if (solutionStatus.equals("ACTIVE")) {

        CreateSolutionVersionRequest createSolutionVersionRequest =
CreateSolutionVersionRequest.builder()
        .solutionArn(solutionArn)
        .build();

        CreateSolutionVersionResponse createSolutionVersionResponse =
personalizeClient.createSolutionVersion(createSolutionVersionRequest);
        solutionVersionArn =
createSolutionVersionResponse.solutionVersionArn();

        System.out.println("Solution version ARN: " + solutionVersionArn);

        DescribeSolutionVersionRequest describeSolutionVersionRequest =
DescribeSolutionVersionRequest.builder()
        .solutionVersionArn(solutionVersionArn)
        .build();

        maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

        while (Instant.now().getEpochSecond() < maxTime) {

            // Use the solution version ARN to get the solution version
status.
            solutionVersionStatus =
personalizeClient.describeSolutionVersion(describeSolutionVersionRequest).solutionVersion()
            System.out.println("Solution version status: " +
solutionVersionStatus);

            if (solutionVersionStatus.equals("ACTIVE") ||
solutionVersionStatus.equals("CREATE FAILED")) {
                break;
            }
            try {
                Thread.sleep(waitInMilliseconds);
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
    }
}
```



```
        return solutionVersionArn;
    }
} catch(PersonalizeException e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
return "";
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { CreateSolutionVersionCommand } from
    "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";
// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the solution version parameters.
export const solutionVersionParam = {
    solutionArn: 'SOLUTION_ARN' /* required */
}

export const run = async () => {
    try {
        const response = await personalizeClient.send(new
        CreateSolutionVersionCommand(solutionVersionParam));
        console.log("Success", response);
        return response; // For unit tests.
    } catch (err) {
        console.log("Error", err);
    }
};
run();
```

현재의 솔루션 버전 상태를 확인하려면 [DescribeSolutionVersion](#) 작업을 호출하고 `CreateSolutionVersion` 작업에서 반환된 솔루션 버전의 ARN을 전달합니다. `status`이 `ACTIVE`이면 교육이 완료된 것이므로, `Personalize`에서 제공하는 지표를 사용하여 교육을 평가할 수

있습니다. 자세한 내용은 [지표를 사용하여 솔루션 버전 평가](#) 단원을 참조하세요. 오류로 인해 교육이 완료되지 않은 경우, 교육 비용은 청구되지 않습니다.

솔루션 버전이 CREATE_PENDING 또는 CREATE_IN_PROGRESS 상태인 경우, [StopSolutionVersionCreation](#) 작업을 사용하여 솔루션 버전 생성 프로세스를 중지할 수 있습니다. [솔루션 버전 생성 중지](#) 단원을 참조하세요.

액티브 솔루션 버전을 사용하는 방법은 권장 사항을 받는 방법에 따라 달라집니다.

- 실시간 추천을 받으려면 Amazon Personalize 캠페인과 함께 액티브 솔루션 버전을 배포하십시오. 캠페인을 사용하여 사용자를 위한 추천을 받을 수 있습니다. [캠페인 생성](#) 섹션을 참조하십시오.
- 일괄 추천의 경우 일괄 추론 작업 또는 배치 세그먼트 작업을 생성할 때 ACTIVE 솔루션 버전을 지정합니다. [배치 권장 사항 및 사용자 세그먼트\(사용자 지정 리소스\)](#) 섹션을 참조하세요.

솔루션 버전 생성 중지

솔루션 버전의 상태가 CREATE_PENDING 또는 CREATE_IN_PROGRESS인 경우 Amazon Personalize 콘솔 또는 [StopSolutionVersionCreation](#) 작업을 사용하여 솔루션 버전 생성을 중지(모델 학습 중지)할 수 있습니다. 중지된 후에는 솔루션 버전 생성을 재개할 수 없습니다. 솔루션 버전 생성이 중지된 시점까지 사용된 리소스에 대한 요금이 청구됩니다.

솔루션 버전 생성을 중지하면 모델 학습이 종료되지만 솔루션 버전이 삭제되지는 않습니다. Amazon Personalize 콘솔에서 [DescribeSolutionVersion](#) 작업을 사용하여 솔루션 버전 세부 정보를 계속 볼 수 있습니다.

Amazon Personalize 콘솔, AWS Command Line Interface(AWS CLI) 또는 AWS SDK를 사용하여 솔루션 버전 생성 프로세스를 중지할 수 있습니다.

주제

- [솔루션 버전 생성 중지\(콘솔\)](#)
- [솔루션 버전 생성 중지\(AWS CLI\)](#)
- [솔루션 버전 생성 중지\(AWS SDK\)](#)

솔루션 버전 생성 중지(콘솔)

솔루션 버전의 상태가 CREATE_PENDING 또는 CREATE_IN_PROGRESS인 경우 솔루션 버전 생성을 중지(모델 학습 중지)할 수 있습니다.

솔루션 버전 생성을 중지하려면(콘솔)

1. <https://console.aws.amazon.com/personalize/home> 에서 Amazon Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터 세트 그룹 페이지에서 중지하려는 솔루션 버전이 포함된 데이터 세트 그룹을 선택합니다.
3. 탐색 창에서 솔루션 및 레시피를 선택합니다.
4. 솔루션 및 레시피 페이지에서 중지하려는 솔루션 버전의 솔루션을 선택합니다.
5. 솔루션 버전에서 중지하려는 솔루션 버전을 선택합니다.
6. 솔루션 버전 세부 정보 페이지에서 생성 중지를 선택합니다. 솔루션 버전의 원래 상태에 따라 솔루션 버전 상태가 다음과 같이 변경됩니다.
 - CREATE_PENDING이 CREATE_STOPPED로 변경됨.
 - CREATE_IN_PROGRESS가 CREATE_STOPPINGERS로 변경된 다음 CREATE_STOPONSEP으로 변경됨.

솔루션 버전 생성 중지(AWS CLI)

솔루션 버전의 상태가 CREATE_PENDING 또는 CREATE_IN_PROGRESS인 경우 솔루션 버전 생성을 중지(모델 학습 중지)할 수 있습니다. AWS CLI에서 솔루션 버전 생성을 중지하려면 다음 `stop-solution-version-creation` 명령을 사용합니다. `solution version arn`을 중지하려는 솔루션 버전의 Amazon 리소스 이름(ARN)으로 바꿉니다. 솔루션 버전 생성이 중지된 시점까지 사용한 리소스에 대한 요금이 청구됩니다.

```
aws personalize stop-solution-version-creation \
  --solution-version-arn solution version arn
```

`describe-solution-version` 명령을 사용하여 솔루션 버전의 학습 상태를 확인합니다.

```
aws personalize describe-solution-version \
  --solution-version-arn solution version arn
```

솔루션 버전의 원래 상태에 따라 솔루션 버전 상태가 다음과 같이 변경됩니다.

- CREATE_PENDING이 CREATE_STOPPED로 변경됨.
- CREATE_IN_PROGRESS가 CREATE_STOPPINGERS로 변경된 다음 CREATE_STOPPED로 변경됨

솔루션 버전 생성 중지(AWS SDK)

솔루션 버전의 상태가 CREATE_PENDING 또는 CREATE_IN_PROGRESS인 경우 솔루션 버전 생성을 중지(모델 학습 중지)할 수 있습니다. 다음 코드는 AWS SDK for Python (Boto3) 또는 AWS SDK for Java 2.x에서 솔루션 버전 생성을 중지하는 방법을 보여줍니다. 솔루션 버전 생성이 중지된 시점까지 사용된 리소스에 대한 요금이 청구됩니다.

SDK for Python (Boto3)

솔루션 버전 생성을 중지하려면 다음 `stop_solution_version_creation` 메서드를 사용합니다. `solution_version_arn`을 중지하려는 솔루션 버전의 Amazon 리소스 이름(ARN)으로 바꿉니다. 메서드는 [DescribeSolutionVersion](#) 작업을 사용하여 솔루션 버전의 상태를 검색합니다.

```
import boto3

personalize = boto3.client('personalize')

response = personalize.stop_solution_version_creation(
    solutionVersionArn = solution_version_arn
)

# Use the solution version ARN to get the solution version status.
solution_version_description = personalize.describe_solution_version(
    solutionVersionArn = solution_version_arn)['solutionVersion']
print('Solution version status: ' + solution_version_description['status'])
```

SDK for Java 2.x

솔루션 버전 생성을 중지하려면 다음 `stopSolutionVersionCreation` 메서드를 사용합니다. 파라미터를 Amazon Personalize 서비스 클라이언트와 생성을 중지하려는 솔루션 버전의 Amazon 리소스 이름(ARN)을 전달합니다. 다음 코드는 [DescribeSolutionVersion](#) 작업을 사용하여 솔루션 버전의 상태를 검색합니다.

```
public static void stopSolutionVersionCreation(PersonalizeClient personalizeClient,
    String solutionVersionArn) {
    String solutionVersionStatus = "";

    StopSolutionVersionCreationRequest stopSolutionVersionCreationRequest =
    StopSolutionVersionCreationRequest.builder()
        .solutionVersionArn(solutionVersionArn)
```

```

        .build();

personalizeClient.stopSolutionVersionCreation(stopSolutionVersionCreationRequest);

// Use the solution version ARN to get the solution version status.
DescribeSolutionVersionRequest describeSolutionVersionRequest =
DescribeSolutionVersionRequest.builder()
    .solutionVersionArn(solutionVersionArn)
    .build();

solutionVersionStatus =
personalizeClient.describeSolutionVersion(describeSolutionVersionRequest)
    .solutionVersion()
    .status();
System.out.println("Solution version status: " + solutionVersionStatus);
}

```

솔루션 버전의 원래 상태에 따라 솔루션 버전 상태가 다음과 같이 변경됩니다.

- CREATE_PENDING이 CREATE_STOPPED로 변경됨.
- CREATE_IN_PROGRESS가 CREATE_STOPPINGERS로 변경된 다음 CREATE_STOPONSEP으로 변경됨.

지표를 사용하여 솔루션 버전 평가

오프라인 지표와 온라인 지표를 통해 솔루션 버전의 성능을 평가할 수 있습니다. 온라인 지표는 실시간 추천과의 사용자 상호작용에서 보게 되는 경험적 결과입니다. 예를 들어 카탈로그를 탐색하는 사용자의 클릭률을 기록할 수 있습니다. 모든 온라인 지표를 생성하고 기록하는 것은 귀하의 책임입니다.

오프라인 지표는 Personalize가 솔루션 버전을 학습시킬 때 생성하는 지표입니다. 캠페인을 생성하고 추천을 제공하기 전에 오프라인 지표를 사용하여 모델의 성능을 평가할 수 있습니다. 오프라인 지표를 사용하면 솔루션의 하이퍼파라미터를 수정한 효과를 보거나 동일한 데이터로 학습된 모델의 결과를 비교할 수 있습니다. 이 단원의 나머지 부분에서 지표란 용어는 오프라인 지표를 가리킵니다.

Amazon Personalize는 성능 지표를 얻기 위해 입력된 상호 작용 데이터를 교육 세트, 테스트 세트, PERSONALIZED_ACTIONS의 경우 검증 세트로 분할합니다. 분리는 선택한 레시피 유형에 따라 달라집니다.

- 사용자_세분화 레시피의 경우 학습 세트는 각 사용자 상호작용 데이터의 80%로 구성되고 테스트 세트는 각 사용자 상호작용 데이터의 20%로 구성됩니다.
- 다른 모든 레시피 유형의 경우 학습 세트는 사용자의 90%와 이들의 상호작용 데이터로 구성됩니다. 테스트 세트는 나머지 10%의 사용자와 이들의 상호작용 데이터로 구성됩니다.

그러면 Personalize가 학습 세트를 사용하여 솔루션 버전을 생성합니다. 학습이 완료되면 Personalize는 새 솔루션 버전에 테스트 세트의 각 사용자 데이터 중 가장 오래된 90%를 입력으로 제공합니다. 그런 다음 Personalize는 솔루션 버전에서 생성한 추천을 테스트 세트의 각 사용자 데이터 중 최신 10%의 실제 상호작용과 비교하여 지표를 계산합니다.

비교를 위한 기준을 생성하려면 가장 인기 있는 상위 K개 항목을 추천하는 [인기도-집계](#) 레시피를 사용하는 것이 좋습니다.

주제

- [솔루션 버전 지표 검색](#)
- [지표 정의](#)
- [예](#)
- [추가적인 리소스](#)

솔루션 버전 지표 검색

솔루션 버전을 만든 후에는 지표를 사용하여 성능을 평가할 수 있습니다. Amazon Personalize 콘솔, AWS Command Line Interface (AWS CLI) 및 AWS SDK를 사용하여 솔루션 버전의 지표를 검색할 수 있습니다.

주제

- [솔루션 버전 지표 검색\(콘솔\)](#)
- [솔루션 버전 지표 검색\(AWS CLI\)](#)
- [솔루션 버전 지표 \(SDK\) 검색AWS](#)

솔루션 버전 지표 검색(콘솔)

콘솔에서 추천 지표를 보려면 솔루션 버전의 세부 정보 페이지로 이동합니다.

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.

2. 데이터세트 그룹 페이지에서 사용자 지정 데이터세트 그룹을 선택합니다.
3. 탐색 창에서 사용자 지정 리소스를 선택한 후 솔루션 및 레시피를 선택합니다.
4. 솔루션을 선택합니다.
5. 솔루션 버전에서 세부 정보 페이지를 보려는 솔루션 버전을 선택합니다. 지표가 하단 창의 솔루션 버전 지표 탭에 나열됩니다. 지표의 정의는 [지표 정의](#) 단원을 참조하세요.

이제 솔루션 버전을 평가했으므로 사용 사례에 가장 적합한 지표를 사용해서 솔루션 버전을 배포하여 캠페인을 생성합니다. 솔루션 배포에 대한 자세한 내용은 [캠페인 생성](#) 단원을 참조하세요.

솔루션 버전 지표 검색(AWS CLI)

[GetSolutionMetrics](#) 작업을 호출하여 특정 솔루션 버전에 대한 지표를 검색합니다. 다음 코드는 AWS CLI를 사용하여 지표를 검색하는 방법을 보여줍니다.

```
personalize get-solution-metrics --solution-version-arn solution version ARN
```

다음은 추가 최적화 목표가 있는 [사용자-개인 맞춤](#) 레시피를 사용하여 만든 솔루션 버전의 결과 예시입니다.

```
{
  "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/SolutionName/
<version-id>",
  "metrics": {
    "coverage": 0.27,
    "mean_reciprocal_rank_at_25": 0.0379,
    "normalized_discounted_cumulative_gain_at_5": 0.0405,
    "normalized_discounted_cumulative_gain_at_10": 0.0513,
    "normalized_discounted_cumulative_gain_at_25": 0.0828,
    "precision_at_5": 0.0136,
    "precision_at_10": 0.0102,
    "precision_at_25": 0.0091,
    "average_rewards_at_k": 0.653
  }
}
```

각 지표에 대한 설명은 [지표 정의](#) 단원을 참조하세요. 이제 솔루션 버전을 평가했으므로 사용 사례에 가장 적합한 지표를 사용해서 솔루션 버전을 배포하여 캠페인을 생성합니다. 솔루션 배포에 대한 자세한 내용은 [캠페인 생성](#) 단원을 참조하세요.

솔루션 버전 지표 (SDK) 검색AWS

[GetSolutionMetrics](#) 작업을 호출하여 특정 솔루션 버전에 대한 지표를 검색합니다. 다음 코드를 사용하여 지표를 검색합니다.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.get_solution_metrics(
    solutionVersionArn = 'solution version arn')

print(response['metrics'])
```

SDK for Java 2.x

```
public static void getSolutionVersionMetrics(PersonalizeClient personalizeClient,
    String solutionVersionArn) {

    try {
        GetSolutionMetricsRequest request = GetSolutionMetricsRequest.builder()
            .solutionVersionArn(solutionVersionArn)
            .build();
        Map<String, Double> metrics =
personalizeClient.getSolutionMetrics(request).metrics();
        metrics.forEach((key, value) -> System.out.println(key + " " + value));
    } catch (PersonalizeException e ) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

다음은 추가 최적화 목표가 있는 [사용자-개인 맞춤](#) 레시피를 사용하여 만든 솔루션 버전의 결과 예시입니다.

```
{
  "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/
MovieSolution/<version-id>",
  "metrics": {
```



```

    "coverage": 0.27,
    "mean_reciprocal_rank_at_25": 0.0379,
    "normalized_discounted_cumulative_gain_at_5": 0.0405,
    "normalized_discounted_cumulative_gain_at_10": 0.0513,
    "normalized_discounted_cumulative_gain_at_25": 0.0828,
    "precision_at_5": 0.0136,
    "precision_at_10": 0.0102,
    "precision_at_25": 0.0091,
    "average_rewards_at_k": 0.653
  }
}

```

각 지표에 대한 설명은 [지표 정의](#) 단원을 참조하세요. 이제 솔루션 버전을 평가했으므로 사용 사례에 가장 적합한 지표를 사용해서 솔루션 버전을 배포하여 캠페인을 생성합니다. 솔루션 배포에 대한 자세한 내용은 [캠페인 생성](#) 단원을 참조하세요.

지표 정의

Personalize가 솔루션 버전에 대해 생성하는 지표는 다음과 같은 용어를 사용하여 아래에 설명되어 있습니다.

- 관련 추천은 사용자가 실제로 상호작용한 항목에 대한 추천입니다. 이들 항목은 테스트 세트의 각 사용자 상호작용 데이터 중 최신 10%에서 가져온 것입니다.
- 순위는 추천 목록 내에서 해당 추천 항목의 위치를 말합니다. 위치 1(목록의 맨 위)이 사용자와 가장 관련 있는 것으로 간주됩니다.

각 지표의 경우 숫자가 높을수록(1에 가까울수록) 더 좋습니다. 더 자세히 알아보려면 [추가적인 리소스](#)에 나열된 리소스를 참조하세요.

비율

커버리지 값은 Amazon Personalize가 데이터 세트의 총 고유 레코드 수 중에서 추천할 수 있는 고유 항목 (항목 권장 사항), 작업 (작업 권장 사항) 또는 사용자 (사용자 세그먼트 권장 사항) 의 비율을 나타냅니다.

적용 범위 점수가 높을수록 Amazon Personalize에서 동일한 레코드보다 더 많은 카탈로그를 추천합니다. 사용자-개인 맞춤과 같이 항목 탐색이 포함된 레시피는 그렇지 않은 레시피(예: 유사-항목)보다 적용 범위 점수가 더 높습니다.

평균 역수 순위 25

이 지표는 관련 항목 추천을 상위 순위에서 생성하는 모델의 능력을 보여줍니다.

사용자에 대한 항목 검색 결과를 생성하는 경우 평균 역수 순위가 25로 높은 모델을 선택할 수 있으며, 사용자가 목록에서 더 낮은 항목을 선택할 것으로 예상하지 않습니다. 예를 들어 사용자는 검색 결과에서 첫 번째 요리 레시피를 선택하는 경우가 많습니다. Amazon Personalize는 PERSONALIZED_ACTIONS 또는 USER_SEGMENTATION 레시피에 대해서는 이 지표를 생성하지 않습니다.

Personalize는 추천 요청에 대해 평균 역수 순위 점수를 사용하여 이 지표를 계산합니다. 각 역수 순위 점수는 $1 / \text{the rank of the highest item interacted with by the user}$ 로 계산되는데, 여기서 가능한 총 순위는 25개입니다. 해당 사용자가 상호작용하는 더 낮은 순위의 다른 항목은 무시됩니다. 사용자가 첫 번째 항목을 선택한 경우 점수는 1입니다. 항목을 선택하지 않은 경우 점수는 0입니다.

예를 들어 세 명의 사용자에게 각각 25개의 추천을 표시할 수 있습니다.

- 사용자 1이 4 순위의 항목과 10 순위의 항목을 클릭한 경우 역수 순위 점수는 1/4이 됩니다.
- 사용자 2가 2 순위의 항목, 4 순위의 항목, 12 순위의 항목을 클릭한 경우 역수 순위 점수는 1/2이 됩니다.
- 사용자 3이 6 순위의 단일 항목을 클릭한 경우 역수 순위 점수는 1/6이 됩니다.

모든 추천 요청의 평균 역수 순위(이 경우 3)는 $(1/4 + 1/2 + 1/6) / 3 = .3056$ 로 계산됩니다.

정규화된 할인 누적 이득(NDCG) K(5/10/25)

이 지표는 모델이 항목 또는 작업 추천의 순위를 얼마나 잘 매겼는지 알려줍니다. 여기서 K는 5, 10 또는 25개의 추천으로 구성된 표본 크기입니다. 이 지표는 순위가 가장 높은 항목 또는 작업 이외의 추천 순위에 관심이 있는 경우에 유용합니다 (자세한 내용은 참조 [mean reciprocal rank at 25](#)). 예를 들어 한 번에 최대 10가지의 영화를 캐루셀에 표시하는 애플리케이션이 있다면 NDCG at 10의 점수가 유용할 것입니다.

Personalize는 테스트 세트의 각 사용자에게 대한 순위 위치를 기반으로 추천에 가중치를 할당하여 NDCG를 계산합니다. 각 추천은 위치에 따라 계수만큼 할인됩니다(더 낮은 가중치가 부여됨). 최종 지표는 테스트 NDCG at K 세트에 속한 모든 사용자의 평균입니다. 는 목록에서 아래에 있는 권장 사항이 목록의 상위에 있는 권장 사항보다 관련성이 낮다고 NDCG at K 가정합니다.

Personalize에서는 $1/\log(1 + \text{position})$ 의 가중 계수를 사용하며 목록의 맨 위가 위치 1입니다.

정밀도 K

이 지표는 K(5, 10 또는 25)개 추천의 표본 크기를 기반으로 한 모델 추천의 관련성을 알려줍니다.

Personalize는 테스트 세트의 각 사용자에게 대한 상위 K개 추천 중 관련성이 있는 추천 수를 K로 나눈 값을 기반으로 이 지표를 계산합니다. 여기서 K는 5, 10 또는 25입니다. 최종 지표는 테스트 세트에 속한 모든 사용자의 평균입니다.

예를 들어, 사용자에게 10개의 항목을 추천하고 사용자가 그 중 3개와 상호작용하는 경우 정밀도 K는 올바르게 예측된 항목 3개를 총 10개의 추천 항목으로 나눈 값, $3 / 10 = .30$ 이 됩니다.

이 지표는 관련 항목의 정확한 추천을 보상합니다. 점수가 1에 가까울수록 모델의 정밀도가 높아집니다.

precision

차선책-작업 레시피로 솔루션 버전을 교육하는 경우 Amazon Personalize는 precision at K 대신 precision 지표를 생성합니다. 이 지표는 모델이 사용자가 실제로 수행할 작업을 얼마나 잘 예측하는지 알려줍니다.

Amazon Personalize는 데이터 세트의 각 작업에 대해 해당 작업을 수행할 것으로 올바르게 예측된 사용자 수를 해당 작업이 추천된 총 횟수로 나누어 precision을 계산합니다. 그 후 Amazon Personalize는 데이터 세트의 모든 작업에 대한 평균을 계산합니다.

예를 들어, 100명의 사용자에게 작업을 추천했는데 60명의 사용자가 해당 작업을 수행했고 40명의 사용자는 수행하지 않은 경우 해당 작업의 precision은 $60 / 100 = .60$ 입니다. 그러면 Amazon Personalize는 이 계산을 모든 작업에 적용하고 평균을 반환합니다.

이 지표는 관련 작업의 정확한 추천을 보상합니다. 점수가 1에 가까울수록 모델의 정밀도가 높아집니다.

average_rewards_at_k

최적화 목표를 가진 솔루션에 대한 솔루션 버전을 생성(모델을 학습)하면 Personalize가 average_rewards_at_k 지표를 생성합니다. average_rewards_at_k의 점수는 목표 달성에 있어 솔루션 버전이 얼마나 잘 수행되는지를 나타냅니다. 이 지표를 계산하기 위해 Personalize는 다음과 같이 각 사용자에게 대한 보상을 계산합니다.

```
rewards_per_user = total rewards from the user's interactions with their
top 25 reward generating recommendations / total rewards from the user's
interactions with recommendations
```

최종 average_rewards_at_k는 1보다 작거나 같고 0보다 큰 10진수 값으로 정규화된 모든 rewards_per_user의 평균입니다. 값이 1에 가까울수록 추천에서 기대할 수 있는 사용자당 평균 이익이 더 커집니다.

예를 들어 클릭으로 인한 수익을 극대화하는 것이 목표인 경우 Personalize는 가장 비용이 많이 드는 상위 25개 추천 항목 중에서 사용자가 클릭한 항목에서 발생한 총 수익을 사용자가 클릭한 모든 추천 항목의 수익으로 나누어 각 사용자 점수를 계산합니다. 그러면 Personalize는 모든 사용자 점수의 정규화된 평균을 반환합니다. `average_rewards_at_k`이 1에 가까울수록 추천을 통해 사용자당 얻을 수 있는 평균 수익이 더 커집니다.

자세한 내용은 [추가 목표에 맞게 솔루션 최적화](#) 단원을 참조하세요.

추세 예측 정확도

지금-유행 레시피로 솔루션 버전을 학습시킨 경우 모델에서 추천하는 항목의 인기 증가율. 추세 예측 정확도가 높을수록(1에 가까울수록) 모델이 유행 항목을 더 정확하게 식별할 수 있습니다.

인기 가속화를 계산하기 위해 Personalize는 모든 추천 아이템의 인기도 증가율을 상위 25개 인기 항목의 총 인기도 증가율로 나눕니다. 이들 항목은 테스트 세트의 실제 상호작용에서 나온 것입니다.

데이터 분포와 추세 발견 빈도에 따라 추세 예측 정확도 값은 0.0이 될 수 있습니다.

히트(K 히트)

사용자_세분화 레시피로 솔루션 버전을 학습시킨 경우, 실제 사용자와 일치하는 예측된 상위 관련 K개 결과의 평균 사용자 수. 실제 사용자는 테스트 세트의 항목과 실제로 상호작용한 사용자입니다. K는 관련성이 가장 높은 사용자 중 상위 1%입니다. 값이 클수록 예측의 정확도가 높아집니다.

재현율(K 재현율)

사용자_세분화 레시피로 솔루션 버전을 학습시킨 경우, 실제 사용자와 일치하는 예측된 상위 연관성 상위 K개 결과에서 예측 사용자의 평균 비율. 실제 사용자는 테스트 세트의 항목과 실제로 상호작용한 사용자입니다. K는 관련성이 가장 높은 사용자 중 상위 1%입니다. 값이 클수록 예측의 정확도가 높아집니다.

리콜

차선택-작업 레시피로 솔루션 버전을 교육하는 경우, 이 지표를 통해 솔루션 버전이 사용자가 상호작용할 작업을 찾는 데 얼마나 효과적인지 알 수 있습니다.

Amazon Personalize는 데이터 세트의 각 작업에 대해 해당 작업을 수행할 것으로 올바르게 예측된 사용자 수를 테스트 세트에서 실제로 작업을 수행한 총 사용자 수로 나누어 `recall`을 계산합니다. 그 후 Amazon Personalize는 데이터 세트의 모든 작업에 대한 평균을 계산합니다.

예를 들어, 테스트 세트에서 100명의 사용자가 작업을 수행하고 Amazon Personalize에서 이러한 사용자 중 50명이 작업을 수행할 것으로 예측한 경우 작업에 대한 `recall`은 $50 / 100 = .50$ 입니다. 그러면 Amazon Personalize는 이 계산을 모든 작업에 적용하고 평균을 반환합니다.

곡선 아래 면적(AUC)

PERSONALIZED_ACTIONS 레시피로 솔루션 버전을 교육한 경우 해당 솔루션 버전의 수신기 작동 특성 곡선 아래 면적입니다. 이 지표는 사용자가 수행할 작업을 정확히 식별하여 솔루션 버전이 얼마나 잘 수행되는지를 보여줍니다.

수신기 작동 특성 곡선은 솔루션 버전의 성능을 나타냅니다. 이 곡선은 서로 다른 임계값에서 참 긍정(관련이 있다고 올바르게 예측된 작업) 및 거짓 긍정(관련이 있다고 잘못 예측된 작업) 비율을 표시합니다. 곡선 아래 면적(AUC)은 곡선을 기반으로 솔루션 버전의 성능을 요약한 점수입니다.

솔루션 버전의 AUC는 0에서 1 사이일 수 있습니다. 1에 가까울수록 모델이 사용자의 관련 작업을 더 잘 예측할 수 있습니다.

예

다음은 솔루션 버전이 특정 사용자에게 대해 추천 목록을 생성하는 간단한 예제입니다. 두 번째 추천과 다섯 번째 추천은 이 사용자에게 대한 테스트 데이터의 레코드와 일치합니다. 이들 두 가지는 관련 추천입니다. K을 5로 설정하면 사용자에게 대해 다음 지표가 생성됩니다.

reciprocal_rank

계산: $1/2$

결과: 0.5000

normalized_discounted_cumulative_gain_at_5

계산: $(1/\log(1 + 2) + 1/\log(1 + 5)) / (1/\log(1 + 1) + 1/\log(1 + 2))$

결과: 0.6241

precision_at_5

계산: $2/5$

결과: 0.4000

추가적인 리소스

A/B 테스트를 통한 솔루션 버전 평가에 대한 자세한 내용은 [A/B 테스트를 사용하여 Personalize에서 생성한 추천의 효율성 측정](#) 단원을 참조하세요. 추천자 시스템을 위한 다양한 유형의 지표를 자세히 알아보려면 다음 외부 리소스를 참조하세요.

- [MRR 대 MAP 대 NDCG: 순위 인식 평가 지표 및 사용 시기](#)
- [할인된 누적 이득: 알아야 할 순위 지표](#)
- [추천자 시스템의 k 재현율 및 정밀도](#)
- [추천자 시스템의 순위 평가 지표](#)
- [수신기 작동 특성](#)

캠페인 생성

사용자 지정 리소스가 포함된 실시간 추천의 경우, [솔루션 버전 생성](#)을 완료하고 나면 캠페인과 함께 솔루션 버전을 배포할 준비가 된 것입니다.

캠페인에서는 프로비저닝 트랜잭션 용량이 포함된 솔루션 버전(학습된 모델)을 배포하여 실시간 추천을 생성합니다. 캠페인을 생성한 후 [GetRecommendations](#) 또는 [GetPersonalizedRankingAPI](#) 작업을 사용하여 추천을 얻을 수 있습니다. 배치 추천을 받는 경우 캠페인을 생성할 필요가 없습니다. 자세한 내용은 [배치 권장 사항 및 사용자 세그먼트\(사용자 지정 리소스\)](#) 단원을 참조하세요.

캠페인을 생성할 때 다음을 구성할 수 있습니다.

- 솔루션의 최신 솔루션 버전을 사용하도록 캠페인이 자동으로 업데이트되도록 구성할 수 있습니다. 자세한 내용은 [자동 캠페인 업데이트 활성화](#) 단원을 참조하세요.
- 권장 사항에서 항목 메타데이터를 활성화할 수 있습니다. 자세한 정보는 [권장 사항의 항목 메타데이터](#)를 참조하세요.
- 캠페인에 초당 프로비저닝된 최소 거래를 지정할 수 있습니다. 이는 Amazon Personalize에서 프로비저닝된 캠페인의 기준 트랜잭션 처리량입니다. 캠페인이 활성 상태일 때의 최소 청구 요금을 설정합니다. 자세한 정보는 [초당 최소 프로비저닝 트랜잭션 수 및 자동 스케일링](#)을 참조하세요.

Amazon Personalize 콘솔, AWS Command Line Interface (AWS CLI) 또는 AWS SDK를 사용하여 캠페인을 생성할 수 있습니다. 권장 사항에서 메타데이터를 활성화하는 등 기존 캠페인의 설정을 변경하려면 캠페인을 업데이트해야 합니다. 자세한 내용은 [캠페인 업데이트](#) 단원을 참조하세요.

캠페인이 활성화된 동안에는 캠페인 비용이 발생합니다. 불필요한 비용이 발생하지 않도록 하려면 완료 후 캠페인을 삭제하세요. 캠페인 비용에 대한 자세한 내용은 [Amazon Personalize](#) 가격을 참조하십시오.

주제

- [자동 캠페인 업데이트 활성화](#)

- [초당 최소 프로비저닝 트랜잭션 수 및 자동 스케일링](#)
- [권장 사항의 항목 메타데이터](#)
- [캠페인 생성\(콘솔\)](#)
- [캠페인 생성\(AWS CLI\)](#)
- [캠페인 \(AWS SDK\) 만들기](#)
- [캠페인 업데이트](#)

자동 캠페인 업데이트 활성화

캠페인을 생성할 때 자동 캠페인 업데이트를 활성화할 수 있습니다. 자동 업데이트를 사용하면 캠페인이 자동으로 업데이트되어 솔루션의 최신 자동 또는 수동 교육 솔루션 버전을 배포할 수 있습니다. 이렇게 하면 캠페인을 최신 상태로 유지하기가 더 쉬워집니다.

예를 들어 솔루션에서 [자동 교육을](#) 사용하여 7일마다 새 솔루션 버전을 만들면 매주 교육마다 최신 솔루션 버전을 사용하도록 캠페인이 자동으로 업데이트됩니다. 자동 캠페인 업데이트를 사용하지 않는 경우 캠페인을 수동으로 업데이트하여 학습된 최신 모델을 배포해야 합니다.

- Amazon Personalize 콘솔로 캠페인을 생성할 때 자동 캠페인 업데이트를 활성화하려면 캠페인 세부 정보에서 자동 업데이트를 선택하여 솔루션의 최신 솔루션 버전을 사용하십시오. 캠페인 세부 정보 페이지에서 최신 업데이트의 타임스탬프를 찾을 수 있습니다.

자세한 정보는 [캠페인 생성\(콘솔\)](#)을 참조하세요.

- [CreateCampaign](#) API 작업을 사용할 때 자동 캠페인 업데이트를 활성화하려면 SolutionVersionArn 파라미터에 솔루션의 Amazon 리소스 이름 (ARN) 을 형식으로 SolutionArn/\$LATEST 지정하십시오. campaignConfig에서는 enableMetadataInInferenceResponse 로 설정합니다. true

최신 캠페인 업데이트의 타임스탬프를 가져오려면 [DescribeCampaign](#) API 작업을 사용하고 응답에서 latestCampaignUpdate 세부 정보를 확인할 수 있습니다.

자동 업데이트를 활성화하는 방법을 보여주는 코드 샘플은 [캠페인 생성\(AWS CLI\)](#) 또는 [캠페인 \(AWS SDK\) 만들기](#) 을 참조하십시오.

초당 최소 프로비저닝 트랜잭션 수 및 자동 스케일링

Important

minProvisionedTPS가 높으면 비용이 증가합니다. minProvisionedTPS(기본값)의 경우 1부터 시작하는 것이 좋습니다. Amazon CloudWatch 지표를 사용하여 사용량을 추적하고 minProvisionedTPS 필요에 따라 사용량을 늘리십시오.

Amazon Personalize 캠페인을 생성할 때 캠페인에 대한 초당 최소 프로비저닝 트랜잭션 수 (minProvisionedTPS)를 지정할 수 있습니다. 이는 Amazon Personalize에서 프로비저닝한 캠페인의 기존 트랜잭션 처리량입니다. 캠페인이 활성 상태일 때의 최소 청구 요금을 설정합니다. 트랜잭션은 단일 GetRecommendations 또는 GetPersonalizedRanking 요청으로 이루어집니다. 기본 minProvisionedTPS 값은 1입니다.

TPS가 minProvisionedTPS 이상으로 증가할 경우 Amazon Personalize는 프로비저닝 용량을 자동으로 확장 및 축소하지만, minProvisionedTPS 이하로는 확장 및 축소하지 않습니다. 용량이 증가 중에 잠시 동안의 지연으로 거래 손실이 발생할 수 있습니다. 트래픽이 감소하면 용량은 minProvisionedTPS로 돌아옵니다.

프로비저닝된 최소 TPS 또는 요청이 실제 TPS를 초과하는 경우 요금이 부과됩니다.

minProvisionedTPS 실제 TPS는 추천을 요청한 총 건수입니다. 낮은 minProvisionedTPS 수치부터 시작하여 Amazon CloudWatch 측정치를 사용하여 사용량을 추적한 다음 minProvisionedTPS 필요에 따라 늘리는 것이 좋습니다.

캠페인 비용에 대한 자세한 내용은 [Amazon Personalize 요금](#)을 참조하세요.

권장 사항의 항목 메타데이터

Important

사용자 맞춤화-v2 또는 개인화된 순위-v2 레시피를 사용하는 경우 메타데이터에 대한 추가 비용이 발생하지 않습니다. 다른 모든 레시피와 모든 도메인 사용 사례의 경우 추가 비용이 발생합니다. 자세한 내용은 [Amazon Personalize 요금](#)을 참조하세요.

추천을 받으면 Amazon Personalize가 추천 결과에 항목 메타데이터를 포함하도록 할 수 있습니다. 요청 시 항목 데이터세트에서 포함할 열을 선택할 수 있습니다. Amazon Personalize는 추천 응답의 각 항목에 대해 이 데이터를 반환합니다.

메타데이터를 통해 사용자 인터페이스에서 추천을 강화할 수 있습니다(예: 캐러셀에 영화 장르 추가). 또는 이를 사용하여 추천 품질을 시각적으로 평가할 수도 있습니다. 앱에서 생성형 AI를 사용하는 경우 메타데이터를 AI 프롬프트에 연결하여 관련성이 더 높은 콘텐츠를 생성할 수 있습니다. Amazon Personalize를 생성형 AI와 함께 사용하는 방법에 대한 자세한 내용은 [Amazon Personalize 및 생성형 AI](#) 섹션을 참조하세요.

메타데이터 활성화

추천에 메타데이터를 추가하려면 메타데이터 열이 포함된 항목 데이터 세트가 있어야 합니다. 교육 시 메타데이터를 사용할 필요는 없습니다. 데이터 세트 생성에 대한 내용은 [데이터세트 및 스키마 생성](#) 섹션을 참조하세요. 데이터 관리 및 업데이트에 대한 내용은 [데이터셋의 훈련 데이터 관리](#) 섹션을 참조하세요.

사용자 맞춤 설정-v2 또는 맞춤형 순위-v2 레시피를 사용하는 경우 캠페인에는 추천 결과에 항목 메타데이터를 포함할 수 있는 옵션이 자동으로 제공됩니다. 캠페인의 메타데이터를 수동으로 활성화할 필요는 없습니다. 다른 모든 레시피와 도메인 사용 사례의 경우 메타데이터 옵션을 활성화해야 합니다.

- Amazon Personalize 콘솔에서 메타데이터를 활성화하려면 캠페인을 생성할 때 캠페인 세부 정보의 추천 결과에서 항목 메타데이터 반환을 선택합니다. 자세한 정보는 [캠페인 생성\(콘솔\)](#)을 참조하세요.
- AWS SDK를 사용하여 메타데이터를 활성화하거나 AWS CLI `campaignConfig enableMetadataInInferenceResponse` 설정하려면 `true` 설정된 [CreateCampaign](#) API 작업을 사용하십시오. 자세한 내용은 [캠페인 생성\(AWS CLI\)](#) 또는 [캠페인 \(AWS SDK\) 만들기](#)을 참조하세요.

캠페인 생성(콘솔)

Important

캠페인이 활성화된 동안에는 캠페인 비용이 발생합니다. 불필요한 비용이 발생하지 않도록 하려면 완료 후 캠페인을 삭제하세요. 캠페인 비용에 대한 자세한 내용은 [Amazon Personalize](#) 가격을 참조하십시오.

솔루션 버전 상태가 활성화되면 Personalize 캠페인을 통해 솔루션을 배포할 준비가 된 것입니다.

캠페인을 생성하려면(콘솔)

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.

2. 배포하려는 솔루션 버전이 포함된 데이터세트 그룹을 선택합니다.
3. 탐색 창의 사용자 지정 리소스에서 캠페인을 선택합니다.
4. 캠페인 페이지에서 캠페인 생성을 선택합니다.
5. 새 캠페인 생성 페이지에서 캠페인 세부 정보에 대해 다음 정보를 제공합니다.
 - 캠페인 이름 - 캠페인 이름을 입력합니다. 여기에 입력하는 텍스트는 캠페인 대시보드 및 세부 정보 페이지에 표시됩니다.
 - 솔루션 - 방금 만든 솔루션을 선택합니다.
 - 솔루션의 최신 솔루션 버전을 사용하도록 자동 업데이트 - 캠페인에서 최신 활성 솔루션 버전을 자동으로 사용하도록 하려면 이 옵션을 선택합니다. 이 옵션을 선택하지 않으면 새 솔루션 버전을 배포할 때마다 캠페인을 수동으로 업데이트해야 합니다. 자세한 정보는 [자동 캠페인 업데이트 활성화](#)를 참조하세요.
 - 솔루션 버전 ID - 자동 캠페인 업데이트를 사용하여 최신 솔루션 버전을 사용하지 않는 경우 배포하려는 솔루션 버전의 ID를 선택하십시오.
 - 초당 최소 프로비저닝된 트랜잭션(API에서 minProvisionedTPS라고 함) - Amazon Personalize에서 지원하는 초당 최소 프로비저닝 트랜잭션을 설정합니다. 값이 높으면 요금이 인상됩니다. 1(기본값)부터 시작하는 것이 좋습니다. Amazon CloudWatch 지표를 사용하여 사용량을 추적하고 minProvisionedTPS 필요에 따라 사용량을 늘리십시오. 자세한 정보는 [초당 최소 프로비저닝 트랜잭션 수 및 자동 스케일링](#)을 참조하세요.
 - 추천 결과에 항목 메타데이터 반환 - 추천 결과에 메타데이터를 포함하는 옵션을 원하면 이 옵션을 선택하십시오. 활성화하면 추천을 받을 때 항목 데이터셋의 열을 지정할 수 있습니다. 자세한 정보는 [권장 사항의 항목 메타데이터](#)을 참조하세요.
6. 사용자 맞춤 설정 레시피를 사용한 경우 캠페인 구성에서 탐색 가중치 및 탐색 항목 연령 컷오프 값을 선택적으로 입력할 수 있습니다. 자세한 정보는 [사용자-개인 맞춤](#)을 참조하세요.
7. 태그의 경우, 원하는 태그를 추가할 수 있습니다. Personalize 리소스 태그 지정에 대한 자세한 내용은 [Personalize 리소스에 태그 지정](#) 단원을 참조하세요.
8. 캠페인 생성을 선택합니다.
9. 캠페인 세부 정보 페이지에서 캠페인 상태가 활성일 때 캠페인을 사용하여 추천을 받고 노출 수를 기록할 수 있습니다. 자세한 내용은 [4단계: 추천 받기](#) 단원을 참조하세요.

캠페인 상태가 활성이면 캠페인 준비가 완료된 것입니다. 솔루션 버전을 재교육하거나 캠페인 설정을 변경하려면 캠페인을 업데이트해야 합니다. 자세한 정보는 [캠페인 업데이트](#)를 참조하세요.

캠페인 생성(AWS CLI)

Important

캠페인이 활성화된 동안에는 캠페인 비용이 발생합니다. 불필요한 비용이 발생하지 않도록 하려면 완료 후 캠페인을 삭제하세요. 캠페인 비용에 대한 자세한 내용은 [Amazon Personalize](#) 가격을 참조하십시오.

솔루션 버전이 활성 상태가 되면 Amazon Personalize 캠페인을 통해 배포할 준비가 된 것입니다. 를 사용하여 캠페인을 AWS CLI 생성하려면 `create-campaign` 명령을 사용합니다.

다음 코드 샘플은 캠페인을 만드는 방법을 보여줍니다. 사용자 개인화 레시피를 사용하는 솔루션의 최신 솔루션 버전을 배포합니다. 만든 캠페인은 향후 솔루션 버전을 사용할 수 있도록 자동으로 업데이트 됩니다. 코드는 다음 구성을 사용합니다.

- 솔루션에 맞는 최신 솔루션 버전을 사용하도록 캠페인이 자동으로 업데이트되도록 구성합니다. `solution-version-arn solution ARN/$LATEST` 형식은 다음과 같습니다. `syncWithLatestSolutionVersion True` 코드를 사용하려면 솔루션의 Amazon 리소스 이름 (ARN) `solution ARN` 으로 바꾸십시오.

`syncWithLatestSolutionVersion` 자동을 사용하지 않도록 설정하려면 솔루션 버전 ARN (제외/\$LATEST) 만 지정하고 로 설정합니다 `syncWithLatestSolutionVersion. False`

- `enableMetadataWithRecommendations` 옵션을 로 True 설정합니다. 이렇게 하면 추천 요청 옵션을 통해 항목 데이터세트의 항목 메타데이터를 추천 결과와 함께 포함할 수 있습니다. 이 옵션을 비활성화하려면 False 로 설정합니다. 자세한 정보는 [권장 사항의 항목 메타데이터](#)을 참조하세요.
- 1 (기본값) 로 설정됩니다. `min-provisioned-tps minProvisionedTPS(기본값)`의 경우 1부터 시작하는 것이 좋습니다. Amazon CloudWatch 지표를 사용하여 사용량을 추적하고 `minProvisionedTPS` 필요에 따라 사용량을 늘리십시오. 자세한 정보는 [초당 최소 프로비저닝 트랜잭션 수 및 자동 스케일링](#)을 참조하세요.

모든 파라미터의 전체 목록은 [CreateCampaign](#) 섹션을 참조하세요.

```
aws personalize create-campaign \
--name campaign-name \
--solution-version-arn solution-arn/$LATEST \
--min-provisioned-tps 1 \
```

```
--campaign-config "{\"syncWithLatestSolutionVersion\": \"true\",
  \"enableMetadataWithRecommendations\": \"true\"}"
```

캠페인 상태가 활성화되면 캠페인 준비가 완료된 것입니다. 현재 상태를 확인하려면 [DescribeCampaign](#)을 호출하고 status 필드가 ACTIVE인지 확인합니다.

솔루션 버전을 재교육했는데 캠페인이 최신 솔루션 버전을 사용하도록 자동으로 업데이트되지 않거나 캠페인 설정을 변경하려는 경우 캠페인을 업데이트해야 합니다. 자세한 정보는 [캠페인 업데이트](#)을 참조하세요.

Amazon Personalize는 생성한 캠페인을 [ListCampaigns](#) 나열하는 등의 캠페인 관리 작업을 제공합니다. 캠페인은 [DeleteCampaign](#)을 호출하여 삭제할 수 있습니다. 캠페인을 삭제하더라도 캠페인의 일부인 솔루션 버전은 삭제되지 않습니다.

캠페인을 생성한 후 이를 사용하여 추천할 수 있습니다. 자세한 정보는 [4단계: 추천 받기](#)을 참조하세요.

캠페인 (AWS SDK) 만들기

Important

캠페인이 활성화되는 동안 캠페인 비용이 발생합니다. 불필요한 비용이 발생하지 않도록 하려면 완료 후 캠페인을 삭제하세요. 캠페인 비용에 대한 자세한 내용은 [Amazon Personalize](#) 가격을 참조하십시오.

솔루션 버전이 활성화 상태가 되면 Amazon Personalize 캠페인을 통해 배포할 준비가 된 것입니다. AWS SDK로 캠페인을 생성하려면 [CreateCampaign](#) API 작업을 사용합니다.

다음 코드 샘플은 캠페인을 만드는 방법을 보여줍니다. 이 코드는 사용자 개인화 레시피를 사용하는 솔루션의 최신 솔루션 버전을 배포합니다. 만든 캠페인은 향후 솔루션 버전을 사용할 수 있도록 자동으로 업데이트됩니다. 코드는 다음 구성을 사용합니다.

- 솔루션에 맞는 최신 솔루션 버전을 사용하도록 캠페인이 자동으로 업데이트되도록 구성합니다. solutionVersionArn *solution ARN*/\$LATEST 형식은 다음과 같습니다. syncWithLatestSolutionVersion True 코드를 사용하려면 솔루션 버전의 Amazon 리소스 이름 (ARN) solution ARN 으로 바꾸십시오.

syncWithLatestSolutionVersion자동 사용하지 않도록 설정하려면 솔루션 버전 ARN (제외/\$LATEST) 만 지정하고 로 설정합니다syncWithLatestSolutionVersion. False

- `enableMetadataWithRecommendations` 옵션을 `True` 설정합니다. 이렇게 하면 추천 요청 옵션을 통해 항목 데이터세트의 항목 메타데이터를 추천 결과와 함께 포함할 수 있습니다. 이 옵션을 비활성화하려면 `False` 로 설정합니다. 자세한 정보는 [권장 사항의 항목 메타데이터](#) 을 참조하세요.
- `1` (기본값) 로 설정됩니다. `minProvisionedTPS` `minProvisionedTPS`(기본값) 의 경우 `1`로 시작하는 것이 좋습니다. Amazon CloudWatch 지표를 사용하여 사용량을 추적하고 `minProvisionedTPS` 필요에 따라 사용량을 늘리십시오. 자세한 정보는 [초당 최소 프로비저닝 트랜잭션 수 및 자동 스케일링](#) 을 참조하세요.

모든 파라미터의 전체 목록은 [CreateCampaign](#) 섹션을 참조하세요.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_campaign(
    name = 'campaign name',
    solutionVersionArn = 'solution ARN/$LATEST',
    minProvisionedTPS = 1,
    campaignConfig = {"syncWithLatestSolutionVersion": True,
                      "enableMetadataWithRecommendations": True}
)

arn = response['campaignArn']

description = personalize.describe_campaign(campaignArn = arn)['campaign']
print('Name: ' + description['name'])
print('ARN: ' + description['campaignArn'])
print('Status: ' + description['status'])
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { CreateCampaignCommand, PersonalizeClient } from
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({ region: "REGION" });

// set the campaign parameters
```

```

export const createCampaignParam = {
  solutionVersionArn: "SOLUTION_ARN/$LATEST" /* required */,
  name: "NAME" /* required */,
  minProvisionedTPS: 1 /* optional */,
  campaignConfig: { /* optional */
    syncWithLatestSolutionVersion: true,
    enableMetadataWithRecommendations: true,
  },
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new CreateCampaignCommand(createCampaignParam)
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};

run();

```

캠페인 상태가 활성이면 캠페인 준비가 완료된 것입니다. 현재 상태를 확인하려면 전화를 걸어 [DescribeCampaign](#) status 필드가 맞는지 확인하십시오 ACTIVE.

솔루션 버전을 수동으로 재교육하거나 캠페인 설정을 변경하려면 캠페인을 업데이트해야 합니다. 자세한 정보는 [캠페인 업데이트](#)을 참조하세요.

Amazon Personalize는 생성한 캠페인을 [ListCampaigns](#) 나열하는 등의 캠페인 관리 작업을 제공합니다. 캠페인은 [DeleteCampaign](#)을 호출하여 삭제할 수 있습니다. 캠페인을 삭제하더라도 캠페인의 일부인 솔루션 버전은 삭제되지 않습니다.

캠페인을 생성한 후에 캠페인을 사용하여 추천을 생성합니다. 자세한 내용은 [4단계: 추천 받기](#) 단원을 참조하세요.

캠페인 업데이트

재교육된 솔루션 버전을 기존 캠페인과 함께 배포하거나 추천에서 메타데이터를 활성화하는 등 캠페인의 [최소 프로비저닝 TPS](#) 또는 캠페인 구성을 변경하려면 캠페인을 수동으로 업데이트해야 합니다.

사용자 맞춤화-v2, 사용자 맞춤화 또는 Next-Best-Action을 사용하면 Amazon Personalize는 권장 사항에 새 항목 또는 작업을 포함하도록 2시간마다 최신 솔루션 버전 (trainingModeset으로 학습FULL)을 자동으로 업데이트하고 캠페인에서는 업데이트된 솔루션 버전을 자동으로 사용합니다. trainingMode을 FULL로 설정한 상태로 솔루션 버전을 수동으로 재학습시키거나 캠페인의 minProvisionedTPS또는 캠페인 구성을 변경하려는 경우에만 캠페인을 수동으로 업데이트합니다. 자동 업데이트에 대한 자세한 내용은 [자동 업데이트](#) 섹션을 참조하세요.

Amazon Personalize 콘솔, AWS Command Line Interface (AWS CLI) 또는 AWS SDK를 사용하여 캠페인을 수동으로 업데이트합니다.

주제

- [캠페인 업데이트\(콘솔\)](#)
- [캠페인 업데이트\(AWS CLI\)](#)
- [캠페인 업데이트 \(SDK\)AWS](#)

캠페인 업데이트(콘솔)

수동으로 재학습된 솔루션 버전을 배포하거나 캠페인 구성을 변경하려면 캠페인을 업데이트해야 합니다.

캠페인을 업데이트하려면(콘솔)

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인하세요.
2. 업데이트하려는 캠페인의 데이터세트 그룹을 선택합니다.
3. 탐색 창에서 캠페인을 선택합니다.
4. 캠페인 페이지에서 업데이트할 캠페인을 선택합니다.
5. 캠페인 세부 정보 페이지에서 업데이트를 선택합니다.
6. 캠페인 업데이트 페이지에서 내용을 변경합니다. 예를 들어, 재학습된 솔루션 버전을 배포하는 경우 솔루션 버전 ID의 경우 새 솔루션 버전의 식별 번호를 선택합니다.
7. 업데이트를 선택합니다. Personalize가 캠페인을 업데이트하도록 새 솔루션 버전과 변경된 구성을 사용합니다.

캠페인 업데이트(AWS CLI)

새 솔루션 버전을 배포하거나, 캠페인의 [최소 프로비저닝 TPS](#)을 변경하거나, 캠페인 구성을 변경하려면 캠페인을 업데이트해야 합니다. AWS CLI의 새 솔루션 버전을 사용하도록 캠페인을 업데이트하려면 다음 `update-campaign` 명령을 사용합니다.

`campaign arn`을 업데이트하려는 캠페인의 리소스 이름(ARN)으로 바꿉니다. `new solution version arn`을 배포하려는 솔루션 버전으로 바꿉니다.

```
aws personalize update-campaign \
--campaign-arn campaign arn \
--solution-version-arn new solution version arn \
--min-provisioned-tps 1
```

캠페인 업데이트 (SDK)AWS

새 솔루션 버전을 배포하거나, 캠페인의 [최소 프로비저닝 TPS](#)을 변경하거나, 캠페인 구성을 변경하려면 캠페인을 업데이트해야 합니다. Python용 SDK(Boto3) 또는 Java 2.x용 SDK를 사용하여 캠페인을 업데이트하려면 다음 코드를 사용합니다. 파라미터의 전체 목록은 [UpdateCampaign](#) 단원을 참조하세요.

SDK for Python (Boto3)

새 솔루션 버전을 배포하려면 다음 `update_campaign` 메서드를 사용합니다. `campaign arn`을 업데이트하려는 캠페인의 리소스 이름(ARN)으로 바꾼 다음 `new solution version arn`을 새 솔루션 버전 ARN으로 바꾼 다음 필요에 따라 `minProvisionedTPS`을 변경합니다.

```
import boto3

personalize = boto3.client('personalize')

response = personalize.update_campaign(
    campaignArn = 'campaign arn',
    solutionVersionArn = 'new solution version arn',
    minProvisionedTPS = 1,
)

arn = response['campaignArn']

description = personalize.describe_campaign(campaignArn = arn)['campaign']
print('Name: ' + description['name'])
```



```
print('ARN: ' + description['campaignArn'])
print('Status: ' + description['status'])
```

SDK for Java 2.x

새 솔루션 버전을 사용하도록 캠페인을 업데이트하려면 다음 `updateCampaign` 메서드를 사용합니다. Personalize 서비스 클라이언트, 새 솔루션 버전의 리소스 이름(ARN) 및 [최소 프로비저닝 TPS](#)를 파라미터로 전달합니다.

```
public static void updateCampaign(PersonalizeClient personalizeClient,
                                 String campaignArn,
                                 String solutionVersionArn,
                                 Integer minProvisionedTPS) {

    try {
        // build the updateCampaignRequest
        UpdateCampaignRequest updateCampaignRequest =
        UpdateCampaignRequest.builder()
            .campaignArn(campaignArn)
            .solutionVersionArn(solutionVersionArn)
            .minProvisionedTPS(minProvisionedTPS)
            .build();

        // update the campaign
        personalizeClient.updateCampaign(updateCampaignRequest);

        DescribeCampaignRequest campaignRequest = DescribeCampaignRequest.builder()
            .campaignArn(campaignArn)
            .build();

        DescribeCampaignResponse campaignResponse =
        personalizeClient.describeCampaign(campaignRequest);
        Campaign updatedCampaign = campaignResponse.campaign();

        System.out.println("The Campaign status is " + updatedCampaign.status());

    } catch (PersonalizeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

4단계: 추천 받기

리소스에 따라 실시간으로 또는 배치 워크플로우를 통해 추천을 받을 수 있습니다.

- 사용자 지정 리소스를 사용하면 실시간 추천 또는 배치 추천을 받을 수 있습니다. 실시간 추천의 경우 추천을 받기 전에 사용자 지정 캠페인을 만들어야 합니다. 배치 추천의 경우 캠페인을 만들 필요가 없습니다.
- 도메인 데이터세트 그룹에 속한 추천자를 사용하면 실시간 추천만 받을 수 있습니다.

다음에 이어지는 항목에서는 각 추천 유형을 사용하는 방법과 시기에 대해 설명합니다.

주제

- [추천 점수](#)
- [실시간 추천 받기](#)
- [배치 권장 사항 및 사용자 세그먼트\(사용자 지정 리소스\)](#)

추천 점수

Amazon Personalize는 사용자 개인화-v2, 사용자 맞춤화, 개인화된 순위-v2, 개인화된 순위 및 PERSONALIZED_ACTIONS 레시피로 만든 사용자 지정 솔루션을 사용하여 권장 사항의 각 항목에 대한 점수를 포함합니다. 이 점수는 사용자가 다음에 선택할 항목이나 작업에 대해 Amazon Personalize가 보유한 상대적 확실성을 나타냅니다. 점수가 높을수록 확실성이 높아집니다.

- 사용자 개인화-v2 및 사용자 [추천 점수 매기기 작동 방식\(사용자 지정 리소스\)](#) 개인화 점수에 대한 자세한 내용은 을 참조하십시오.
- PERSONALIZED_ACTIONS 레시피 점수에 대한 자세한 내용은 [작업 추천 점수를 매기는 방식](#) 섹션을 참조하십시오.
- 맞춤형 순위-v2 및 맞춤형 순위 추천 점수에 대한 자세한 내용은 을 참조하십시오. [개인 맞춤형 순위 점수 매기기의 작동 방식](#)

배치 추천 작업의 경우 항목 점수는 [추천 점수 매기기 작동 방식\(사용자 지정 리소스\)](#) 및 [개인 맞춤형 순위 점수 매기기의 작동 방식](#)에 설명된 대로 계산됩니다. 점수는 배치 추천 작업의 출력 JSON 파일에서 볼 수 있습니다.

실시간 추천 받기

실시간 추천은 애플리케이션을 사용할 때 요청하여 사용자에게 보여주는 추천입니다. Personalize에서 추천자(도메인 데이터세트 그룹의 경우) 또는 사용자 지정 캠페인을 통해 실시간 추천을 받을 수 있습니다.

- 도메인 추천자의 경우 [the section called “GetRecommendations”](#) 작업을 통해 실시간 추천을 받을 수 있습니다. 또는 Personalize 콘솔을 사용하여 추천자를 테스트할 수 있습니다.
- 사용자 지정 리소스의 경우 캠페인을 지원하는 솔루션 버전을 만드는 데 사용한 레시피에 따라 [the section called “GetRecommendations”](#), [GetActionRecommendations](#) 또는 [the section called “GetPersonalizedRanking”](#) API 작업을 통해 사용자에게 대한 추천을 받을 수 있습니다. 또는 Personalize 콘솔을 사용하여 캠페인을 테스트할 수 있습니다.

실시간 개인 맞춤(예: 가장 적합한 추천 제품 사용 사례 또는 사용자-개인 맞춤 레시피)을 제공하는 도메인 사용 사례 또는 레시피를 사용하는 경우, Amazon Personalize는 사용자가 카탈로그와 상호 작용을 기록할 때 사용자의 가장 최근 활동을 기반으로 추천을 업데이트합니다. 실시간 이벤트 기록 및 개인화에 대한 자세한 내용은 [이벤트 기록](#) 단원을 참조하세요.

추천 항목에 대한 메타데이터를 반환하도록 캠페인을 구성한 경우, [GetRecommendations](#) 또는 [GetPersonalizedRanking](#) API 작업에 포함할 열을 지정할 수 있습니다. 또는 Amazon Personalize 콘솔로 캠페인을 테스트할 때 열을 지정할 수 있습니다.

일부 사용 사례 및 레시피의 경우 요청에 홍보를 지정할 수 있습니다. 홍보는 추천 항목의 구성 가능한 하위 집합에 적용되는 추가 비즈니스 규칙을 정의합니다. 자세한 내용은 [추천의 항목 홍보](#) 단원을 참조하세요.

주제

- [항목 추천 받기](#)
- [작업 추천 받기](#)
- [개인 맞춤형 순위 받기\(사용자 지정 리소스\)](#)
- [컨텍스트 메타데이터로 추천 관련성 높이기](#)

항목 추천 받기

Amazon Personalize 추천자 또는 Amazon Personalize 콘솔, AWS Command Line Interface (AWS CLI) 또는 SDK를 사용하는 사용자 지정 캠페인에서 아이템 추천을 받을 수 있습니다. AWS

Note

PERSONALIZED_RANKING 사용자 지정 레시피를 사용한 경우 [개인 맞춤형 순위 받기\(사용자 지정 리소스\)](#) 단원을 참조하세요.

주제

- [추천 점수 매기기 작동 방식 \(사용자 지정 리소스\)](#)
- [추천 이유 \(사용자 맞춤화-v2\)](#)
- [항목 추천 받기\(콘솔\)](#)
- [항목 추천 받기\(AWS CLI\)](#)
- [항목 추천 \(SDK\) 가져오기AWS](#)
- [추천의 항목 홍보](#)

추천 점수 매기기 작동 방식 (사용자 지정 리소스)

Amazon Personalize는 사용자 개인화-v2 및 사용자 개인화 레시피를 사용하여 사용자의 상호 작용 데이터 및 메타데이터를 기반으로 항목에 대한 점수를 생성합니다. 이들 점수는 사용자가 다음에 해당 항목과 상호작용할 것인지 여부에 대해 Personalize가 갖고 있는 상대적 확실성을 나타냅니다. 점수가 높을수록 확실성이 높아집니다.

Note

Personalize는 도메인 추천자 또는 유사-항목, SIMS 또는 인기도-집계 레시피에 대한 점수는 표시하지 않습니다. 개인 맞춤형-순위 추천 점수에 대한 자세한 내용은 [개인 맞춤형 순위 점수 매기기의 작동 방식](#) 단원을 참조하세요.

Amazon Personalize는 0에서 1까지의 척도로 항목 간 점수를 생성합니다 (둘 다 포함). 사용자 개인화-v2를 사용하면 Amazon Personalize는 항목의 하위 집합에 대한 점수를 생성합니다. 사용자 맞춤 설정을 사용하면 Amazon Personalize는 카탈로그의 모든 항목에 점수를 매깁니다.

User-Personalization-v2를 사용하고 권장 사항에 필터를 적용하는 경우 필터가 제거하는 권장 사항의 수에 따라 Amazon Personalize에서 자리 표시자 항목을 추가할 수 있습니다. 이는 추천 요청을 충족하기 위한 것입니다. numResults 이러한 항목은 상호 작용 데이터의 양을 기준으로 필터 기준을 충족하는 인기 항목입니다. 해당 항목에는 사용자에게 대한 관련성 점수가 없습니다.

사용자 맞춤설정-v2와 사용자 개인화 모두에서 모든 점수의 합계는 1입니다. 예를 들어 사용자에게 영화 추천을 받고 있는데 항목 데이터세트와 상호작용 데이터세트에 세 편의 영화가 나오는 경우 점수는, 일 수 있습니다. 0.6 0.3 0.1 마찬가지로 인벤토리에 10,000개의 영화가 있는 경우 가장 높은 점수를 받은 영화의 점수는 매우 적을 수 있지만 (평균 점수는 다음과 같음. 001), 점수는 상대적이므로 권장 사항은 여전히 유효합니다.

수학 용어로, 각 사용자 항목 쌍 (u, i) 의 점수는 다음 공식에 따라 계산됩니다. 여기서 \exp 는 지수 함수, u 와 w_i 는 각각 사용자 및 항목 i 임베딩이며 그리스 문자 σ () 는 점수가 있는 모든 항목에 대한 합계를 나타냅니다.

$$\text{score}(u, i) = \frac{\exp(\bar{w}_u^\top w_i)}{\sum_j \exp(\bar{w}_u^\top w_j)}$$

추천 이유 (사용자 맞춤화-v2)

User-Personalization-v2를 사용하는 경우 각 권장 항목에는 해당 항목이 권장 사항에 포함된 이유 목록이 포함될 수 있습니다. 가능한 원인은 다음과 같습니다.

- 프로모션 아이템 — 추천 요청에 적용한 프로모션의 일부로 아이템이 포함되었음을 나타냅니다.
- 탐색 — 항목이 탐색에 포함되었음을 나타냅니다. 탐색의 경우 권장 사항에는 상호 작용 데이터가 적거나 사용자와 관련성이 낮은 항목이 포함됩니다. [탐색에 대한 자세한 내용은 탐색을 참조하십시오.](#)
- 인기 항목 — 항목이 자리 표시자 인기 항목으로 포함되었음을 나타냅니다. 필터를 사용하는 경우 필터가 제거하는 추천 개수에 따라 Amazon Personalize에서 추천 요청에 맞는 자리 표시자 항목을 추가할 수 있습니다. numResults 이러한 항목은 상호 작용 데이터를 기반으로 하는 인기 품목으로, 필터 기준을 충족합니다. 해당 항목에는 사용자에게 대한 관련성 점수가 없습니다.

항목 추천 받기(콘솔)

Personalize 콘솔에서 추천을 받으려면 추천자(도메인 데이터세트 그룹) 또는 사용자 지정 캠페인의 세부 정보 페이지에서 요청 정보를 제공합니다.

추천을 받으려면

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 사용 중인 캠페인 또는 추천자가 포함된 데이터세트 그룹을 선택합니다.

3. 탐색 창에서 캠페인이나 추천자를 선택합니다.
4. 대상 캠페인 또는 추천자를 선택합니다.
5. 캠페인의 경우 캠페인 결과 테스트에서, 사용한 레시피를 기반으로 추천 요청 세부 정보를 입력합니다. 추천자의 경우 추천자 테스트를 선택하고 사용 사례를 기반으로 추천 요청 세부 정보를 입력합니다.

로그인하기 전에 사용자(익명 사용자)에 대한 이벤트를 기록한 경우, 해당 이벤트의 sessionId를 userId인 것처럼 제공하여 해당 사용자에 대한 추천을 받을 수 있습니다. 익명 사용자의 이벤트 기록에 대한 자세한 내용은 [익명 사용자를 위한 이벤트 기록](#) 단원을 참조하세요.

6. 필터를 선택합니다. 자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 단원을 참조하세요.
7. 컨텍스트 메타데이터를 사용하는 경우 각 컨텍스트에 대한 데이터를 제공합니다. 각 컨텍스트에 대해 키에 메타데이터 필드를 입력합니다. 값에 컨텍스트 데이터를 입력합니다. 자세한 정보는 [컨텍스트 메타데이터로 추천 관련성 높이기](#)을 참조하세요.
8. 캠페인 또는 추천자에 대한 추천에서 메타데이터를 활성화한 경우, 항목 데이터 세트 열에서 추천 결과에 포함할 메타데이터 열을 선택합니다. 캠페인의 메타데이터 활성화에 대한 자세한 내용은 [권장 사항의 항목 메타데이터](#) 섹션을 참조하세요. 추천자의 메타데이터 활성화에 대한 자세한 내용은 [추천에서 메타데이터 활성화](#) 섹션을 참조하세요.
9. 항목의 하위 세트를 홍보하려면 홍보 필드를 완성합니다(선택 사항). 자세한 내용은 [추천의 항목 홍보](#) 단원을 참조하세요.
10. 추천 받기를 선택합니다. 사용자의 상위 25개 추천 항목이 포함된 표가 표시됩니다. User- Personalization-v2를 사용하는 경우 각 권장 항목에는 해당 항목이 권장 사항에 포함된 이유 목록이 포함됩니다. 자세한 정보는 [추천 이유 \(사용자 맞춤화-v2\)](#)을 참조하세요.

항목 추천 받기(AWS CLI)

다음 코드 샘플은 AWS CLI를 사용하여 항목 추천을 받는 방법을 다양하게 보여줍니다.

주제

- [항목 추천 받기](#)
- [추천에 항목 메타데이터 포함](#)

항목 추천 받기

캠페인에서 추천을 받으려면 다음 코드를 사용합니다. 추천자로부터 추천을 받으려면 campaign-arn 파라미터를 recommender-arn으로 바꿉니다.

추천을 받을 사용자의 ID와 캠페인 또는 추천자의 리소스 이름(ARN)을 지정합니다. 사용자에게 대한 상위 10개 추천 항목의 목록이 표시됩니다. User-Personalization-v2를 사용하는 경우 각 권장 항목에는 해당 항목이 권장 사항에 포함된 이유 목록이 포함됩니다. 자세한 정보는 [추천 이유 \(사용자 맞춤화-v2\)](#)을 참조하세요.

추천 항목 수를 변경하려면 numResults의 값을 변경합니다. 기본값은 25개 항목입니다. 최대 항목 수는 500개입니다. RELATED_ITEMS 레시피를 사용하여 캠페인을 지원하는 솔루션 버전을 학습시킨 경우 user-id파라미터를 item-id로 대체하고 항목 ID를 지정합니다.

로그인하기 전에 사용자(익명 사용자)에 대한 이벤트를 기록한 경우, 해당 이벤트의 sessionId를 userId인 것처럼 제공하여 해당 사용자에게 대한 추천을 받을 수 있습니다. 익명 사용자의 이벤트 기록에 대한 자세한 내용은 [익명 사용자를 위한 이벤트 기록](#) 단원을 참조하세요.

```
aws personalize-runtime get-recommendations \
--campaign-arn campaign arn \
--user-id User ID \
--num-results 10
```

추천에 항목 메타데이터 포함

캠페인 또는 추천자에 대한 추천에서 메타데이터를 활성화한 경우, 응답에 포함할 항목 데이터 세트 메타데이터 열을 지정할 수 있습니다. 캠페인의 메타데이터 활성화에 대한 자세한 내용은 [권장 사항의 항목 메타데이터](#) 섹션을 참조하세요. 추천자의 메타데이터 활성화에 대한 자세한 내용은 [추천에서 메타데이터 활성화](#) 섹션을 참조하세요.

다음 코드 샘플은 추천 요청의 일부로 메타데이터 열을 지정하는 방법을 보여줍니다.

```
aws personalize-runtime get-recommendations \
--campaign-arn campaign arn \
--user-id User ID \
--num-results 10 \
--metadata-columns "{\"ITEMS\": [\"columnA\", \"columnB\"]}"
```

항목 추천 (SDK) 가져오기AWS

다음 코드 샘플은 AWS SDK를 사용하여 항목 추천을 받는 방법의 다양한 변형을 보여줍니다.

주제

- [항목 추천 받기](#)
- [추천에 항목 메타데이터 포함](#)

항목 추천 받기

다음 코드는 캠페인으로부터 사용자에게 대한 Personalize 추천을 받는 방법을 보여줍니다. 추천자로부터 추천을 받으려면 `campaignArn` 파라미터를 `recommenderArn`으로 바꿉니다.

추천을 받을 사용자의 ID와 캠페인 또는 추천자의 리소스 이름(ARN)을 지정합니다. 사용자에게 대한 상위 10개 추천 항목의 목록이 표시됩니다. User-Personalization-v2를 사용하는 경우 각 권장 항목에는 해당 항목이 권장 사항에 포함된 이유 목록이 포함됩니다. 자세한 정보는 [추천 이유 \(사용자 맞춤화-v2\)](#)을 참조하세요.

추천 항목 수를 변경하려면 `numResults`의 값을 변경합니다. 기본값은 25개 항목입니다. 최대 항목 수는 500개입니다. RELATED_ITEMS 레시피를 사용하여 캠페인을 지원하는 솔루션 버전을 학습시킨 경우 `userId`파라미터를 `itemId`로 대체하고 항목 ID를 지정합니다.

캠페인 또는 추천자에 대한 추천에서 메타데이터를 활성화한 경우, 응답에 포함할 항목 데이터 세트 메타데이터 열을 지정할 수 있습니다. 코드 샘플에 대해서는 [추천에 항목 메타데이터 포함](#) 단원을 참조하세요. 메타데이터 활성화에 대한 자세한 내용은 [권장 사항의 항목 메타데이터](#) 섹션을 참조하세요.

로그인하기 전에 사용자(익명 사용자)에 대한 이벤트를 기록한 경우, 해당 이벤트의 `sessionId`를 `userId`인 것처럼 제공하여 해당 사용자에게 대한 추천을 받을 수 있습니다. 익명 사용자의 이벤트 기록에 대한 자세한 내용은 [익명 사용자를 위한 이벤트 기록](#) 단원을 참조하세요.

SDK for Python (Boto3)

```
import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_recommendations(
    campaignArn = 'Campaign ARN',
    userId = 'User ID',
    numResults = 10
)

print("Recommended items")
for item in response['itemList']:
    print (item['itemId'])
```

SDK for Java 2.x

```
public static void getRecs(PersonalizeRuntimeClient personalizeRuntimeClient,
    String campaignArn, String userId) {
```



```

    try {
        GetRecommendationsRequest recommendationsRequest =
GetRecommendationsRequest.builder()
            .campaignArn(campaignArn)
            .numResults(20)
            .userId(userId)
            .build();

        GetRecommendationsResponse recommendationsResponse =
personalizeRuntimeClient
            .getRecommendations(recommendationsRequest);
        List<PredictedItem> items = recommendationsResponse.itemList();
        for (PredictedItem item : items) {
            System.out.println("Item Id is : " + item.itemId());
            System.out.println("Item score is : " + item.score());
        }

    } catch (AwsServiceException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

```

SDK for JavaScript v3

```

// Get service clients module and commands using ES6 syntax.
import { GetRecommendationsCommand } from
    "@aws-sdk/client-personalize-runtime";

import { personalizeRuntimeClient } from "./libs/personalizeClients.js";
// Or, create the client here.
// const personalizeRuntimeClient = new PersonalizeRuntimeClient({ region:
    "REGION"});

// Set the recommendation request parameters.
export const getRecommendationsParam = {
    campaignArn: 'CAMPAIGN_ARN', /* required */
    userId: 'USER_ID', /* required */
    numResults: 15 /* optional */
}

export const run = async () => {

```

```

try {
  const response = await personalizeRuntimeClient.send(new
  GetRecommendationsCommand(getRecommendationsParam));
  console.log("Success!", response);
  return response; // For unit tests.
} catch (err) {
  console.log("Error", err);
}
};
run();

```

추천에 항목 메타데이터 포함

캠페인 또는 추천자에 대한 추천에서 메타데이터를 활성화한 경우, 응답에 포함할 항목 데이터 세트 메타데이터 열을 지정할 수 있습니다. 캠페인의 메타데이터 활성화에 대한 자세한 내용은 [권장 사항의 항목 메타데이터](#) 섹션을 참조하세요. 추천자의 메타데이터 활성화에 대한 자세한 내용은 [추천에서 메타데이터 활성화](#) 섹션을 참조하세요.

다음 코드 샘플은 추천 요청의 일부로 메타데이터 열을 지정하는 방법을 보여줍니다.

```

import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_recommendations(
  campaignArn = 'Campaign ARN',
  userId = 'User ID',
  numResults = 10
  metadataColumns = {
    "ITEMS": ['columnNameA', 'columnNameB']
  }
)

print("Recommended items")
for item in response['itemList']:
  print(item['itemId'])
  print(item['metadata'])

```

추천의 항목 홍보

모든 도메인 사용 사례와 일부 맞춤 레시피로 실시간 추천을 받을 때 프로모션을 지정할 수 있습니다.

홍보는 추천 항목의 구성 가능한 하위 집합에 적용되는 추가 비즈니스 규칙을 정의합니다. 예를 들어 스트리밍 앱이 있는데 자체 프로그램과 영화를 홍보하면서 관련 타이틀도 추천하고 싶을 수 있습니다. 홍보를 통해 특정 비율의 추천 항목이 사내 카테고리에 속하도록 지정할 수 있습니다. 나머지 추천 항목은 레시피와 요청 필터를 기반으로 한 관련 추천 항목으로 계속 제공될 것입니다.

홍보를 적용하려면 추천 요청에 다음을 지정합니다.

- [홍보 필터를 적용할 추천 항목의 비율](#).
- [홍보 기준을 지정하는 필터](#). 자세한 내용은 [홍보 필터](#) 단원을 참조하세요.

추천 응답에서 홍보 대상 항목은 다른 추천 항목을 기준으로 무작위로 배치되지만 다른 홍보 대상 항목을 기준으로 정렬된 순서로 배치됩니다. 레시피에 따라 홍보에 포함되지 않은 추천 항목은 사용자와의 관련성, 인기도 또는 유사도를 기준으로 정렬됩니다. 홍보 기준을 충족하는 항목이 충분하지 않은 경우, 결과에는 최대한 많은 홍보 대상 항목이 포함될 것입니다.

Amazon Personalize 콘솔, AWS Command Line Interface (AWS CLI) 또는 AWS SDK를 사용하여 권장 사항에 프로모션을 적용할 수 있습니다.

주제

- [홍보를 지원하는 사용 사례 및 레시피](#)
- [홍보 필터](#)
- [새 항목 홍보](#)
- [항목 홍보\(콘솔\)](#)
- [항목 홍보\(AWS CLI\)](#)
- [프로모션 아이템 \(AWS SDK\)](#)

홍보를 지원하는 사용 사례 및 레시피

모든 사용 사례는 홍보를 지원합니다. 홍보를 지원하는 사용자 지정 레시피는 다음과 같습니다.

- [USER_PERSONALIZATION](#) 레시피
- [RELATED_ITEMS](#) 레시피
- [POPULAR_ITEMS](#) 레시피

홍보 필터

추천 요청에 홍보를 적용할 때는 홍보 기준을 지정하는 필터를 선택합니다. 기존 데이터세트를 사용하거나 데이터세트를 새로 만들어도 됩니다. 홍보용 필터도 Personalize의 다른 필터와 마찬가지로 생성하고 관리합니다. 필터를 생성하고 관리하는 방법에 대한 자세한 내용은 [결과 필터링](#) 단원을 참조하세요.

홍보 필터와 홍보 외부에서 선택하는 필터(요청 필터) 간의 유일한 차이점은 Personalize가 이들 필터를 적용하는 방식입니다. 홍보 필터는 홍보 대상 항목에만 적용되고, 요청 필터는 나머지 추천 아이템에만 적용됩니다. 요청 필터와 홍보 필터를 지정하고 홍보 대상 항목에 두 필터를 모두 적용하려는 경우 홍보 필터의 표현식에 두 표현식이 모두 포함되어야 합니다. 두 표현식을 결합하는 방법은 사용하는 데이터세트에 따라 달라집니다. 필터 표현식, 규칙, 생성 방법에 대한 자세한 내용은 [필터 표현식](#) 단원을 참조하세요.

필터 표현식 예제

다음 표현식에는 "사내"범주의 항목만 포함됩니다. 추천에서 자체 콘텐츠를 홍보하려는 경우 이 표현식을 사용할 수 있습니다.

```
INCLUDE ItemID WHERE Items.OWNER IN ("in-house")
```

다음 표현식에는 지정한 타임스탬프보다 최근에 생성된 항목만 포함됩니다. 이 표현식을 사용하여 권장 사항의 새 항목을 홍보할 수 있습니다.

```
INCLUDE ItemID WHERE Items.CREATION_TIMESTAMP > $DATE
```

다음 표현식은 홍보 대상 항목에 요청 필터를 적용하는 방법을 보여줍니다. 여기에는 홍보 대상 항목으로 사용 가능한 의류 품목만 포함됩니다. 이 시나리오에서는 요청 필터 표현식에도 `Items.AVAILABLE IN ("True")`이 사용되므로 모든 추천은 사용 가능한 항목에 적용됩니다.

```
INCLUDE ItemID WHERE Items.CATEGORY IN ("clothing") AND Items.AVAILABLE IN ("True")
```

전체 필터 목록 예제를 더 살펴보려면 [필터 표현식 예제](#) 단원을 참조하세요.

새 항목 홍보

를 사용하는 경우 Amazon Personalize는 사용자에게 가장 관련성이 높은 항목을 추천하고 상호 작용 데이터가 있는 기존 항목을 더 자주 추천합니다. [사용자 맞춤화-v2 레시피](#) 권장 사항에 새 항목이 포함되도록 하려면 생성 타임스탬프를 기반으로 항목이 포함된 추천 요청에 프로모션을 적용할 수 있습니다.

프로모션을 아직 사용하지 않은 경우 필터 표현식을 사용하여 특정 날짜 이후에 생성된 항목을 프로모션할 수 있습니다.

```
INCLUDE ItemID WHERE Items.CREATION_TIMESTAMP > $DATE
```

이미 프로모션을 사용하고 있다면 프로모션과 새 아이템 조건 설명을 모두 연결하는 표현식을 생성하십시오.

```
INCLUDE ItemID WHERE Items.CATEGORY IN ("clothing") OR Items.CREATION_TIMESTAMP > $DATE
```

항목 홍보(콘솔)

Personalize 콘솔을 사용하여 추천의 특정 항목을 홍보하려면 필터를 생성한 다음 추천 요청에 홍보 세부 정보를 제공합니다. 다른 필드에 대한 자세한 내용은 [항목 추천 받기\(콘솔\)](#) 단원을 참조하세요.

추천의 항목을 홍보하려면

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 사용 중인 캠페인 또는 추천자가 포함된 데이터세트 그룹을 선택합니다.
3. 아직 만들지 않았다면 홍보 기준을 지정하는 필터를 만듭니다. 요청 필터를 생성하는 것과 동일한 방식으로 홍보용 필터를 생성합니다. 필터를 생성하고 관리하는 방법에 대한 자세한 내용은 [결과 필터링](#) 단원을 참조하세요.
4. 탐색 창에서 추천자나 캠페인을 선택합니다.
5. 대상 캠페인 또는 추천자를 선택합니다.
6. 캠페인의 경우 캠페인 결과 테스트에서, 사용한 레시피를 기반으로 추천 요청 세부 정보를 입력합니다. 추천자의 경우 추천자 테스트를 선택하고 추천 요청 세부 정보를 입력합니다.
7. (선택 사항) 요청에 사용할 필터를 선택합니다. 이 필터는 비 홍보 대상 항목에만 적용됩니다. 필터를 생성하고 관리하는 방법에 대한 자세한 내용은 [결과 필터링](#) 단원을 참조하세요.
8. 컨텍스트 메타데이터를 사용하는 경우 각 컨텍스트에 대한 데이터를 제공합니다. 각 컨텍스트에 대해 키에 메타데이터 필드를 입력합니다. 값에 컨텍스트 데이터를 입력합니다. 자세한 내용은 [컨텍스트 메타데이터로 추천 관련성 높이기](#) 단원을 참조하세요.
9. 홍보에 다음을 지정합니다.
 - **홍보 대상 항목 비율:** 홍보를 적용할 추천 아이템의 비율을 입력합니다.
 - **필터:** 홍보 기준을 지정하는 필터를 선택합니다. 이 필터는 7단계에서 지정한 요청 필터 대신 홍보 대상 항목에 적용됩니다.

- 필터 파라미터: 홍보에서 자리 표시자 파라미터가 포함된 필터를 사용하는 경우 각 파라미터에 대해 값을 입력하여 필터 기준을 설정합니다. 한 파라미터에 여러 값을 사용하려면 각 값을 쉼표로 구분합니다.
10. 추천 받기를 선택합니다. 사용자의 상위 25개 추천 항목이 포함된 표가 표시됩니다. 홍보 대상 항목 열에는 홍보 때문에 해당 항목이 포함되었는지 여부가 표시됩니다. 홍보 대상 항목은 다른 추천 항목을 기준으로 무작위로 배치되지만 다른 홍보 대상 항목을 기준으로 정렬된 순서로 배치됩니다. 사용 사례 또는 레시피에 따라 홍보에 포함되지 않은 추천 항목은 사용자와의 관련성, 인기도 또는 유사성을 기준으로 정렬됩니다. 홍보 기준을 충족하는 항목이 충분하지 않은 경우, 결과에는 최대한 많은 홍보 대상 항목이 포함될 것입니다.

항목 홍보(AWS CLI)

다음 코드는 AWS CLI 및 사용자 지정 캠페인을 사용하여 권장 사항으로 항목을 홍보하는 방법을 보여줍니다. 추천자를 사용하여 항목을 홍보하려면 `campaign-arn` 파라미터를 `recommender-arn`로 바꾸고 추천자의 Amazon 리소스 이름(ARN)을 지정합니다. 홍보 필드에 다음을 지정합니다.

- 이름: 홍보 이름을 지정합니다. 추천 응답은 이름을 사용하여 홍보 대상 항목을 식별합니다.
- percent-promoted-items: 프로모션을 적용할 추천 아이템의 비율입니다. 이 예제에서는 항목의 50%가 홍보 대상 항목이 됩니다.
- filterArn: 홍보 기준을 정의하는 필터의 Amazon 리소스 이름(ARN)을 지정합니다. 자세한 내용은 [홍보 필터](#) 단원을 참조하세요.
- 파라미터 이름 및 값: 필터 표현식에 파라미터가 있는 경우 파라미터 이름(대소문자 구분)과 값을 제공합니다. 예를 들어 필터 표현식에 `$GENRE` 파라미터가 있는 경우 `GENRE`를 키로 입력하고 장르(예: 코미디)를 값으로 입력합니다. 값이 여러 개인 경우 쉼표로 구분합니다. 를 사용할 때는 각 값에 대해 해당 문자를 사용하여 따옴표와 / 문자를 모두 이스케이프 처리해야 합니다. AWS CLI/ 다음 코드 예제는 값의 형식을 지정하는 방법을 보여줍니다.

코드는 요청 필터와 홍보 필터를 모두 사용하는 방법을 보여줍니다. 홍보 필터는 홍보 대상 항목에만 적용되고, 요청 필터는 나머지 추천 아이템에만 적용됩니다. 자세한 내용은 [홍보 필터](#) 단원을 참조하세요.

추가 필드에 대한 자세한 내용은 [항목 추천 \(SDK\) 가져오기 AWS](#) 및 [상황별 메타데이터를 사용하여 개인 맞춤형 순위 받기](#) 단원을 참조하세요.

```
aws personalize-runtime get-recommendations \
  --campaign-arn CampaignArn \
  --user-id 1 \
```

```
--num-results 10 \
--filter-arn RequestFilterArn \
--filter-values '{
  "RequestFilterParameterName": "\"value\"",
  "RequestFilterParameterName": "\"value1\", \"value2\", \"value3\""
}' \
--promotions "[{
  \"name\": \"promotionName\",
  \"percentPromotedItems\": 50,
  \"filterArn\": \"PromotionFilterARN\",
  \"filterValues\": {\"PromotionParameterName\": \"\\\"value1, value2\\\"\"}
}]"
```

추천 항목 목록이 표시됩니다. 홍보 대상 항목은 다른 추천 항목을 기준으로 무작위로 배치되지만 다른 홍보 대상 항목을 기준으로 정렬된 순서로 배치됩니다. 레시피에 따라 홍보에 포함되지 않은 추천 항목은 사용자와의 관련성, 인기도 또는 유사도를 기준으로 정렬됩니다. 홍보 기준을 충족하는 항목이 충분하지 않은 경우, 결과에는 최대한 많은 홍보 대상 항목이 포함될 것입니다.

```
{
  "itemList": [
    {
      "itemId1": "123",
      "score": .0117211,
      "promotionName": "promotionName"
    },
    {
      "itemId2": "456",
      "score": .0077976
    },
    {
      "itemId3": "789",
      "score": .0067171
    },
    .....
  ]
}
```

프로모션 아이템 (AWS SDK)

다음 코드는 Python용 SDK(Boto3) 및 Java 2.x용 SDK와 사용자 지정 캠페인을 사용하여 추천의 항목을 홍보하는 방법을 보여줍니다. 추천을 사용하여 항목을 홍보하려면 `campaignArn` 파라미터를 `recommenderArn`로 대체하고 추천자의 Amazon 리소스 이름(ARN)을 지정합니다. 홍보 필드에 다음을 지정합니다.

- 이름: 홍보 이름을 지정합니다. 추천 응답에는 홍보 대상 항목을 식별할 수 있는 이름이 포함됩니다.
- percentPromotedItems: 프로모션을 적용할 추천 아이템의 비율입니다.
- promotionFilterARN: 홍보 기준을 정의하는 필터의 Amazon 리소스 이름(ARN)입니다. 자세한 내용은 [홍보 필터](#) 단원을 참조하세요.
- 모든 파라미터 이름 및 값: 필터 표현식에 파라미터가 있는 경우 필터 표현식의 각 파라미터에 대해 파라미터 이름(대소문자 구분)과 값을 제공합니다. 예를 들어 필터 표현식에 \$GENRE 파라미터가 있는 경우 "GENRE"를 키로 제공하고 장르(예: "\"Comedy\"")를 값으로 제공합니다. 값이 여러 개인 경우 쉼표로 구분합니다. 예를 들어 "\"comedy\"\", \"drama\"\", \"horror\"\"입니다.

다음 코드는 요청 필터와 홍보 필터를 모두 사용하는 방법을 보여줍니다. 홍보 필터는 홍보 대상 항목에만 적용되고, 요청 필터는 나머지 추천 아이템에만 적용됩니다. 자세한 내용은 [홍보 필터](#) 단원을 참조하세요.

추가 필드에 대한 자세한 내용은 [항목 추천 \(SDK\) 가져오기AWS](#) 및 [상황별 메타데이터를 사용하여 개인 맞춤형 순위 받기](#) 단원을 참조하세요.

SDK for Python (Boto3)

```
import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_recommendations(
    campaignArn = "CampaignARN",
    userId = '1',
    numResults = 10,
    filterArn = 'RequestFilterARN',
    filterValues = {
        "RequestFilterParameterName": "\"value1\"",
        "RequestFilterParameterName": "\"value1\", \"value2\", \"value3\""
        ....
    },
    promotions = [{
        "name" : "promotionName",
        "percentPromotedItems" : 50,
        "filterArn": "promotionFilterARN",
        "filterValues": {
            "PromotionParameterName": "\"Value1\", \"Value2\""
            ...
        }
    }
]
```



```

    ]]
  )

  print("Recommended items")
  for item in response['itemList']:
    print (item['itemId'])
    if ("promotionName" in item):
      print(item['promotionName'])

```

SDK for Java 2.x

```

public static void getRecommendationsWithPromotedItems(PersonalizeRuntimeClient
personalizeRuntimeClient,
                                                         String campaignArn,
                                                         String userId,
                                                         String requestFilterArn,
                                                         String requestParameterName,
                                                         String requestParameterValue1,
                                                         String requestParameterValue2,
                                                         String promotionName,
                                                         int percentPromotedItems,
                                                         String promotionFilterArn,
                                                         String promotionParameterName,
                                                         String promotionParameterValue1,
                                                         String promotionParameterValue2) {

  try {

    Map<String, String> promotionFilterValues = new HashMap<>();

    promotionFilterValues.put(promotionParameterName, String.format("\'%1$s\'",
\''%2$s\'",
    promotionParameterValue1, promotionParameterValue2));

    Promotion newPromotion = Promotion.builder()
      .name(promotionName)
      .percentPromotedItems(percentPromotedItems)
      .filterArn(promotionFilterArn)
      .filterValues(promotionFilterValues)
      .build();

    List<Promotion> promotionList = new List<>();

```

```

    promotionsList.add(newPromotion);

    Map<String, String> requestfilterValues = new HashMap<>();

    requestfilterValues.put(requestParameterName, String.format("\'%1$s\'",\'%2$s
\'",
        requestParameterValue1, requestParameterValue2));

    GetRecommendationsRequest recommendationsRequest =
    GetRecommendationsRequest.builder()
        .campaignArn(campaignArn)
        .numResults(20)
        .userId(userId)
        .filterArn(requestFilterArn)
        .filterValues(requestFilterValues)
        .promotions(promotionList)
        .build();

    GetRecommendationsResponse recommendationsResponse =
    personalizeRuntimeClient.getRecommendations(recommendationsRequest);
    List<PredictedItem> items = recommendationsResponse.itemList();

    for (PredictedItem item: items) {
        System.out.println("Item Id is : "+item.itemId());
        System.out.println("Item score is : "+item.score());
        System.out.println("Promotion name is : "+item.promotionName());
    }
} catch (PersonalizeRuntimeException e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
}
}

```

SDK for JavaScript v3

```

// Get service clients and commands using ES6 syntax.
import { GetRecommendationsCommand, PersonalizeRuntimeClient } from
"@aws-sdk/client-personalize-runtime";

// create personalizeRuntimeClient.
const personalizeRuntimeClient = new PersonalizeRuntimeClient({
    region: "REGION",
});

```

```
// set recommendation request param
export const getRecommendationsParam = {
  campaignArn: "CAMPAIGN_ARN", /* required */
  userId: "USER_ID", /* required */
  numResults: 25, /* optional */
  filterArn: "FILTER_ARN", /* provide if you are applying a custom filter */
  filterValues: {
    "PARAM_NAME": "\"PARAM_VALUE\"" /* provide if your filter has a placeholder
parameter */
  },
  promotions: [
    {
      name: "PROMOTION_NAME", /* specify the name of the promotion. The
recommendation response includes the name to identify promoted items. */
      percentPromotedItems: 50, /* the percentage of recommended items to apply the
promotion to. */
      filterArn:
        "PROMOTION_FILTER_ARN", /* the Amazon Resource Name (ARN) of the filter that
defines the promotion criteria. */
      filterValues: {
        "PARAM_NAME": "\"PARAM_VALUE\"" /* provide if your promotion filter has a
placeholder parameter */
      },
    },
  ],
];

export const run = async () => {
  try {
    const response = await personalizeRuntimeClient.send(new
GetRecommendationsCommand(getRecommendationsParam));
    console.log("Success!", "\nItems are: ");
    response.itemList.forEach(element => console.log(element.itemId))
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};

run();
```

추천 항목 목록이 표시됩니다. 홍보 대상 항목은 다른 추천 항목을 기준으로 무작위로 배치되지만 다른 홍보 대상 항목을 기준으로 정렬된 순서로 배치됩니다. 레시피에 따라 홍보에 포함되지 않은 추천 항목은 사용자와의 관련성, 인기도 또는 유사도를 기준으로 정렬됩니다. 홍보 기준을 충족하는 항목이 충분하지 않은 경우, 결과에는 최대한 많은 홍보 대상 항목이 포함될 것입니다.

```
{
  "itemList": [
    {
      "itemId1": "123",
      "score": .0117211,
      "promotionName": "promotionName"
    },
    {
      "itemId2": "456",
      "score": .0077976
    },
    {
      "itemId3": "789",
      "score": .0067171
    },
    .....
  ]
}
```

작업 추천 받기

PERSONALIZED_ACTIONS 레시피를 사용하면 캠페인에서 실시간으로 작업 추천을 받을 수 있습니다. Amazon Personalize 콘솔, AWS Command Line Interface(AWS CLI) 또는 AWS SDK를 사용하여 작업 추천을 받을 수 있습니다.

주제

- [작업 추천 점수를 매기는 방식](#)
- [작업 추천 받기\(콘솔\)](#)
- [작업 추천 받기\(AWS CLI\)](#)
- [작업 추천 받기\(AWS SDK\)](#)

작업 추천 점수를 매기는 방식

Amazon Personalize는 차선택-작업 레시피를 통해 사용자가 해당 작업과 상호 작용할 가능성을 기반으로 작업에 대한 점수를 생성합니다. 점수는 0에서 1.0 사이일 수 있습니다. 1.0에 가까울수록 사용자가 작업과 상호 작용할 가능성이 높아집니다.

작업 상호 작용 데이터를 가져오지 않은 경우 모든 추천 작업의 점수는 0.0입니다. Amazon Personalize가 탐색의 일환으로 작업을 추천하는 경우 해당 항목의 점수는 0.0입니다. Amazon Personalize는 탐색을 사용하여 작업 상호 작용 데이터 없이 작업을 추천합니다. 탐색에 대한 자세한 내용은 [탐색](#) 단원을 참조하세요.

작업 추천 받기(콘솔)

Amazon Personalize 콘솔에서 작업 추천을 받으려면 사용자 지정 캠페인의 세부 정보 페이지에서 요청 정보를 제공하면 됩니다.

작업 추천을 받으려면

1. <https://console.aws.amazon.com/personalize/home> 에서 Amazon Personalize 콘솔을 열고 계정에 로그인합니다.
2. 사용 중인 캠페인이 포함된 데이터 세트 그룹을 선택합니다.
3. 탐색 창의 사용자 지정 리소스에서 캠페인을 선택합니다.
4. 대상 캠페인을 선택합니다.
5. 테스트 캠페인 결과에서 추천 요청 세부 정보를 입력합니다.

로그인하기 전에 사용자(익명 사용자)에 대한 이벤트를 기록한 경우, 해당 이벤트의 sessionId를 userId인 것처럼 제공하여 해당 사용자에 대한 추천을 받을 수 있습니다. 익명 사용자의 이벤트 기록에 대한 자세한 내용은 [익명 사용자를 위한 이벤트 기록](#) 단원을 참조하세요.

6. (선택 사항) 필터를 선택합니다. 자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 섹션을 참조하세요.
7. 추천 받기를 선택합니다. 사용자의 상위 5개 추천 작업이 포함된 표가 나타납니다.

작업 추천 받기(AWS CLI)

캠페인에서 작업 추천을 받으려면 다음 코드를 사용합니다. 추천을 받을 사용자의 ID와 캠페인의 Amazon 리소스 이름(ARN)을 지정합니다.

추천 작업 수를 변경하려면 numResults의 값을 변경합니다. 기본값은 5개 작업입니다. 최대 값은 100개 작업입니다.

사용자 지정 기준에 따라 작업 추천을 필터링하려면 필터를 생성하여 `get-action-recommendations` 작업에 적용할 수 있습니다. 자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 섹션을 참조하세요.

로그인하기 전에 사용자(익명 사용자)에 대한 이벤트를 기록한 경우, 해당 이벤트의 `sessionId`를 `userId`인 것처럼 제공하여 해당 사용자에 대한 추천을 받을 수 있습니다. 익명 사용자의 이벤트 기록에 대한 자세한 내용은 [익명 사용자를 위한 이벤트 기록](#) 단원을 참조하세요.

```
aws personalize-runtime get-action-recommendations \
--campaign-arn campaign arn \
--user-id User ID \
--num-results 10
```

작업 추천 받기(AWS SDK)

다음 코드는 캠페인으로부터 사용자에 대한 Amazon Personalize 추천을 받는 방법을 보여줍니다. 추천을 받을 사용자의 ID와 캠페인의 Amazon 리소스 이름(ARN)을 지정합니다.

추천 작업 수를 변경하려면 `numResults`의 값을 변경합니다. 기본값은 5개 작업입니다. 최대 값은 100개 작업입니다.

사용자 지정 기준에 따라 작업 추천을 필터링하려면 필터를 생성하여 [GetActionRecommendations](#) API 요청에 적용할 수 있습니다. 자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 섹션을 참조하세요.

로그인하기 전에 사용자(익명 사용자)에 대한 이벤트를 기록한 경우, 해당 이벤트의 `sessionId`를 `userId`인 것처럼 제공하여 해당 사용자에 대한 추천을 받을 수 있습니다. 익명 사용자의 이벤트 기록에 대한 자세한 내용은 [익명 사용자를 위한 이벤트 기록](#) 단원을 참조하세요.

```
import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_action_recommendations(
    campaignArn = 'Campaign ARN',
    userId = 'User ID',
    numResults = 10
)

print("Recommended actions")
for item in response['actionList']:
    print (item['actionId'])
```

개인 맞춤형 순위 받기(사용자 지정 리소스)

개인 맞춤형 순위는 특정 사용자에게 맞춰 순위가 다시 지정된 추천 항목의 목록입니다. 개인 맞춤형 순위를 받으려면 [GetPersonalizedRanking](#) API 작업을 호출하거나 콘솔의 캠페인으로부터 추천을 받습니다.

Note

캠페인을 지원하는 솔루션은 PERSONALIZED_RANKING 유형 레시피를 사용하여 생성한 것 이어야 합니다. 자세한 내용은 [레시피 선택](#) 단원을 참조하세요.

주제

- [개인 맞춤형 순위 점수 매기기의 작동 방식](#)
- [개인 맞춤형 순위 받기\(콘솔\)](#)
- [개인 맞춤형 순위 받기\(AWS CLI\)](#)
- [맞춤형 순위 \(AWS SDK\) 가져오기](#)
- [개인 맞춤형-순위 샘플 노트북](#)

개인 맞춤형 순위 점수 매기기의 작동 방식

User-Personalization-v2 및 User-Personalization 레시피로 만든 솔루션에 대해 GetRecommendations 작업 시 반환되는 점수와 마찬가지로 GetPersonalizedRanking 점수 합계는 1이지만 입력 항목만 점수를 받고 추천 점수는 더 높은 경향이 있습니다.

수학적으로 보면 u 의 점수 함수는 입력 항목만 GetPersonalizedRanking 고려한다는 GetRecommendations 점을 제외하면 과 동일합니다. 다시 말해서, 점수를 나눌 다른 선택 항목 수가 적기 때문에 점수가 1에 가까울수록 나올 가능성이 더 높습니다.

$$\text{score}(u, i) = \frac{\exp(\bar{w}_u^\top w_i)}{\sum_{j \in \text{input}} \exp(\bar{w}_u^\top w_j)}$$

개인 맞춤형 순위 받기(콘솔)

Personalize 콘솔에서 사용자에게 대한 개인 맞춤형 순위를 받으려면 사용 중인 캠페인을 선택한 다음 해당 사용자 ID를 제공하고, 사용자에게 대해 순위를 매길 항목의 목록을 지정하고, 선택적으로 필터를 선택하고, 원하는 경우 컨텍스트 데이터를 제공합니다.

사용자에게 대한 개인 맞춤형 순위를 받으려면

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 결과를 필터링할 캠페인이 포함된 데이터세트 그룹을 선택합니다.
3. 탐색 창에서 캠페인을 선택합니다.
4. 캠페인 페이지에서 대상 캠페인을 선택합니다.
5. 캠페인 결과 테스트에서 추천을 받으려는 사용자의 사용자 ID를 입력합니다.
6. 항목 ID에서 해당 사용자에게 대해 순위를 매길 항목의 목록을 입력합니다.
7. 필터를 선택합니다. 자세한 정보는 [추천 및 사용자 세그먼트 필터링](#)을 참조하세요.
8. 캠페인에 대한 추천에서 메타데이터를 활성화한 경우, 항목 데이터 세트 열에서 추천 결과에 포함할 메타데이터 열을 선택합니다. 메타데이터 활성화에 대한 자세한 내용은 [권장 사항의 항목 메타데이터](#) 섹션을 참조하세요.
9. 캠페인에서 컨텍스트 메타데이터(요구 사항은 [컨텍스트 메타데이터로 추천 관련성 높이기](#)단원 참조)를 사용하는 경우 선택적으로 컨텍스트 데이터를 제공합니다.

각 컨텍스트에 대해 키에 메타데이터 필드를 입력하고 값에 컨텍스트 데이터를 입력합니다.

10. 개인 맞춤형 항목 순위 받기를 선택합니다. 사용자의 예상 관심도에 따라 순위가 매겨진 항목이 포함된 표가 나타납니다.

개인 맞춤형 순위 받기(AWS CLI)

다음 코드 샘플은 AWS CLI를 사용하여 개인 맞춤형 순위를 받을 방법을 다양하게 보여줍니다.

주제

- [개인 맞춤형 순위 받기](#)
- [개인 맞춤형 순위에 항목 메타데이터 포함](#)

개인 맞춤형 순위 받기

AWS CLI를 사용하여 개인 맞춤형 순위를 받으려면 다음과 같은 `get-personalized-ranking` 명령을 사용합니다. 캠페인의 리소스 이름(ARN), 해당 사용자의 사용자 ID를 지정하고 해당 사용자의 순위가 매겨진 항목의 항목 ID 목록(각각 공백으로 구분)을 제공합니다. 순위를 매길 항목은 솔루션 버전을 학습시키는 데 사용한 데이터에 포함되어야 합니다. 순위가 지정된 추천의 목록이 표시됩니다. Personalize는 목록에서 사용자가 가장 관심을 갖는 첫 번째 항목을 고려합니다.

```
aws personalize-runtime get-personalized-ranking \
--campaign-arn Campaign ARN \
--user-id 12 \
--input-list 3 4 10 8 12 7
```

개인 맞춤형 순위에 항목 메타데이터 포함

캠페인에 대한 추천에서 메타데이터를 활성화한 경우, 응답에 포함할 항목 데이터 세트 메타데이터 열을 지정할 수 있습니다. 메타데이터 활성화에 대한 자세한 내용은 [권장 사항의 항목 메타데이터](#) 섹션을 참조하세요.

다음 코드 샘플은 개인 맞춤형 순위 요청의 일부로 메타데이터 열을 지정하는 방법을 보여줍니다.

```
aws personalize-runtime get-personalized-ranking \
--campaign-arn Campaign ARN \
--user-id 12 \
--input-list 3 4 10 8 12 7
--metadata-columns "{\"ITEMS\": [\"columnA\", \"columnB\"]}"
```

맞춤형 순위 (AWS SDK) 가져오기

다음 코드 샘플은 SDK를 사용하여 개인화된 순위를 얻는 방법이 얼마나 다양한지 보여줍니다. AWS

주제

- [개인 맞춤형 순위 받기](#)
- [개인 맞춤형 순위에 항목 메타데이터 포함](#)
- [상황별 메타데이터를 사용하여 개인 맞춤형 순위 받기](#)

개인 맞춤형 순위 받기

다음 코드는 사용자의 개인 맞춤형 순위를 받는 방법을 보여줍니다. 사용자 ID와 사용자에 대해 순위를 매길 항목 ID의 목록을 지정합니다. 항목 ID는 솔루션 버전을 학습시키는 데 사용한 데이터에 포함되

어야 합니다. 순위가 지정된 추천의 목록이 반환됩니다. Personalize는 목록에서 사용자가 가장 관심을 갖는 첫 번째 항목을 고려합니다.

SDK for Python (Boto3)

```
import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_personalized_ranking(
    campaignArn = "Campaign arn",
    userId = "UserID",
    inputList = ['ItemID1', 'ItemID2']
)

print("Personalized Ranking")
for item in response['personalizedRanking']:
    print (item['itemId'])
```

SDK for Java 2.x

```
public static List<PredictedItem> getRankedRecs(PersonalizeRuntimeClient
personalizeRuntimeClient,
                                           String campaignArn,
                                           String userId,
                                           ArrayList<String> items) {

    try {
        GetPersonalizedRankingRequest rankingRecommendationsRequest =
        GetPersonalizedRankingRequest.builder()
            .campaignArn(campaignArn)
            .userId(userId)
            .inputList(items)
            .build();

        GetPersonalizedRankingResponse recommendationsResponse =

personalizeRuntimeClient.getPersonalizedRanking(rankingRecommendationsRequest);
        List<PredictedItem> rankedItems =
recommendationsResponse.personalizedRanking();
        int rank = 1;
        for (PredictedItem item : rankedItems) {
            System.out.println("Item ranked at position " + rank + " details");
```

```
        System.out.println("Item Id is : " + item.itemId());
        System.out.println("Item score is : " + item.score());
        System.out.println("-----");
        rank++;
    }
    return rankedItems;
} catch (PersonalizeRuntimeException e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
return null;
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { GetPersonalizedRankingCommand } from
    "@aws-sdk/client-personalize-runtime";
import { personalizeRuntimeClient } from "./libs/personalizeClients.js";
// Or, create the client here.
// const personalizeRuntimeClient = new PersonalizeRuntimeClient({ region:
    "REGION"});

// Set the ranking request parameters.
export const getPersonalizedRankingParam = {
    campaignArn: "CAMPAIGN_ARN", /* required */
    userId: 'USER_ID', /* required */
    inputList: ["ITEM_ID_1", "ITEM_ID_2", "ITEM_ID_3", "ITEM_ID_4"]
}

export const run = async () => {
    try {
        const response = await personalizeRuntimeClient.send(new
        GetPersonalizedRankingCommand(getPersonalizedRankingParam));
        console.log("Success!", response);
        return response; // For unit tests.
    } catch (err) {
        console.log("Error", err);
    }
};
run();
```

개인 맞춤형 순위에 항목 메타데이터 포함

캠페인에 대한 추천에서 메타데이터를 활성화한 경우, 응답에 포함할 항목 데이터 세트 메타데이터 열을 지정할 수 있습니다. 메타데이터 활성화에 대한 자세한 내용은 [권장 사항의 항목 메타데이터](#) 섹션을 참조하세요.

다음 코드 샘플은 개인 맞춤형 순위 요청의 일부로 메타데이터 열을 지정하는 방법을 보여줍니다.

```
import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_personalized_ranking(
    campaignArn = "Campaign arn",
    userId = "UserID",
    inputList = [ItemID1, ItemID2],
    metadataColumns = {
        "ITEMS": [columnNameA, columnNameB]
    }
)

print("Personalized Ranking")
for item in response['personalizedRanking']:
    print (item['itemId'])
    print (item['metadata'])
```

상황별 메타데이터를 사용하여 개인 맞춤형 순위 받기

컨텍스트 메타데이터를 기반으로 개인 맞춤형 순위를 받으려면 다음 코드를 사용합니다. context의 경우, 각 키-값 쌍에 대해 메타데이터 필드를 키로 제공하고 컨텍스트 데이터를 값으로 제공합니다. 다음 샘플 코드에서는 키가 DEVICE이고 값은 mobile phone입니다. 이들 값과 Campaign ARN 및 User ID를 사용자의 값으로 바꿉니다. inputList를 솔루션을 학습시키는 데 사용한 데이터에 들어 있는 항목 ID의 목록으로 변경합니다. Personalize는 목록에서 사용자가 가장 관심을 갖는 첫 번째 항목을 고려합니다.

```
import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_personalized_ranking(
    campaignArn = "Campaign ARN",
    userId = "User ID",
```

```

inputList = ['ItemID1', 'ItemID2'],
context = {
    'DEVICE': 'mobile phone'
}
)

print("Personalized Ranking")
for item in response['personalizedRanking']:
    print(item['itemId'])

```

개인 맞춤형-순위 샘플 노트북

개인 맞춤형-순위 레시피를 사용하는 방법을 보여주는 Jupyter 노트북 샘플은 [Personalize 순위 예제](#) 단원을 참조하세요.

컨텍스트 메타데이터로 추천 관련성 높이기

추천 관련성을 높이려면 사용자의 기기 유형이나 하루 중 항목 추천을 받거나 개인 맞춤형 순위를 받는 시간과 같이 사용자에게 대한 상황별 메타데이터를 포함시킵니다.

상황별 메타데이터를 사용하려면 항목 상호 작용 데이터 세트의 스키마에 상황별 데이터에 대한 메타데이터 필드가 있어야 합니다. DEVICE 필드를 예로 들 수 있습니다([스키마](#) 참조).

도메인 데이터세트 그룹의 경우 다음과 같은 추천자 사용 사례에서 컨텍스트 메타데이터를 사용할 수 있습니다.

- [추천 제품](#)(ECOMMERCE 도메인)
- [가장 적합한 추천 제품](#)(VIDEO_ON_DEMAND 도메인)

사용자 지정 리소스의 경우 컨텍스트 메타데이터를 사용하는 레시피에는 다음이 포함됩니다.

- [사용자 개인화 - v2](#) 및 [사용자-개인 맞춤](#)
- [맞춤형 랭킹-v2](#) 및 [개인 맞춤형-순위](#)

컨텍스트 정보에 대한 자세한 내용은 다음 AWS Machine Learning 블로그 게시물을 참조하십시오. 컨텍스트 정보를 활용하여 [Amazon Personalize 권장 사항의 관련성 높이기](#).

Amazon Personalize 콘솔 AWS Command Line Interface ,AWS CLI() 또는 SDK를 사용하여 상황별 메타데이터가 포함된 권장 사항을 얻을 수 있습니다. AWS

상황별 메타데이터를 사용하여 추천 받기 (AWS Python SDK)

추천 관련성을 높이려면 사용자의 기기 유형이나 하루 중 항목 추천을 받거나 개인 맞춤형 순위를 받는 시간과 같이 사용자에게 대한 상황별 메타데이터를 포함시킵니다.

상황별 컨텍스트 메타데이터를 기반으로 추천을 받으려면 다음 코드를 사용합니다. context의 경우, 각 키-값 쌍에 대해 메타데이터 필드를 키로 제공하고 컨텍스트 데이터를 값으로 제공합니다. 다음 샘플 코드에서는 키가 DEVICE이고 값은 mobile phone입니다. 이들 값과 Campaign ARN 및 User ID를 사용자의 값으로 바꿉니다. 추천자를 만든 경우 campaignArn을 recommenderArn로 바꿉니다. 해당 사용자에게 대한 추천 항목의 목록이 표시됩니다.

```
import boto3

personalizeRt = boto3.client('personalize-runtime')

response = personalizeRt.get_recommendations(
    campaignArn = 'Campaign ARN',
    userId = 'User ID',
    context = {
        'DEVICE': 'mobile phone'
    }
)

print("Recommended items")
for item in response['itemList']:
    print (item['itemId'])
```

배치 권장 사항 및 사용자 세그먼트(사용자 지정 리소스)

사용자 지정 리소스를 사용하면 비동기 배치 워크플로로 배치 추천 또는 사용자 세그먼트를 확보할 수 있습니다. 예를 들어, 이메일 목록의 모든 사용자에게 대한 제품 권장 사항을 가져오거나 전체 인벤토리에서 [항목 간 유사점](#)을 가져올 수 있습니다. 또는 USER_SEGMENTATION 레시피를 사용하면 인벤토리의 항목과 사용자 상호작용을 기반으로 데이터 기반 광고를 위한 사용자 세그먼트를 받을 수 있습니다.

- 배치 권장 사항을 가져오려면 배치 추론 작업을 사용합니다. 배치 추론 작업은 Amazon S3 버킷에서 배치 입력 데이터를 가져오고, 솔루션 버전을 사용하여 항목 권장 사항을 생성하고, 권장 사항을 Amazon S3 버킷에 JSON 형식으로 내보내는 톨입니다.

- 사용자 세그먼트를 가져오려면 배치 세그먼트 작업을 사용합니다. 배치 세그먼트 작업은 Amazon S3 버킷에서 배치 입력 데이터를 가져오고, SER_SEGMENTATION로 교육받은 솔루션 버전을 사용하여 사용자 세그먼트를 생성하고, 권장 사항을 Amazon S3 버킷에 내보내는 툴입니다.

주제

- [배치 추천 받기](#)
- [사용자 세그먼트 가져오기](#)

배치 추천 받기

사용자 지정 리소스를 사용하면 비동기 배치 워크플로우로 항목 추천을 받을 수 있습니다. 예를 들어 이메일 목록에 있는 모든 사용자를 위한 제품 추천이나 인벤토리 전반의 [item-to-item 유사점](#)을 받을 수 있습니다.

항목에 대한 배치 추천을 받으려면 배치 추론 작업을 사용합니다. 배치 추론 작업은 S3 버킷에서 배치 입력 데이터를 가져오고, 사용자 지정 솔루션 버전을 사용하여 항목 추천을 생성한 다음, 이 항목 추천을 S3 버킷으로 내보내는 도구입니다. 레시피에 따라 입력 데이터는 사용자 또는 항목 목록이거나 각각 항목 모음이 있는 사용자 목록이 될 수 있습니다.

솔루션에서 유사 항목 레시피를 사용하고 텍스트 데이터 및 항목 제목 데이터가 포함된 항목 데이터 세트가 있는 경우, 각 항목 그룹에 대한 테마가 있는 배치 추천을 생성할 수 있습니다. 자세한 정보는 [콘텐츠 생성기에서 테마가 있는 배치 추천](#)을 참조하세요.

배치 추천을 생성할 때 Personalize는 마지막 솔루션 버전 생성 시점에 존재하는 모든 대량 데이터를 고려합니다. 이 데이터는 전체 또는 증분 가져오기 모드를 사용하여 가져올 수 있습니다. 더 나중의 대량 레코드가 배치 추천에 영향을 미치게 하려면 새 솔루션 버전을 만든 다음 배치 추론 작업을 생성해야 합니다.

Personalize는 배치 추천을 생성할 때 다음과 같이 개별 가져오기의 데이터를 사용합니다.

- 기존 항목 및 사용자와의 새로운 상호 작용: USER_PERSONALIZED 또는 PERSONALIZED_RANKING 레시피를 사용하는 경우 Amazon Personalize는 데이터 가져오기 후 약 15분 이내에 기존 항목 및 사용자와의 새 상호 작용 데이터를 고려합니다. 이벤트를 고려하게 하려면 가져오기 후 배치 추론 작업을 시작하기 전에 최소 15분 정도 기다렸다가 진행하는 것이 좋습니다. 다른 모든 레시피의 경우 배치 추천에 영향을 미치게 하려면 스트리밍된 이벤트에 대한 새로운 솔루션 버전을 만들어야 합니다.
- 신규 사용자: 상호작용 데이터가 없는 사용자의 경우 처음에는 인기 항목만 추천됩니다. USER_PERSONALIZATION 또는 PERSONALIZED_RANKING 레시피를 사용하고 사용자를 위한

이벤트를 기록하는 경우 가져오기 후 약 15분 이내에 재교육 없이 해당 권장 사항이 관련성을 높일 수 있습니다. 이벤트를 고려하게 하려면 가져오기 후 배치 추론 작업을 시작하기 전에 최소 15분 정도 기다렸다가 진행하는 것이 좋습니다. 다른 모든 레시피의 경우, 스트리밍된 이벤트에 대한 새 솔루션 버전을 생성하여 상호작용 데이터가 없는 사용자를 위한 배치 추천에 영향을 주어야 합니다.

- 새 항목: User-Personalization-v2 및 User-Personalization을 사용하여 일괄 추론 작업을 생성하고 솔루션에 대해 완전히 훈련된 최신 솔루션 버전을 지정하면 Amazon Personalize는 탐색과 함께 권장 사항에 새 항목을 포함하도록 솔루션 버전을 자동으로 업데이트합니다. 최신 솔루션 버전을 지정하지 않으면 업데이트는 발생하지 않습니다. 다른 레시피의 경우 새 항목을 배치 추천에 포함시키려면 새 솔루션 버전을 생성해야 합니다. 탐색에 대한 자세한 내용은 [탐색](#) 단원을 참조하세요.

주제

- [배치 워크플로](#)
- [지침 및 요구 사항](#)
- [배치 워크플로우 점수 매기기](#)
- [콘텐츠 생성기에서 테마가 있는 배치 추천](#)
- [배치 추천을 위한 입력 데이터 준비](#)
- [배치 추론 작업 생성](#)
- [배치 추론 작업 출력 예제](#)

배치 워크플로

배치 워크플로우는 다음과 같습니다.

1. 입력 데이터를 JSON 형식으로 준비하고 S3 버킷에 업로드합니다. 입력 데이터의 형식은 사용하는 레시피에 따라 달라집니다. [배치 추천을 위한 입력 데이터 준비](#) 단원을 참조하세요.
2. 출력 데이터를 저장할 별도의 위치(폴더 또는 다른 S3 버킷)를 만듭니다.
3. 배치 추론 작업을 생성합니다. [배치 추론 작업 생성](#)를 참조하세요.
4. 배치 추론이 완료되면 S3의 출력 위치에서 항목 추천을 검색합니다.

지침 및 요구 사항

배치 추천을 받기 위한 지침 및 요구 사항은 다음과 같습니다.

- Personalize IAM 서비스 역할에 S3 버킷에 파일을 읽고 추가할 수 있는 권한이 있어야 합니다. 이러한 권한 부여에 대한 자세한 내용은 [배치 워크플로우에 대한 서비스 역할 정책](#) 단원을 참조하세요.

버킷 권한에 대한 자세한 내용은 Simple Storage Service 개발자 안내서의 [사용자 정책 예제](#) 단원을 참조하세요. 암호화에 AWS Key Management Service (AWS KMS) 를 사용하는 경우 Amazon Personalize와 Amazon Personalize IAM 서비스 역할에 키를 사용할 권한을 부여해야 합니다. 자세한 정보는 [Amazon Personalize에 AWS KMS 키 사용 권한 부여](#)를 참조하세요.

- 배치 추론 작업을 생성하기 전에 사용자 지정 솔루션과 솔루션 버전을 생성해야 합니다. 그러나 Personalize 캠페인은 생성할 필요가 없습니다. 도메인 데이터세트 그룹을 생성한 경우에도 사용자 지정 리소스를 생성할 수 있습니다.
- 추천이 포함된 테마를 생성하려면 유사-항목 레시피를 사용해야 합니다. 텍스트 데이터 및 항목 제목 데이터가 포함된 항목 데이터 세트도 있어야 합니다. 테마가 적용된 추천에 대한 자세한 내용은 [콘텐츠 생성기에서 테마가 있는 배치 추천](#) 섹션을 참조하세요.
- 입력 데이터는 [사용자 세그먼트를 위한 입력 데이터 준비](#)에 설명된 대로 형식을 지정해야 합니다.
- 지금-유행 또는 차선책-작업 레시피로는 배치 추천을 받을 수 없습니다.
- 자리 표시자 파라미터가 있는 필터를 사용하는 경우 filterValues객체의 입력 데이터에 파라미터 값을 포함시켜야 합니다. 자세한 내용은 [입력 JSON에서 필터 값 제공](#) 단원을 참조하세요.
- 출력 데이터에는 입력 데이터와 다른 위치(폴더 또는 다른 S3 버킷)를 사용하는 것이 좋습니다.
- Batch 권장 사항은 실시간 권장 사항과 정확히 동일하지 않을 수 있습니다. 이는 일괄 추론 작업을 완료하는 데 시간이 더 오래 걸리고 작업 시작 15분 전에 사용 가능한 데이터만 고려하기 때문입니다.

배치 워크플로우 점수 매기기

배치 추천에는 다음과 같은 점수가 포함됩니다.

- Personalize는 사용자 개인 맞춤 및 개인 맞춤형 순위 레시피를 사용하여 [추천 점수 매기기 작동 방식 \(사용자 지정 리소스\)](#) 및 [개인 맞춤형 순위 점수 매기기의 작동 방식](#)에 설명된 대로 배치 추론 작업 추천 점수를 계산합니다. 점수는 배치 추론 작업의 출력 JSON 파일에서 볼 수 있습니다.
- 유사-항목 레시피를 사용하면, 테마가 적용된 배치 추천을 받는 경우 Amazon Personalize는 테마가 각 항목과 얼마나 관련이 있는지를 기준으로 각 관련 항목 세트의 순위를 매깁니다. 각 항목에는 0에서 1까지의 점수가 포함됩니다. 점수가 높을수록 항목이 테마와 더 밀접하게 관련되어 있습니다. 테마가 있는 추천에 대한 자세한 내용은 [콘텐츠 생성기에서 테마가 있는 배치 추천](#) 섹션을 참조하세요.

콘텐츠 생성기에서 테마가 있는 배치 추천

Important

테마가 있는 배치 추천을 받으면 추가 비용이 발생합니다. 자세한 내용은 [Amazon Personalize 요금](#)을 참조하세요.

[유사-항목 레시피](#)를 사용하는 경우 Amazon Personalize 콘텐츠 생성기가 설명 테마를 배치 추천에 추가할 수 있습니다. 콘텐츠 생성기는 Amazon Personalize에서 관리하는 생성형 인공지능(생성형 AI) 기능입니다.

테마가 있는 배치 추천을 받으면 Amazon Personalize 콘텐츠 생성기가 유사한 각 항목 세트에 대한 설명 테마를 추가합니다. 테마는 항목 데이터 세트의 항목 설명 및 항목 이름 데이터를 기반으로 합니다. Amazon Personalize는 배치 추론 작업의 출력에 테마를 포함합니다. 테마를 사용하여 애플리케이션 또는 마케팅 메시지의 텍스트를 더욱 설득력 있게 만들 수 있습니다.

예를 들어, 아침 식사 음식과 관련된 항목을 추천받는 경우 Amazon Personalize는 개운하게 일어나기 또는 아침 필수품과 같은 테마를 생성할 수 있습니다. 이 테마를 사용하여 함께 자주 구매하는 제품과 같은 일반적인 캐러셀 제목을 대체할 수 있습니다. 또는 새 메뉴 옵션에 대한 홍보 이메일이나 마케팅 캠페인에 테마를 통합할 수도 있습니다.

AWS에서는 콘텐츠 생성기의 테마를 모니터링하지 않습니다. 추천 항목별로 산출된 점수를 사용하여 테마 품질을 확인할 수 있습니다. 자세한 내용은 [테마가 있는 배치 추천의 순위 및 점수 산정](#) 섹션을 참조하세요.

주제

- [지원되는 리전](#)
- [지침 및 요구 사항](#)
- [테마가 있는 배치 추천의 순위 및 점수 산정](#)
- [테마가 있는 배치 추천 생성](#)

지원되는 리전

Amazon Personalize 콘텐츠 생성기는 다음 AWS 리전에서만 사용할 수 있습니다.

- 미국 동부(버지니아 북부)

- 미국 서부(오레곤)
- 아시아 태평양(도쿄)

지침 및 요구 사항

테마가 있는 추천을 생성하기 위한 지침 및 요구 사항은 다음과 같습니다.

- 입력 파일은 최대 100개의 항목을 포함할 수 있습니다. 배치 추천의 입력 데이터에 대한 자세한 내용은 [배치 추천을 위한 입력 데이터 준비](#) 섹션을 참조하세요.
- 솔루션은 [유사-항목 레시피](#)를 사용해야 합니다.
- 다음 데이터가 포함된 항목 데이터 세트가 있어야 합니다. 이 데이터는 관련성이 더 높은 테마를 생성하는 데 도움이 될 수 있습니다.
 - DESCRIPTION 필드와 같은 텍스트 필드가 있어야 합니다. 텍스트 데이터에 대한 자세한 내용은 [비정형 텍스트 메타데이터](#) 섹션을 참조하세요.
 - 항목 이름 데이터가 포함된 문자열 열(예: TITLE 필드)이 있어야 합니다.

항목 데이터 세트에 이 데이터가 없는 경우 추가할 수 있습니다. 기존 데이터 업데이트에 대한 자세한 내용은 [데이터세트로 더 많은 훈련 데이터 가져오기](#) 단원을 참조하세요.

테마가 있는 배치 추천의 순위 및 점수 산정

테마가 있는 배치 추천을 받으면 Amazon Personalize는 테마가 각 항목과 얼마나 관련이 있는지를 기준으로 각 항목 세트의 순위를 매깁니다. 각 항목에는 대략 -0.1에서 0.6까지의 점수가 포함됩니다. 점수가 높을수록 항목이 테마와 더 밀접하게 관련되어 있습니다. 점수를 사용하여 테마와 밀접한 관련이 있는 항목만 표시하도록 임계값을 설정할 수 있습니다.

예를 들어, Amazon Personalize는 For your sweet tooth 테마를 반환할 수 있으며 관련 항목 및 점수는 하드 캔디(0.19884521점), 초콜릿(0.17664525점), 사과(0.08994528점), 아이스크림(0.14294521점), 고구마(0.07794527점), 당근(0.04994523점)일 수 있습니다. 애플리케이션에 과일과 채소를 제외하고 .10점 이상의 항목만 포함하도록 규칙을 추가할 수 있습니다.

다음 예제는 테마가 있는 영화 추천을 생성하는 배치 추천 작업의 출력 형식을 보여줍니다.

```
{
  "input": {
    "itemId": "40",
    "output": {
      "recommendedItems": [
        "36", "50", "44", "22", "21", "29", "3", "1", "2", "39"
      ],
      "theme": "Movies with a strong female lead",
      "itemsThemeRelevanceScores": [
        0.19994527, 0.183059963, 0.17478035, 0.1618133, 0.1574806, 0.15468733, 0.1499242, 0.14353688, 0.135314
      ]
    }
  }
}
```

```
{
  "input": {
    "itemId": "43"
  },
  "output": {
    "recommendedItems": [
      "50", "21", "36", "3", "17", "2", "39", "1", "10", "5"
    ],
    "theme": "The best movies of 1995",
    "itemsThemeRelevanceScores": [
      0.184988, 0.1795761, 0.11143453, 0.0989443, 0.08258403, 0.07952615, 0.07115086, 0.0621634, -0.138913, ...
    ]
  }
}
```

테마가 있는 배치 추천 생성

테마가 있는 배치 추천을 생성하려면 [배치 워크플로](#)에 설명된 대로 일괄 워크플로를 완료해야 합니다. RELATED_ITEMS 레시피와 동일한 방식으로 입력 데이터를 준비합니다. 예시는 [RELATED_ITEMS 레시피](#)에서 확인하십시오.

배치 추론 작업을 생성할 때 테마 생성을 활성화하고 항목 데이터 세트의 항목 제목 열을 지정합니다.

- Amazon Personalize 콘솔을 사용하여 테마를 생성하는 배치 추론 작업을 만드는 방법에 대한 자세한 내용은 [배치 추론 작업 생성](#) 섹션을 참조하세요.
- Python용 SDK(Boto3)를 사용하여 테마를 생성하는 배치 추론 작업을 만드는 방법을 보여주는 코드 샘플은 [테마를 생성하는 배치 추론 작업 생성](#) 섹션을 참조하세요.

배치 추천을 위한 입력 데이터 준비

배치 추론 작업은 S3 버킷에서 배치 입력 JSON 데이터를 가져오고, 사용자 지정 솔루션 버전을 사용하여 추천을 생성한 다음, 항목 추천을 S3 버킷으로 내보냅니다. 배치 추천을 받으려면 먼저 JSON 파일을 준비하고 S3 버킷에 업로드해야 합니다. S3 버킷에 출력 폴더를 생성하거나 별도의 출력 S3 버킷을 사용하는 것이 좋습니다. 그런 다음 동일한 입력 데이터 위치를 사용하여 여러 배치 추론 작업을 실행할 수 있습니다.

자리 표시자 파라미터가 포함된 필터(예: \$GENRE)를 사용하는 경우 filterValues객체의 파라미터 값을 입력 JSON에 제공해야 합니다. 자세한 내용은 [입력 JSON에서 필터 값 제공](#) 단원을 참조하세요.

데이터를 준비하고 가져오려면

1. 레시피에 따라 배치 입력 데이터의 형식을 지정합니다. 지금-유행 레시피로는 배치 추천을 받을 수 없습니다.
 - USER_PERSONALIZATION 레시피와 인기도-집계 레시피의 경우 입력 데이터는 사용자 ID 목록이 포함된 JSON 파일입니다
 - RELATED_ITEMS 레시피의 경우 입력 데이터는 항목 ID 목록입니다
 - PERSONALIZED_RANKING 레시피의 경우 입력 데이터는 사용자 ID 목록이며, 각 목록은 ItemID 컬렉션과 쌍을 이룹니다

- 각 행을 새 줄로 구분합니다. 입력 데이터 예제는 [배치 추론 작업 입력 및 출력 JSON 예제](#) 단원을 참조하세요.
2. S3 버킷의 입력 폴더에 입력 JSON을 업로드합니다. 자세한 내용은 Simple Storage Service 사용 설명서의 [드래그 앤 드롭을 사용하여 파일 및 폴더 업로드](#) 단원을 참조하세요.
 3. 출력 데이터를 저장할 별도의 위치(폴더 또는 다른 S3 버킷)를 만듭니다. 출력 JSON을 위한 별도의 위치를 생성하면 동일한 입력 데이터 위치로 여러 배치 추론 작업을 실행할 수 있습니다.
 4. 배치 추론 작업을 생성합니다. Personalize는 솔루션 버전의 추천을 출력 데이터 위치로 출력합니다.

배치 추론 작업 입력 및 출력 JSON 예제

사용하는 레시피에 따라 입력 데이터 형식을 지정하는 방법. 자리 표시자 파라미터가 포함된 필터(예: \$GENRE)를 사용하는 경우 `filterValues` 객체의 파라미터 값을 입력 JSON에 제공해야 합니다. 자세한 내용은 [입력 JSON에서 필터 값 제공](#) 단원을 참조하세요.

다음에 이어지는 단원에는 배치 추론 작업에 대한 올바른 형식의 JSON 입력 및 출력 예제가 나열되어 있습니다. 지금-유행 레시피로는 배치 추천을 받을 수 없습니다.

주제

- [USER_PERSONALIZATION 레시피](#)
- [POPULAR_ITEMS 레시피\(인기도-집계만 가능\)](#)
- [PERSONALIZED_RANKING 레시피](#)
- [RELATED_ITEMS 레시피](#)

USER_PERSONALIZATION 레시피

다음은 USER_PERSONALIZATION 레시피에 대한 올바른 형식의 JSON 입력 및 출력 예제를 보여줍니다. User-Personalization-v2를 사용하는 경우 각 권장 항목에는 해당 항목이 권장 사항에 포함된 이유 목록이 포함됩니다. 이 목록은 비어 있을 수 있습니다. 가능한 이유에 대한 자세한 내용은 [참조하십시오 추천 이유 \(사용자 맞춤화-v2\)](#).

Input

다음과 같이 각 `userId`을 새 줄로 구분합니다.

```
{"userId": "4638"}
```

```

{"userId": "663"}
{"userId": "3384"}
...

```

Output

```

{"input":{"userId":"4638"},"output":{"recommendedItems":
["63992","115149","110102","148626","148888","31685","102445","69526","92535","143355","6237
[0.0152238,0.0069081,0.0068222,0.006394,0.0059746,0.0055851,0.0049357,0.0044644,0.0042968,0.
{"input":{"userId":"663"},"output":{"recommendedItems":
["368","377","25","780","1610","648","1270","6","165","1196","1097","300","1183","608","104
[0.0406197,0.0372557,0.0254077,0.0151975,0.014991,0.0127175,0.0124547,0.0116712,0.0091098,0.
{"input":{"userId":"3384"},"output":{"recommendedItems":
["597","21","223","2144","208","2424","594","595","920","104","520","367","2081","39","1035
[0.0241061,0.0119394,0.0118012,0.010662,0.0086972,0.0079428,0.0073218,0.0071438,0.0069602,0.
...

```

POPULAR_ITEMS 레시피(인기도-집계만 가능)

다음은 인기도-집계 레시피에 대한 올바른 형식의 JSON 입력 및 출력 예제를 보여줍니다. 지금-유형 레시피로는 배치 추천을 받을 수 없습니다.

Input

다음과 같이 각 `userId`을 새 줄로 구분합니다.

```

{"userId": "12"}
{"userId": "105"}
{"userId": "41"}
...

```

Output

```

{"input": {"userId": "12"}, "output": {"recommendedItems": ["105", "106", "441"]}}
{"input": {"userId": "105"}, "output": {"recommendedItems": ["105", "106", "441"]}}
{"input": {"userId": "41"}, "output": {"recommendedItems": ["105", "106", "441"]}}
...

```

PERSONALIZED_RANKING 레시피

다음은 PERSONALIZED_RANKING 레시피에 대한 올바른 형식의 JSON 입력 및 출력 예제를 보여줍니다.

Input

다음과 같이 순위가 매겨질 각 `userId`와 `itemIds`목록을 새 줄로 구분합니다.

```

{"userId": "891", "itemList": ["27", "886", "101"]}
{"userId": "445", "itemList": ["527", "55", "901"]}
{"userId": "71", "itemList": ["27", "351", "101"]}
...

```

Output

```

{"input":{"userId":"891","itemList":["27","886","101"]},"output":
{"recommendedItems":["27","101","886"],"scores":[0.48421,0.28133,0.23446]}}
{"input":{"userId":"445","itemList":["527","55","901"]},"output":
{"recommendedItems":["901","527","55"],"scores":[0.46972,0.31011,0.22017]}}
{"input":{"userId":"71","itemList":["29","351","199"]},"output":{"recommendedItems":
["351","29","199"],"scores":[0.68937,0.24829,0.06232]}}
...

```

RELATED_ITEMS 레시피

다음은 RELATED_ITEMS 레시피에 대한 올바른 형식의 JSON 입력 및 출력 예제를 보여줍니다.

Input

다음과 같이 각 `itemId`을 새 줄로 구분합니다.

```

{"itemId": "105"}
{"itemId": "106"}
{"itemId": "441"}
...

```

Output

```

{"input": {"itemId": "105"}, "output": {"recommendedItems": ["106", "107", "49"]}}
{"input": {"itemId": "106"}, "output": {"recommendedItems": ["105", "107", "49"]}}
{"input": {"itemId": "441"}, "output": {"recommendedItems": ["2", "442", "435"]}}

```

...

다음은 테마가 있는 유사-항목 레시피에 대한 올바른 형식의 JSON 입력 및 출력 예제를 보여줍니다.

Input

다음과 같이 각 itemId을 새 줄로 구분합니다.

```
{"itemId": "40"}
{"itemId": "43"}
...
```

Output

```
{"input":{"itemId":"40"},"output":{"recommendedItems":
["36","50","44","22","21","29","3","1","2","39"],"theme":"Movies
with a strong female lead","itemsThemeRelevanceScores":
[0.19994527,0.183059963,0.17478035,0.1618133,0.1574806,0.15468733,0.1499242,0.14353688,0.135
{"input":{"itemId":"43"},"output":{"recommendedItems":
["50","21","36","3","17","2","39","1","10","5"],"theme":"The best movies of
1995","itemsThemeRelevanceScores":
[0.184988,0.1795761,0.11143453,0.0989443,0.08258403,0.07952615,0.07115086,0.0621634,-0.13891
...
```

배치 추론 작업 생성

배치 추론 작업을 생성하여 Amazon S3에서 배치 입력 데이터를 기반으로 항목 권장 사항을 받습니다. 입력 데이터는 JSON 형식의 사용자 또는 항목 (또는 둘 다) 목록일 수 있습니다. Amazon Personalize 콘솔, AWS Command Line Interface AWS CLI () 또는 SDK를 사용하여 일괄 추론 작업을 생성할 수 있습니다. AWS

배치 추론 작업을 생성할 때 입력 및 출력 위치에 Amazon S3 경로를 지정합니다. Amazon S3는 접두사를 기반으로 합니다. 입력 데이터 위치에 접두사를 제공하는 경우, Amazon Personalize는 해당 접두사와 일치하는 모든 파일을 입력 데이터로 사용합니다. 예를 들어, 사용자가 s3://<name of your S3 bucket>/folderName을 제공하고 버킷에 경로가 s3://<name of your S3 bucket>/folderName_test인 폴더가 있는 경우, Amazon Personalize는 두 폴더의 모든 파일을 입력 데이터로 사용합니다. 특정 폴더 내의 파일만을 입력 데이터로 사용하려면 Amazon S3 경로 끝에 접두사 구분 기호를 붙입니다(예: /: s3://<name of your S3 bucket>/folderName/). Amazon S3에서 객체를 구성하는 방법에 대한 자세한 내용은 [객체 구성, 나열 및 객체 작업](#)을 참조하세요.

권한 요구 사항, 권장 점수, 입력 데이터 준비 및 가져오기 등 Amazon Personalize의 배치 워크플로에 대한 자세한 내용은 [배치 추천 받기](#)을 참조하세요.

주제

- [배치 추론 작업 생성\(콘솔\)](#)
- [배치 추론 작업 생성\(AWS CLI\)](#)
- [일괄 추론 작업 \(AWS SDK\) 생성](#)

배치 추론 작업 생성(콘솔)

[배치 추천을 위한 입력 데이터 준비](#) 작업을 완료했으면 일괄 추론 작업을 만들 준비가 된 것입니다. 이 절차에서는 솔루션과 솔루션 버전(교육된 모델)을 이미 생성한 것으로 가정합니다.

배치 추론 작업을 생성하려면(콘솔)

1. <https://console.aws.amazon.com/acm-pca/home>에서 Amazon Personalize 콘솔을 열고 계정에 로그인하세요.
2. 데이터 세트 그룹 페이지에서 데이터 세트 그룹을 선택합니다.
3. 탐색 창의 사용자 지정 리소스에서 배치 추론 작업을 선택합니다.
4. 배치 추론 작업 생성을 선택합니다.
5. 배치 추론 작업 유형을 선택합니다.
 - 테마 없이 항목 추천을 생성하려면 항목 추천을 선택합니다.
 - 유사-항목 레시피를 사용하고 유사한 항목 그룹에 설명 테마를 추가하려면 콘텐츠 생성기를 통한 테마 적용 추천을 선택합니다. 테마를 생성하려면 항목 이름 데이터 및 텍스트 데이터가 포함된 항목 데이터 세트가 있어야 합니다. 자세한 설명은 [콘텐츠 생성기에서 테마가 있는 배치 추천](#) 섹션을 참조하세요.
6. 배치 추론 작업 세부 정보의 배치 추론 작업 이름에서 배치 추론 작업의 이름을 지정합니다.
7. 솔루션의 경우, 솔루션을 선택한 다음 권장 사항을 생성하는 데 사용하려는 솔루션 버전 ID를 선택합니다.
8. 결과의 수의 경우, 입력 데이터의 각 줄에 대한 권장 사항의 수를 원하는 대로 지정할 수 있습니다. 기본값은 25입니다.
9. 배치 작업으로 테마가 포함된 추천이 생성되는 경우 테마 적용 추천 세부 정보에서 항목 데이터 세트의 항목 이름이나 제목이 포함된 열을 선택합니다. 이 데이터는 관련성이 더 높은 테마를 생성하는 데 도움이 될 수 있습니다. 자세한 설명은 [콘텐츠 생성기에서 테마가 있는 배치 추천](#) 섹션을 참조하세요.

10. 입력 소스에서 입력 파일의 Amazon S3 경로를 지정합니다.

다음 구문 사용: **s3://<name of your S3 bucket>/<folder name>/<input JSON file name>.json**

입력 데이터는 솔루션에서 사용하는 레시피에 맞는 올바른 형식이어야 합니다. 입력 데이터 예제는 [배치 추론 작업 입력 및 출력 JSON 예제](#) 단원을 참조하세요.

11. 복호화 키의 경우 버킷 암호화에 자체 AWS KMS 키를 사용하는 경우 키의 Amazon 리소스 이름 (ARN) 을 지정하십시오. 사용자의 키를 사용하려면 Amazon Personalize에 권한이 있어야 합니다. 권한을 부여하는 방법에 대한 자세한 내용은 [Amazon Personalize에 AWS KMS 키 사용 권한 부여](#) 섹션을 참조하세요.
12. 출력 대상에서 출력 위치의 경로를 지정합니다. 출력 데이터에 대한 다른 위치(폴더 또는 다른 Amazon S3 버킷)를 사용하는 것이 좋습니다.

다음 구문 사용: **s3://<name of your S3 bucket>/<output folder name>/**

13. 암호화 키의 경우 암호화에 자체 AWS KMS 키를 사용하는 경우 키의 ARN을 지정하십시오. 사용자의 키를 사용하려면 Amazon Personalize에 권한이 있어야 합니다. 권한을 부여하는 방법에 대한 자세한 내용은 [Amazon Personalize에 AWS KMS 키 사용 권한 부여](#) 섹션을 참조하세요.
14. IAM 서비스 역할의 경우, 설정 중에 Amazon Personalize용으로 생성한 IAM 서비스 역할을 선택합니다. 이 역할에는 각각 입력 및 출력 Amazon S3 버킷에 대한 읽기 및 쓰기 액세스 권한이 있어야 합니다.
15. 필터의 경우 필요에 따라 필터를 선택하여 배치 추천에 필터를 적용합니다. 필터가 자리표시자 파라미터를 사용하는 경우, 파라미터 값이 입력 JSON에 포함되어 있는지 확인하세요. 자세한 내용은 [입력 JSON에서 필터 값 제공](#) 섹션을 참조하세요.
16. 태그의 경우, 원하는 태그를 추가할 수 있습니다. Amazon Personalize 리소스 태그 지정에 대한 자세한 내용은 [Personalize 리소스에 태그 지정](#) 단원을 참조하세요.
17. 배치 추론 작업 생성을 선택합니다. 배치 추론 작업 생성이 시작되고 배치 추론 작업 페이지가 나타나고 배치 추론 작업 세부 정보 단원이 표시됩니다.

배치 추론 작업의 상태가 활성으로 변경되면 지정된 출력 S3 버킷에서 작업 출력을 찾을 수 있습니다. 출력 파일 이름의 형식은 *input-name.out*이 됩니다.

배치 추론 작업 생성(AWS CLI)

[배치 추천을 위한 입력 데이터 준비](#) 작업을 완료했으면 [CreateBatchInferenceJob](#) 작업을 사용하여 배치 추론 작업을 생성할 준비가 된 것입니다.

주제

- [배치 추론 작업 생성](#)
- [테마를 생성하는 배치 추론 작업 생성](#)

배치 추론 작업 생성

create-batch-inference-job 명령을 사용하여 배치 추론 작업을 생성할 수 있습니다. 작업 이름을 지정하고, Solution version ARN을 솔루션 버전의 Amazon 리소스 이름 (ARN)으로 대체하고, IAM service role ARN을 설정 중에 Amazon Personalize용으로 생성한 IAM 서비스 역할의 ARN으로 대체하세요. 이 역할에는 각각 입력 및 출력 Amazon S3 버킷에 대한 읽기 및 쓰기 액세스 권한이 있어야 합니다. 원하는 필터 ARN을 제공하여 권장 사항을 필터링합니다. 필터가 자리표시자 파라미터를 사용하는 경우, 파라미터 값이 입력 JSON에 포함되어 있는지 확인하세요. 자세한 내용은 [배치 추천 및 사용자 세그먼트\(사용자 지정 리소스\) 필터링](#) 섹션을 참조하세요.

S3 input path 및 S3 output path를 입력 파일 및 출력 위치에 대한 Amazon S3 경로로 대체합니다. 출력 데이터에 대한 다른 위치(폴더 또는 다른 Amazon S3 버킷)를 사용하는 것이 좋습니다. 입력 및 출력 위치에는 구문: **s3://<name of your S3 bucket>/<folder name>/<input JSON file name>.json** 및 **s3://<name of your S3 bucket>/<output folder name>/**을 사용합니다.

이 예제에는 선택 가능한 사용자-개인화 레시피별 itemExplorationConfig 하이퍼파라미터: explorationWeight 및 explorationItemAgeCutOff 가 포함되어 있습니다. 원하는 대로 explorationWeight 및 explorationItemAgeCutOff 값을 포함하여 탐색을 구성합니다. 자세한 설명은 [사용자-개인 맞춤 레시피](#) 섹션을 참조하세요.

```
aws personalize create-batch-inference-job \
  --job-name Batch job name \
  --solution-version-arn Solution version ARN \
  --filter-arn Filter ARN \
  --job-input s3DataSource={path=s3://S3 input path} \
  --job-output s3DataDestination={path=s3://S3 output path} \
  --role-arn IAM service role ARN \
  --batch-inference-job-config "{\"itemExplorationConfig\":{\"explorationWeight\":\
  \"0.3\", \"explorationItemAgeCutOff\": \"30\"}}\""
```

테마를 생성하는 배치 추론 작업 생성

유사한 항목에 대한 테마를 생성하려면 유사-항목 레시피를 사용해야 하며, 항목 데이터 세트에는 텍스트 필드와 항목 이름 데이터 열이 있어야 합니다. 테마가 있는 추천에 대한 자세한 내용은 [콘텐츠 생성기에서 테마가 있는 배치 추천](#) 섹션을 참조하세요.

다음 코드는 테마가 있는 추천을 생성하는 배치 추론 작업을 만듭니다. batch-inference-job-mode 설정을 THEME_GENERATION으로 둡니다. 항목 이름 데이터를 저장하는 열의 이름으로 COLUMN_NAME을 바꿉니다.

```
aws personalize create-batch-inference-job \
--job-name Themed batch job name \
--solution-version-arn Solution version ARN \
--filter-arn Filter ARN \
--job-input s3DataSource={path=s3://S3 input path} \
--job-output s3DataDestination={path=s3://S3 output path} \
--role-arn IAM service role ARN \
--batch-inference-job-mode THEME_GENERATION \
--theme-generation-config "{\"fieldsForThemeGeneration\": {\"itemName\":  
\"COLUMN_NAME\"}}"
```

일괄 추론 작업 (AWS SDK) 생성

[배치 추천을 위한 입력 데이터 준비](#) 작업을 완료했으면 [CreateBatchInferenceJob](#) 작업을 사용하여 배치 추론 작업을 생성할 준비가 된 것입니다.

주제

- [배치 추론 작업 생성](#)
- [테마를 생성하는 배치 추론 작업 생성](#)

배치 추론 작업 생성

다음 코드를 사용하여 배치 추론 작업을 생성할 수 있습니다. 작업 이름, 솔루션 버전의 Amazon 리소스 이름(ARN) 설정 중에 Amazon Personalize용으로 생성한 IAM 서비스 역할의 ARN을 지정합니다. 이 역할에는 입력 및 출력 Amazon S3 버킷에 대한 읽기 및 쓰기 액세스 권한이 있어야 합니다.

출력 데이터에 대한 다른 위치(폴더 또는 다른 Amazon S3 버킷)를 사용하는 것이 좋습니다. 입력 및 출력 위치에는 구문: **s3://<name of your S3 bucket>/<folder name>/<input JSON file name>.json** 및 **s3://<name of your S3 bucket>/<output folder name>/**을 사용합니다.

numResults의 경우, Amazon Personalize가 각 입력 데이터 라인에 대해 예측하기를 원하는 품목의 수를 지정합니다. 원하는 대로 필터 ARN을 제공하여 권장 사항을 필터링합니다. 필터가 자리표시자 파라미터를 사용하는 경우, 파라미터 값이 입력 JSON에 포함되어 있는지 확인하세요. 자세한 내용은 [배치 추천 및 사용자 세그먼트\(사용자 지정 리소스\) 필터링](#) 섹션을 참조하세요.

SDK for Python (Boto3)

이 예제에는 선택 가능한 사용자-개인화 레시피별 itemExplorationConfig 하이퍼파라미터: explorationWeight 및 explorationItemAgeCutOff 가 포함되어 있습니다. 원하는 대로 explorationWeight 및 explorationItemAgeCutOff 값을 포함하여 탐색을 구성합니다. 자세한 설명은 [사용자-개인 맞춤 레시피](#) 섹션을 참조하세요.

```
import boto3

personalize_rec = boto3.client(service_name='personalize')

personalize_rec.create_batch_inference_job (
    solutionVersionArn = "Solution version ARN",
    jobName = "Batch job name",
    roleArn = "IAM service role ARN",
    filterArn = "Filter ARN",
    batchInferenceJobConfig = {
        # optional USER_PERSONALIZATION recipe hyperparameters
        "itemExplorationConfig": {
            "explorationWeight": "0.3",
            "explorationItemAgeCutOff": "30"
        }
    },
    jobInput =
        {"s3DataSource": {"path": "s3://<name of your S3 bucket>/<folder name>/<input JSON file name>.json"}},
    jobOutput =
        {"s3DataDestination": {"path": "s3://<name of your S3 bucket>/<output folder name>/"}}
)
```

SDK for Java 2.x

이 예제에는 선택적 사용자-개인화 레시피별 itemExplorationConfig 필드: explorationWeight 및 explorationItemAgeCutOff이(가) 포함되어 있습니다. 원하는 대로 explorationWeight 및 explorationItemAgeCutOff 값을 포함하여 탐색을 구성합니다. 자세한 설명은 [사용자-개인 맞춤 레시피](#) 섹션을 참조하세요.

```

public static String createPersonalizeBatchInferenceJob(PersonalizeClient
personalizeClient,
                                                    String solutionVersionArn,
                                                    String jobName,
                                                    String filterArn,
                                                    String
s3InputDataSourcePath,
                                                    String
s3DataDestinationPath,
                                                    String roleArn,
                                                    String explorationWeight,
                                                    String
explorationItemAgeCutOff) {

    long waitInMilliseconds = 60 * 1000;
    String status;
    String batchInferenceJobArn;

    try {
        // Set up data input and output parameters.
        S3DataConfig inputSource = S3DataConfig.builder()
            .path(s3InputDataSourcePath)
            .build();
        S3DataConfig outputDestination = S3DataConfig.builder()
            .path(s3DataDestinationPath)
            .build();

        BatchInferenceJobInput jobInput = BatchInferenceJobInput.builder()
            .s3DataSource(inputSource)
            .build();
        BatchInferenceJobOutput jobOutputLocation = BatchInferenceJobOutput.builder()
            .s3DataDestination(outputDestination)
            .build();

        // Optional code to build the User-Personalization specific item exploration
        config.
        HashMap<String, String> explorationConfig = new HashMap<>();

        explorationConfig.put("explorationWeight", explorationWeight);
        explorationConfig.put("explorationItemAgeCutOff", explorationItemAgeCutOff);

        BatchInferenceJobConfig jobConfig = BatchInferenceJobConfig.builder()
            .itemExplorationConfig(explorationConfig)

```

```
        .build());
    // End optional User-Personalization recipe specific code.

    CreateBatchInferenceJobRequest createBatchInferenceJobRequest =
    CreateBatchInferenceJobRequest.builder()
        .solutionVersionArn(solutionVersionArn)
        .jobInput(jobInput)
        .jobOutput(jobOutputLocation)
        .jobName(jobName)
        .filterArn(filterArn)
        .roleArn(roleArn)
        .batchInferenceJobConfig(jobConfig) // Optional
        .build();

    batchInferenceJobArn =
    personalizeClient.createBatchInferenceJob(createBatchInferenceJobRequest)
        .batchInferenceJobArn();

    DescribeBatchInferenceJobRequest describeBatchInferenceJobRequest =
    DescribeBatchInferenceJobRequest.builder()
        .batchInferenceJobArn(batchInferenceJobArn)
        .build();

    long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

    // wait until the batch inference job is complete.
    while (Instant.now().getEpochSecond() < maxTime) {

        BatchInferenceJob batchInferenceJob = personalizeClient
            .describeBatchInferenceJob(describeBatchInferenceJobRequest)
            .batchInferenceJob();

        status = batchInferenceJob.status();
        System.out.println("Batch inference job status: " + status);

        if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
            break;
        }
        try {
            Thread.sleep(waitInMilliseconds);
        } catch (InterruptedException e) {
            System.out.println(e.getMessage());
        }
    }
    return batchInferenceJobArn;
```

```
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { CreateBatchInferenceJobCommand } from
    "@aws-sdk/client-personalize";
import { personalizeClient } from "../libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the batch inference job's parameters.

export const createBatchInferenceJobParam = {
    jobName: 'JOB_NAME',
    jobInput: { /* required */
        s3DataSource: { /* required */
            path: 'INPUT_PATH', /* required */
            // kmsKeyArn: 'INPUT_KMS_KEY_ARN' /* optional */
        }
    },
    jobOutput: { /* required */
        s3DataDestination: { /* required */
            path: 'OUTPUT_PATH', /* required */
            // kmsKeyArn: 'OUTPUT_KMS_KEY_ARN' /* optional */
        }
    },
    roleArn: 'ROLE_ARN', /* required */
    solutionVersionArn: 'SOLUTION_VERSION_ARN', /* required */
    numResults: 20 /* optional integer*/
};

export const run = async () => {
    try {
        const response = await personalizeClient.send(new
            CreateBatchInferenceJobCommand(createBatchInferenceJobParam));
        console.log("Success", response);
    }
}
```



```

    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();

```

배치 작업을 처리하는 데 시간이 걸릴 수 있습니다. [DescribeBatchInferenceJob](#)을 호출하고 `batchRecommendationsJobArn`을 입력 파라미터로 전달하여 작업 상태를 확인할 수 있습니다. 또한 호출을 통해 사용자 AWS 환경의 모든 Amazon Personalize 배치 추론 작업을 나열할 수 있습니다. [ListBatchInferenceJobs](#)

테마를 생성하는 배치 추론 작업 생성

유사한 항목에 대한 테마를 생성하려면 유사-항목 레시피를 사용해야 하며, 항목 데이터 세트에는 텍스트 필드와 항목 이름 데이터 열이 있어야 합니다. 테마가 있는 추천에 대한 자세한 내용은 [콘텐츠 생성 기에서 테마가 있는 배치 추천](#) 섹션을 참조하세요.

다음 코드는 테마가 있는 추천을 생성하는 배치 추론 작업을 만듭니다. `batchInferenceJobMode` 설정을 "THEME_GENERATION"으로 둡니다. 항목 이름 데이터를 저장하는 열의 이름으로 `COLUMN_NAME`을 바꿉니다.

```

import boto3

personalize_rec = boto3.client(service_name='personalize')

personalize_rec.create_batch_inference_job (
    solutionVersionArn = "Solution version ARN",
    jobName = "Batch job name",
    roleArn = "IAM service role ARN",
    filterArn = "Filter ARN",
    batchInferenceJobMode = "THEME_GENERATION",
    themeGenerationConfig = {
        "fieldsForThemeGeneration": {
            "itemName": "COLUMN_NAME"
        }
    },
    jobInput =
        {"s3DataSource": {"path": "s3://<name of your S3 bucket>/<folder name>/<input
JSON file name>.json"}},

```

```

jobOutput =
    {"s3DataDestination": {"path": "s3://<name of your S3 bucket>/<output folder
name>/"}}
)

```

배치 추론 작업 출력 예제

배치 추론 작업을 생성하면 작업이 Amazon S3 버킷에서 배치 입력 데이터를 가져오고, 솔루션 버전을 사용하여 항목 추천을 생성하고, 권장 사항을 Amazon S3 버킷에 JSON 형식으로 내보냅니다.

다음 섹션에는 레시피 유형별로 배치 추론 작업에 대한 출력 파일 예제가 나열되어 있습니다. 지금-유행 또는 차선택-작업 레시피로는 배치 추천을 받을 수 없습니다.

주제

- [USER_PERSONALIZATION 레시피](#)
- [POPULAR_ITEMS 레시피](#)
- [PERSONALIZED_RANKING 레시피](#)
- [RELATED_ITEMS 레시피](#)

USER_PERSONALIZATION 레시피

다음은 USER_PERSONALIZATION 레시피에 대한 출력 JSON 파일의 예제입니다.

```

{"input":{"userId":"4638"},"output":{"recommendedItems":
["63992","115149","110102","148626","148888","31685","102445","69526","92535","143355","62374",
[0.0152238,0.0069081,0.0068222,0.006394,0.0059746,0.0055851,0.0049357,0.0044644,0.0042968,0.004
{"input":{"userId":"663"},"output":{"recommendedItems":
["368","377","25","780","1610","648","1270","6","165","1196","1097","300","1183","608","104","4
[0.0406197,0.0372557,0.0254077,0.0151975,0.014991,0.0127175,0.0124547,0.0116712,0.0091098,0.008
{"input":{"userId":"3384"},"output":{"recommendedItems":
["597","21","223","2144","208","2424","594","595","920","104","520","367","2081","39","1035","2
[0.0241061,0.0119394,0.0118012,0.010662,0.0086972,0.0079428,0.0073218,0.0071438,0.0069602,0.005
...

```

POPULAR_ITEMS 레시피

다음 예제는 Popularity-Count 레시피의 출력 JSON 파일 형식을 보여줍니다. Trending-Now 레시피로는 배치 추천을 받을 수 없습니다.

```

{"input": {"userId": "12"}, "output": {"recommendedItems": ["105", "106", "441"]}}

```

```

{"input": {"userId": "105"}, "output": {"recommendedItems": ["105", "106", "441"]}}
{"input": {"userId": "41"}, "output": {"recommendedItems": ["105", "106", "441"]}}
...

```

PERSONALIZED_RANKING 레시피

다음 예제는 USER_PERSONALIZATION 레시피에 대한 출력 JSON 파일을 보여줍니다.

```

{"input":{"userId":"891","itemList":["27","886","101"]},"output":{"recommendedItems":
["27","101","886"],"scores":[0.48421,0.28133,0.23446]}}
{"input":{"userId":"445","itemList":["527","55","901"]},"output":{"recommendedItems":
["901","527","55"],"scores":[0.46972,0.31011,0.22017]}}
{"input":{"userId":"71","itemList":["29","351","199"]},"output":{"recommendedItems":
["351","29","199"],"scores":[0.68937,0.24829,0.06232]}}
...

```

RELATED_ITEMS 레시피

다음 예제는 Popularity-Count 레시피의 출력 JSON 파일 형식을 보여줍니다.

```

{"input": {"itemId": "105"}, "output": {"recommendedItems": ["106", "107", "49"]}}
{"input": {"itemId": "106"}, "output": {"recommendedItems": ["105", "107", "49"]}}
{"input": {"itemId": "441"}, "output": {"recommendedItems": ["2", "442", "435"]}}
...

```

다음 예제는 테마가 있는 유사-항목 레시피의 출력 JSON 파일 형식을 보여줍니다. 테마가 있는 추천에 대한 자세한 내용은 [콘텐츠 생성기에서 테마가 있는 배치 추천](#) 섹션을 참조하세요.

```

{"input":{"itemId":"40"},"output":{"recommendedItems":
["36","50","44","22","21","29","3","1","2","39"],"theme":"Movies
with a strong female lead","itemsThemeRelevanceScores":
[0.19994527,0.183059963,0.17478035,0.1618133,0.1574806,0.15468733,0.1499242,0.14353688,0.135314
{"input":{"itemId":"43"},"output":{"recommendedItems":
["50","21","36","3","17","2","39","1","10","5"],"theme":"The best movies of
1995","itemsThemeRelevanceScores":
[0.184988,0.1795761,0.11143453,0.0989443,0.08258403,0.07952615,0.07115086,0.0621634,-0.138913,-
...

```

사용자 세그먼트 가져오기

사용자 세그먼트를 가져오려면 배치 세그먼트 작업을 사용합니다. 배치 세그먼트 작업은 Amazon S3 버킷에서 배치 입력 데이터를 가져오고, SER_SEGMENTATION 레시피로 학습시킨 솔루션 버전을 사용하여 각 입력 데이터 행에 대해 사용자 세그먼트를 생성하는 도구입니다.

레시피에 따라 입력 데이터는 JSON 형식의 항목 또는 항목 메타데이터 속성 목록입니다. 항목 속성의 경우 입력 데이터에 표현식을 포함시키면 여러 메타데이터 속성을 기반으로 사용자 세그먼트를 생성할 수 있습니다. 배치 세그먼트 작업은 사용자 세그먼트를 출력 Amazon S3 버킷으로 내보냅니다. 각 사용자 세그먼트는 각 사용자가 입력 데이터의 항목과 상호 작용할 확률을 기준으로 내림차순으로 정렬됩니다.

사용자 세그먼트를 생성할 때 Amazon Personalize는 대량 및 개별 가져오기의 데이터 세트에 있는 데이터를 고려합니다.

- 대량 데이터의 경우 Amazon Personalize는 마지막 전체 솔루션 버전 학습에 있는 대량 데이터만 사용하여 세그먼트를 생성합니다. 또한 FULL 가져오기 모드로 가져온 대량 데이터만 사용합니다(기존 데이터 대체).
- 개별 데이터 가져오기 작업의 데이터의 경우 Amazon Personalize는 마지막 전체 솔루션 버전 학습 시 제공된 데이터를 사용하여 사용자 세그먼트를 생성합니다. 새 레코드가 사용자 세그먼트에 영향을 미치도록 하려면 새 솔루션 버전을 생성한 다음 배치 세그먼트 작업을 생성합니다.

사용자 세그먼트 생성은 다음과 같이 작동합니다.

1. 입력 데이터를 JSON 형식으로 준비하고 Amazon S3 버킷에 업로드합니다. 입력 데이터의 형식은 사용하는 레시피와 생성 중인 작업에 따라 달라집니다. [사용자 세그먼트를 위한 입력 데이터 준비 단원을 참조하세요.](#)
2. 출력 데이터를 저장할 별도의 위치(다른 폴더 또는 다른 Amazon S3 버킷)를 만듭니다.
3. 배치 세그먼트 작업을 생성합니다. [배치 세그먼트 작업 생성](#) 섹션을 참조하십시오.
4. 배치 세그먼트 작업이 완료되면 Amazon S3의 출력 위치에서 사용자 세그먼트를 검색합니다.

주제

- [지침 및 요구 사항](#)
- [사용자 세그먼트를 위한 입력 데이터 준비](#)
- [배치 세그먼트 작업 생성](#)

• [배치 세그먼트 작업 출력 예제](#)

지침 및 요구 사항

배치 세그먼트를 일괄적으로 가져오기 위한 지침 및 요구 사항은 다음과 같습니다.

- USER_SEGMENTATION 레시피를 사용해야 합니다.
- Amazon Personalize IAM 서비스 역할에 Amazon S3 버킷의 파일을 읽고 추가할 수 있는 권한이 필요합니다. 이러한 권한 부여에 대한 자세한 내용은 [배치 워크플로우에 대한 서비스 역할 정책](#) 단원을 참조하세요. 버킷 권한에 대한 자세한 내용은 Amazon Simple Storage Service 개발자 안내서의 [사용자 정책 예제](#) 단원을 참조하세요.

암호화에 AWS Key Management Service (AWS KMS) 를 사용하는 경우 Amazon Personalize 와 Amazon Personalize IAM 서비스 역할에 키를 사용할 권한을 부여해야 합니다. 자세한 설명은 [Amazon Personalize에 AWS KMS 키 사용 권한 부여](#) 섹션을 참조하세요.

- 배치 추론 작업을 생성하기 전에 사용자 지정 솔루션과 솔루션 버전을 생성해야 합니다. 그러나 Amazon Personalize 캠페인은 생성할 필요가 없습니다. 도메인 데이터 세트 그룹을 생성한 경우에도 사용자 지정 리소스를 생성할 수 있습니다.
- 입력 데이터는 [사용자 세그먼트를 위한 입력 데이터 준비](#)에 설명된 대로 형식을 지정해야 합니다.
- 항목-속성-선호도 레시피를 사용하는 경우 입력 데이터의 속성에 제품 설명과 같은 구조화되지 않은 텍스트 항목 메타데이터를 포함할 수 없습니다.
- 자리 표시자 파라미터가 있는 필터를 사용하는 경우 filterValues 객체의 입력 데이터에 파라미터 값을 포함시켜야 합니다. 자세한 설명은 [입력 JSON에서 필터 값 제공](#) 섹션을 참조하세요.
- 출력 데이터에는 입력 데이터와 다른 위치(폴더 또는 다른 Amazon S3 버킷)를 사용하는 것이 좋습니다.

사용자 세그먼트를 위한 입력 데이터 준비

배치 세그먼트 작업은 솔루션 버전을 사용하여 입력 JSON 파일에 제공한 데이터를 기반으로 사용자 세그먼트를 만듭니다. 사용자 세그먼트를 가져오려면 먼저 JSON 파일을 준비해서 Amazon S3 버킷에 업로드해야 합니다. Amazon S3 버킷에 출력 폴더를 생성하거나 별도의 출력 Amazon S3 버킷을 사용하는 것이 좋습니다. 그런 다음 동일한 입력 데이터 위치를 사용하여 여러 배치 추론 작업을 실행할 수 있습니다.

자리 표시자 파라미터가 포함된 필터(예: \$GENRE)를 사용하는 경우 filterValues 객체의 파라미터 값을 입력 JSON에 제공해야 합니다. 자세한 내용은 [입력 JSON에서 필터 값 제공](#) 섹션을 참조하세요.

데이터를 준비하고 가져오려면

1. 솔루션에서 사용하는 레시피에 따라 배치 입력 데이터의 형식을 지정합니다. 입력 데이터 요소를 새 줄로 구분합니다. 입력 데이터는 항목 ID 목록(항목 번호도) 또는 항목 속성(항목-속성-번호도)입니다.

항목 속성의 경우 입력 데이터에 AND 연산자와 함께 논리적 표현식을 포함하여 쿼리당 여러 항목 또는 속성에 대한 사용자를 가져올 수 있습니다. 자세한 설명은 [품목-속성-친화성 레시피의 품목 속성 지정](#) 섹션을 참조하세요.

두 레시피의 입력 데이터 예시는 [여기](#)를 참조하십시오. [배치 세그먼트 작업 입력 및 출력 JSON 예제](#)

2. Amazon S3 버킷의 입력 폴더에 입력 JSON을 업로드합니다. 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 [드래그 앤 드롭을 사용하여 파일 및 폴더 업로드](#) 단원을 참조하세요.
3. 출력 데이터를 저장할 별도의 위치(폴더 또는 다른 Amazon S3 버킷)를 만듭니다. 출력 JSON을 위한 별도의 위치를 생성하면 동일한 입력 데이터 위치로 여러 배치 세그먼트 작업을 실행할 수 있습니다.

입력 데이터를 준비하고 Amazon S3 버킷에 업로드했으면 배치 세그먼트 작업으로 사용자 세그먼트를 생성할 준비가 된 것입니다. 자세한 설명은 [배치 세그먼트 작업 생성](#) 섹션을 참조하세요.

주제

- [품목-속성-친화성 레시피의 품목 속성 지정](#)
- [배치 세그먼트 작업 입력 및 출력 JSON 예제](#)

품목-속성-친화성 레시피의 품목 속성 지정

품목-속성-친화성 레시피를 사용하는 경우 입력 데이터는 품목 속성 목록입니다. 서로 다른 메타데이터 열을 혼합할 수 있습니다. 예를 들어 한 행은 숫자 열이고 다음 행은 범주형 열일 수 있습니다. 비정형 텍스트 항목 메타데이터를 항목 속성으로 사용할 수 없습니다.

입력 항목 메타데이터에는 AND 연산자를 사용하여 여러 속성에 대한 사용자 세그먼트를 가져오는 논리적 표현식이 포함될 수 있습니다. 예를 들어 입력 데이터의 한 줄은 {"itemAttributes": "ITEMS.genres = "\Comedy\" AND ITEMS.genres = "\Action\""} 또는 {"itemAttributes": "ITEMS.genres = "\Comedy\" AND ITEMS.audience = "\teen\""}일 수 있습니다.

AND 연산자로 두 속성을 결합하면 사용자의 상호작용 이력을 기반으로 두 속성이 모두 있는 항목과 상호작용할 가능성이 높은 사용자가 포함된 사용자 세그먼트가 생성됩니다. 문자열 같음에 IN 연산자를 사용하는 필터 표현식과 달리 배치 세그먼트 입력 표현식은 문자열 일치의 경우 같음 = 기호만 지원합니다.

배치 세그먼트 작업 입력 및 출력 JSON 예제

배치 세그먼트 작업의 경우 입력 데이터는 itemID 목록(항목-번호도 레시피) 또는 항목 속성(항목-속성-번호도) 중 하나여야 합니다. 입력 데이터의 각 라인은 별도의 쿼리입니다. 각 사용자 세그먼트는 각 사용자가 인벤토리의 항목과 상호작용할 확률을 기준으로 내림차순으로 정렬됩니다.

자리 표시자 파라미터가 포함된 필터(예: \$GENRE)를 사용하는 경우 filterValues 객체의 파라미터 값을 입력 JSON에 제공해야 합니다. 자세한 내용은 [입력 JSON에서 필터 값 제공](#) 섹션을 참조하세요.

다음은 레시피별로 정리된 배치 세그먼트 작업에 대한 올바른 형식의 JSON 입력 및 출력 예제입니다.

항목-친화도

Input

입력 데이터가 최대 500개의 항목을 사용할 수 있습니다. 다음과 같이 각 itemId를 새 줄로 구분합니다.

```

{"itemId": "105"}
{"itemId": "106"}
{"itemId": "441"}
...

```

Output

```

{"input": {"itemId": "105"}, "output": {"recommendedUsers": ["106", "107", "49"]}}
{"input": {"itemId": "106"}, "output": {"recommendedUsers": ["105", "107", "49"]}}
{"input": {"itemId": "441"}, "output": {"recommendedUsers": ["2", "442", "435"]}}
...

```

항목-속성-친화도

Input

입력 데이터에는 최대 10개의 쿼리가 포함될 수 있으며, 각 쿼리는 하나 이상의 비텍스트 항목 속성입니다. 다음과 같이 각 속성 또는 속성 식을 새 줄로 구분합니다.

```

{"itemAttributes": "ITEMS.genres = \"Comedy\" AND ITEMS.genres = \"Action\""}
{"itemAttributes": "ITEMS.genres = \"Comedy\""}
{"itemAttributes": "ITEMS.genres = \"Horror\" AND ITEMS.genres = \"Action\""}
...

```

Output

```

{"itemAttributes": "ITEMS.genres = \"Comedy\" AND ITEMS.genres = \"Action\"",
 "output": {"recommendedUsers": ["25", "78", "108"]}}
{"itemAttributes": "ITEMS.genres = \"Adventure\"", "output": {"recommendedUsers":
 ["87", "31", "129"]}}
{"itemAttributes": "ITEMS.genres = \"Horror\" AND ITEMS.genres = \"Action\"",
 "output": {"recommendedUsers": ["8", "442", "435"]}}
...

```

배치 세그먼트 작업 생성

사용자_세분화 레시피를 사용한 경우, 일괄 세그먼트 작업을 생성하여 솔루션 버전으로 사용자 세그먼트를 가져올 수 있습니다. 각 사용자 세그먼트는 각 사용자가 인벤토리의 항목과 상호작용할 확률을 기준으로 내림차순으로 정렬됩니다. 레시피에 따라 입력 데이터는 JSON 형식의 항목 목록([항목-친화도 레시피](#)) 또는 항목 속성([항목-속성-친화도 레시피](#))이어야만 합니다. Amazon Personalize 콘솔, AWS Command Line Interface (AWS CLI) 또는 AWS SDK를 사용하여 배치 세그먼트 작업을 생성할 수 있습니다.

배치 세그먼트 작업을 생성할 때 입력 및 출력 위치에 S3 경로를 지정합니다. S3는 접두사를 기반으로 합니다. 입력 데이터 위치에 접두사를 제공하는 경우, Personalize는 해당 접두사와 일치하는 모든 파일을 입력 데이터로 사용합니다. 예를 들어, 사용자가 `s3://<name of your S3 bucket>/folderName`을 제공하고 버킷에 경로가 `s3://<name of your S3 bucket>/folderName_test`인 폴더가 있는 경우, Personalize는 두 폴더의 모든 파일을 입력 데이터로 사용합니다. 특정 폴더 내의 파일만을 입력 데이터로 사용하려면 S3 경로 끝에 접두사 구분 기호를 붙입니다 (예: `/: s3://<name of your S3 bucket>/folderName/`). S3에서 객체를 구성하는 방법에 대한 자세한 내용은 [객체 구성, 나열 및 객체 작업](#)을 참조하세요.

주제

- [배치 세그먼트 작업 생성\(콘솔\)](#)
- [배치 세그먼트 작업 생성\(AWS CLI\)](#)
- [일괄 세그먼트 작업 \(AWS SDK\) 생성](#)

배치 세그먼트 작업 생성(콘솔)

[배치 추천을 위한 입력 데이터 준비](#) 작업을 완료했으면 배치 세그먼트 작업을 생성할 준비가 된 것입니다. 이 절차에서는 USER_SEGEMENTATION 레시피를 사용하여 솔루션과 솔루션 버전(학습된 모델)을 이미 생성한 것으로 가정합니다.

배치 세그먼트 작업을 생성하려면(콘솔)

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터세트 그룹 페이지에서 데이터세트 그룹을 선택합니다.
3. 탐색 창에서 배치 세그먼트 작업을 선택한 다음 배치 세그먼트 작업 생성을 선택합니다.
4. 배치 세그먼트 작업 세부 정보에서 배치 세그먼트 작업 이름의 경우, 배치 세그먼트 작업의 이름을 지정합니다.
5. 솔루션의 경우, 솔루션을 선택한 다음 추천을 생성하는 데 사용하려는 솔루션 버전 ID를 선택합니다. USER_SEGEMENTATION 레시피를 사용한 경우에만 배치 세그먼트 작업을 생성할 수 있습니다.
6. 사용자 수의 경우, Personalize가 각 사용자 세그먼트에 대해 생성하는 사용자의 수를 원하는 대로 지정합니다. 기본값은 25입니다. 최대값은 5백만 명입니다.
7. 입력 소스의 경우, 입력 파일에 대한 S3 경로를 지정하거나 S3 찾아보기를 사용하여 S3 버킷을 선택합니다.

다음 구문 사용: **s3://<name of your S3 bucket>/<folder name>/<input JSON file name>.json**

입력 데이터는 솔루션에서 사용하는 레시피에 맞는 올바른 형식이어야 합니다. 입력 데이터 예제는 [배치 세그먼트 작업 입력 및 출력 JSON 예제](#) 단원을 참조하세요.

8. 출력 대상의 경우, 출력 위치에 대한 경로를 지정하거나 S3 찾아보기를 사용하여 S3 버킷을 선택합니다. 출력 데이터에 대한 다른 위치(폴더 또는 다른 S3 버킷)를 사용하는 것이 좋습니다.

다음 구문 사용: **s3://<name of your S3 bucket>/<output folder name>/**

9. IAM 역할의 경우, 다음 중 하나를 선택합니다.
 - 새 서비스 역할 생성 및 사용을 선택하고 서비스 역할 이름을 입력하여 새 역할을 생성합니다. 또는,
 - 올바른 권한이 있는 역할을 이미 생성한 경우, 기존 서비스 역할 사용을 선택하고 IAM 역할을 선택합니다.

- 사용하는 역할에는 각각 입력 및 출력 S3 버킷에 대한 읽기 및 쓰기 액세스 권한이 있어야 합니다.
10. 필터 구성 경우, 원하는 필터를 선택하여 사용자 세그먼트에 필터를 적용할 수 있습니다. 필터가 자리표시자 파라미터를 사용하는 경우, 파라미터 값이 입력 JSON에 포함되어 있는지 확인하세요. 자세한 내용은 [입력 JSON에서 필터 값 제공](#) 단원을 참조하세요.
 11. 태그의 경우, 원하는 태그를 추가할 수 있습니다. Personalize 리소스 태그 지정에 대한 자세한 내용은 [Personalize 리소스에 태그 지정](#) 단원을 참조하세요.
 12. 배치 세그먼트 작업 생성을 선택합니다. 배치 세그먼트 작업 생성이 시작되고 배치 세그먼트 작업 페이지가 나타나고 배치 세그먼트 작업 세부 정보) 단원이 표시됩니다.
 13. 배치 세그먼트 작업의 상태가 활성으로 변경되면 지정된 출력 S3 버킷에서 작업 출력을 찾을 수 있습니다. 출력 파일 이름의 형식은 *input-name.out*이 됩니다.

배치 세그먼트 작업 생성(AWS CLI)

[배치 추천을 위한 입력 데이터 준비](#) 작업을 완료했으면 다음 `create-batch-segment-job` 코드를 사용하여 배치 세그먼트 작업을 생성할 준비가 된 것입니다. 작업 이름을 지정하고, Solution version ARN을 솔루션 버전의 리소스 이름(ARN)으로 대체하고, IAM service role ARN을 설정 중에 Personalize용으로 생성한 IAM 서비스 역할의 ARN으로 대체하세요. 이 역할에는 각각 입력 및 출력 S3 버킷에 대한 읽기 및 쓰기 액세스 권한이 있어야 합니다. `num-results`의 경우, Personalize가 각 입력 데이터 라인에 대해 예측하기를 원하는 사용자의 수를 지정합니다. 기본값은 25입니다. 최대 값은 5백만 명입니다. 원하는 대로 `filter-arn`을 제공하여 사용자 세그먼트를 필터링합니다. 필터가 자리표시자 파라미터를 사용하는 경우, 파라미터 값이 입력 JSON에 포함되어 있는지 확인하세요. 자세한 내용은 [배치 추천 및 사용자 세그먼트\(사용자 지정 리소스\) 필터링](#) 단원을 참조하세요.

S3 input path 및 S3 output path를 입력 파일 및 출력 위치에 대한 S3 경로로 대체합니다. 출력 데이터에 대한 다른 위치(폴더 또는 다른 S3 버킷)를 사용하는 것이 좋습니다. 입력 및 출력 위치에는 구문: `s3://<name of your S3 bucket>/<folder name>/<input JSON file name>.json` 및 `s3://<name of your S3 bucket>/<output folder name>/`을 사용합니다.

```
aws personalize create-batch-segment-job \
  --job-name Job name \
  --solution-version-arn Solution version ARN \
  --num-results The number of predicted users \
  --filter-arn Filter ARN \
  --job-input s3DataSource={path=s3://S3 input path} \
  --job-output s3DataDestination={path=s3://S3 output path} \
  --role-arn IAM service role ARN
```

```
{
  "batchSegmentJobArn": "arn:aws:personalize:us-west-2:acct-id:batch-segment-job/
batchSegmentJobName"
}
```

일괄 세그먼트 작업 (AWS SDK) 생성

[배치 추천을 위한 입력 데이터 준비](#) 작업을 완료했다면 CreateBatchSegmentJob 작업을 사용하여 배치 세그먼트 작업을 생성할 준비가 된 것입니다. 다음 코드는 배치 세그먼트 작업을 생성하는 방법을 보여줍니다. 작업 이름을 지정하고, 사용할 솔루션 버전의 리소스 이름(ARN)을 지정하고, Personalize IAM 역할의 ARN을 지정하고, 입력 파일 및 출력 위치에 대한 S3 경로를 지정합니다. IAM 서비스 역할에는 각각 입력 및 출력 S3 버킷에 대한 읽기 및 쓰기 액세스 권한이 있어야 합니다.

출력 데이터에 대한 다른 위치(폴더 또는 다른 S3 버킷)를 사용하는 것이 좋습니다. 입력 및 출력 위치에는 구문: **s3://<name of your S3 bucket>/<folder name>/<input JSON file name>.json** 및 **s3://<name of your S3 bucket>/<output folder name>/**을 사용합니다.

numResults의 경우, Personalize가 각 입력 데이터 라인에 대해 예측하기를 원하는 사용자의 수를 지정합니다. 기본값은 25입니다. 최대값은 5백만 명입니다. 원하는 대로 filterArn을 제공하여 사용자 세그먼트를 필터링합니다. 필터가 자리표시자 파라미터를 사용하는 경우, 파라미터 값이 입력 JSON에 포함되어 있는지 확인하세요. 자세한 내용은 [배치 추천 및 사용자 세그먼트\(사용자 지정 리소스\) 필터링](#) 단원을 참조하세요.

SDK for Python (Boto3)

```
import boto3

personalize_rec = boto3.client(service_name='personalize')

personalize_rec.create_batch_segment_job (
    solutionVersionArn = "Solution version ARN",
    jobName = "Job name",
    numResults = 25,
    filterArn = "Filter ARN",
    roleArn = "IAM service role ARN",
    jobInput =
        {"s3DataSource": {"path": "s3://<name of your S3 bucket>/<folder name>/<input
JSON file name>.json"}},
    jobOutput =
        {"s3DataDestination": {"path": "s3://<name of your S3 bucket>/<output folder
name>/"}}}
```

)

SDK for Java 2.x

```

public static String createBatchSegmentJob(PersonalizeClient personalizeClient,
                                           String solutionVersionArn,
                                           String jobName,
                                           String filterArn,
                                           int numResults,
                                           String
s3InputDataSourcePath,
                                           String
s3DataDestinationPath,
                                           String roleArn,
                                           String explorationWeight,
                                           String
explorationItemAgeCutOff) {

    long waitInMilliseconds = 60 * 1000;
    String status;
    String batchSegmentJobArn;

    try {
        // Set up data input and output parameters.
        S3DataConfig inputSource = S3DataConfig.builder()
            .path(s3InputDataSourcePath)
            .build();
        S3DataConfig outputDestination = S3DataConfig.builder()
            .path(s3DataDestinationPath)
            .build();

        BatchSegmentJobInput jobInput = BatchSegmentJobInput.builder()
            .s3DataSource(inputSource)
            .build();
        BatchSegmentJobOutput jobOutputLocation = BatchSegmentJobOutput.builder()
            .s3DataDestination(outputDestination)
            .build();

        CreateBatchSegmentJobRequest createBatchSegmentJobRequest =
        CreateBatchSegmentJobRequest.builder()
            .solutionVersionArn(solutionVersionArn)
            .filterArn(filterArn)

```

```
        .jobInput(jobInput)
        .jobOutput(jobOutputLocation)
        .jobName(jobName)
        .numResults(numResults)
        .roleArn(roleArn)
        .build();

    batchSegmentJobArn =
personalizeClient.createBatchSegmentJob(createBatchSegmentJobRequest)
        .batchSegmentJobArn();
    DescribeBatchSegmentJobRequest describeBatchSegmentJobRequest =
DescribeBatchSegmentJobRequest.builder()
        .batchSegmentJobArn(batchSegmentJobArn)
        .build();

    long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

    // wait until the batch segment job is complete.
    while (Instant.now().getEpochSecond() < maxTime) {

        BatchSegmentJob batchSegmentJob = personalizeClient
            .describeBatchSegmentJob(describeBatchSegmentJobRequest)
            .batchSegmentJob();

        status = batchSegmentJob.status();
        System.out.println("batch segment job status: " + status);

        if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
            break;
        }
        try {
            Thread.sleep(waitInMilliseconds);
        } catch (InterruptedException e) {
            System.out.println(e.getMessage());
        }
    }
    return batchSegmentJobArn;

} catch (PersonalizeException e) {
    System.out.println(e.awsErrorDetails().errorMessage());
}
return "";
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { CreateBatchSegmentJobCommand } from
  "@aws-sdk/client-personalize";
import { personalizeClient } from "../libs/personalizeClients.js";

// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the batch segment job's parameters.

export const createBatchSegmentJobParam = {
  jobName: 'NAME',
  jobInput: { /* required */
    s3DataSource: { /* required */
      path: 'INPUT_PATH', /* required */
      // kmsKeyArn: 'INPUT_KMS_KEY_ARN' /* optional */
    }
  },
  jobOutput: { /* required */
    s3DataDestination: { /* required */
      path: 'OUTPUT_PATH', /* required */
      // kmsKeyArn: 'OUTPUT_KMS_KEY_ARN' /* optional */
    }
  },
  roleArn: 'ROLE_ARN', /* required */
  solutionVersionArn: 'SOLUTION_VERSION_ARN', /* required */
  numResults: 20 /* optional */
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
      CreateBatchSegmentJobCommand(createBatchSegmentJobParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

배치 작업을 처리하는 데 시간이 걸릴 수 있습니다. [DescribeBatchSegmentJob](#)을 호출하고 `batchSegmentJobArn`을 입력 파라미터로 전달하여 작업 상태를 확인할 수 있습니다. 또한 호출을 통해 사용자 AWS 환경의 모든 Amazon Personalize 배치 세그먼트 작업을 나열할 수 있습니다. [ListBatchSegmentJobs](#)

배치 세그먼트 작업 출력 예제

배치 세그먼트 작업은 Amazon S3 버킷에서 배치 입력 데이터를 가져오고, `USER_SEGMENTATION` 레시피로 학습된 솔루션 버전을 사용하여 사용자 세그먼트를 생성하고, 세그먼트를 Amazon S3 버킷으로 내보냅니다.

다음 단원에는 배치 세그먼트 작업에 대한 올바른 형식의 JSON 출력 예제가 레시피별로 나열되어 있습니다.

주제

- [항목-친화도](#)
- [항목-속성-친화도](#)

항목-친화도

다음 예제는 항목-친화도 레시피의 출력 JSON 파일 형식을 보여줍니다.

```
{ "input": { "itemId": "105"}, "output": { "recommendedUsers": ["106", "107", "49"]}}
{ "input": { "itemId": "106"}, "output": { "recommendedUsers": ["105", "107", "49"]}}
{ "input": { "itemId": "441"}, "output": { "recommendedUsers": ["2", "442", "435"]}}
...
```

항목-속성-친화도

다음 예제는 항목-속성-친화도 레시피의 출력 JSON 파일 형식을 보여줍니다.

```
{ "itemAttributes": "ITEMS.genres = \"Comedy\" AND ITEMS.genres = \"Action\"", "output": { "recommendedUsers": ["25", "78", "108"]}}
{ "itemAttributes": "ITEMS.genres = \"Adventure\"", "output": { "recommendedUsers": ["87", "31", "129"]}}
{ "itemAttributes": "ITEMS.genres = \"Horror\" AND ITEMS.genres = \"Action\"", "output": { "recommendedUsers": ["8", "442", "435"]}}
...
```

추천 관련성 유지

관련 추천은 카탈로그가 커질수록 애플리케이션에 대한 사용자 참여, 클릭률 및 전환율을 높일 수 있습니다. Personalize 추천의 관련성을 유지하고 개선하려면 데이터와 사용자 지정 리소스를 최신 상태로 유지합니다. 이를 통해 Personalize는 사용자의 최근 행동을 학습하여 최신 항목을 추천에 포함시킬 수 있습니다.

주제

- [데이터세트를 최신 상태로 유지](#)
- [도메인 추천자 유지 관리](#)
- [사용자 지정 솔루션 유지 관리](#)

데이터세트를 최신 상태로 유지

카탈로그가 커지면 대량 또는 개별 데이터 가져오기 작업을 통해 과거 데이터를 업데이트합니다. 과거 데이터 가져오기에 대한 자세한 내용은 [단계 2: 데이터 준비 및 가져오기](#) 단원을 참조하세요. 모델을 학습시킨 후 가져오는 데이터가 추천에 미치는 영향에 대한 자세한 내용은 [새 데이터가 실시간 추천에 영향을 미치는 방식](#) 단원을 참조하세요.

개인 맞춤형 실시간 추천을 제공하는 사용 사례와 레시피의 경우 항목 상호 작용 데이터 세트를 사용자 행동에 따른 최신 상태로 유지합니다. 이벤트 트래커 및 PutEvents API 작업을 통해 항목 상호 작용을 기록하면 됩니다. Personalize는 사용자가 카탈로그와 상호작용할 때 가장 최근 활동을 기반으로 추천을 업데이트합니다. 실시간 개인화에 대한 자세한 내용은 [실시간 개인 맞춤](#) 단원을 참조하세요. 실시간 이벤트 기록에 대한 자세한 내용은 [이벤트 기록](#) 단원을 참조하세요.

도메인 추천자 유지 관리

Personalize는 7일마다 추천자를 지원하는 모델을 자동으로 재학습합니다. 이는 데이터세트의 전체 데이터를 기반으로 완전히 새로운 모델을 생성하는 전체 재교육입니다. 학습에 사용된 열을 수정하면 Personalize이 추천자를 지원하는 모델의 전체 재학습을 자동으로 시작합니다.

- 가장 적합한 추천 제품 및 추천 제품 사용 사례의 경우 Personalize는 새 항목을 추천하도록 추천자를 업데이트합니다. 자동 업데이트는 사용자 행동을 통해 모델이 학습하는 전체 재학습에 해당되지 않습니다. 대신 자동 업데이트를 통해 Personalize는 추천자의 다음 전체 재학습 전에 새 항목을 추천에 포함시킬 수 있습니다. 자동 업데이트에 대한 자세한 내용은 [자동 업데이트](#) 단원을 참조하세요.

- 지금 유행 사용 사례를 사용하는 경우 Personalize는 2시간마다 상호작용 데이터를 자동으로 평가하고 유행 항목을 식별합니다. 추천자가 재학습을 받을 때까지 기다릴 필요가 없습니다.

추천자 재학습이 진행 중인 동안에도 추천자로부터 추천을 받을 수 있습니다. 재학습이 완료될 때까지 추천자는 이전 구성과 모델을 사용합니다. 업데이트를 추적하려면 Personalize 콘솔의 추천자 세부 정보 페이지에서 마지막 추천자 업데이트의 타임스탬프를 볼 수 있습니다. 또는 [DescribeRecommender](#) 작업의 latestRecommenderUpdate 세부 정보를 볼 수도 있습니다.

사용자 지정 솔루션 유지 관리

기본적으로 모든 새 솔루션은 자동 교육을 사용하여 7일마다 새 솔루션 버전을 생성합니다. 교육은 솔루션을 삭제할 때까지 계속됩니다.

솔루션을 만들 때는 자동 교육을 사용하여 솔루션 버전 생성을 관리하는 것이 좋습니다. 이렇게 하면 솔루션을 더 쉽게 유지 관리할 수 있습니다. 이를 통해 솔루션이 최신 데이터에서 학습하는 데 필요한 수동 교육을 생략할 수 있습니다. 자동 교육을 사용하지 않으면 솔루션이 최신 데이터에서 학습할 수 있도록 새 솔루션 버전을 수동으로 만들어야 합니다. 자동 학습 구성에 대한 자세한 내용은 [참조하십시오 자동 교육 구성](#).

교육 빈도는 비즈니스 요구 사항, 사용하는 레시피, 데이터 가져오기 빈도에 따라 달라집니다. 모든 레시피의 경우 최소 매주 교육을 받는 것이 좋습니다. 자동 훈련의 경우 이것이 기본 훈련 빈도입니다. 새 항목이나 동작을 자주 추가하는 경우 레시피에 따라 훈련 빈도를 높이는 것이 좋습니다.

- 사용자 맞춤화-v2, 사용자 맞춤화 또는 Next-Best-Action을 사용하는 경우 솔루션이 자동으로 업데이트되어 새 항목이나 권장 사항을 위한 조치를 고려합니다. 자동 업데이트는 자동 학습과 다릅니다. 자동 업데이트는 완전히 새로운 솔루션 버전을 만들지 않으며, 모델은 최신 데이터를 통해 학습하지 않습니다. 솔루션을 유지 관리하려면 훈련 빈도가 여전히 1주 이상이어야 합니다. 추가 지침 및 요구 사항을 포함한 자동 업데이트에 대한 자세한 내용은 [참조하십시오 자동 업데이트](#).
- 지금-유행을 사용하는 경우 Personalize는 구성 가능한 시간 간격으로 상호작용 데이터에서 가장 인기 있는 항목을 자동으로 식별합니다. Trending-Now는 대량 또는 스트리밍 상호 작용 데이터를 통해 마지막 교육 이후 추가된 항목을 추천할 수 있습니다. 훈련 빈도는 여전히 1주 이상이어야 합니다. 자세한 정보는 [지금-유행 레시피](#)를 참조하세요.
- 자동 업데이트 기능이 있는 레시피 또는 Trending-Now 레시피를 사용하지 않는 경우 Amazon Personalize는 다음 교육 이후에만 새 항목을 추천 대상으로 간주합니다. 예를 들어, Similar-Items 레시피를 사용하고 매일 새 항목을 추가하는 경우 이러한 항목이 같은 날 권장 사항에 표시되려면 일일 교육 빈도를 사용해야 합니다.

이벤트 기록

이벤트는 사용자와 카탈로그 간의 상호 작용입니다. 사용자가 품목을 구매하거나 비디오를 보는 것처럼 항목과의 상호 작용 또는 행동하는 것과 같은 작업일 수 있습니다. 신용카드를 신청하거나 멤버십 프로그램에 등록하는 경우를 예로 들 수 있습니다.

Amazon Personalize는 실시간 이벤트 데이터만 기반으로 하거나 과거 이벤트 데이터만 기반으로 하거나 이 둘을 혼합하여 추천할 수 있습니다. 고객이 작업 추천에 반응을 보이면 실시간 이벤트를 기록하세요. 이렇게 하면 상호 작용 데이터가 쌓이고 데이터를 최신으로 유지할 수 있습니다. 그리고 Amazon Personalize에 사용자의 현재 관심사를 알려주므로, 추천 관련성을 높일 수 있습니다.

도메인 사용 사례 또는 사용자 지정 레시피가 [실시간 개인 맞춤](#)을 지원하는 경우 Amazon Personalize는 실시간으로 이벤트를 사용하여 사용자의 변화하는 관심에 따라 추천을 업데이트하고 조정합니다.

실시간 이벤트를 기록하는 방법은 가져오는 상호 작용 데이터의 유형에 따라 다릅니다.

- 항목 상호 작용의 경우 [PutEvents](#) API 작업을 통해 실시간 이벤트를 기록합니다. Amazon Personalize는 이 데이터를 데이터 세트 그룹의 [항목 상호 작용 데이터 세트](#)에 추가합니다. 자세한 정보는 [이벤트 기록](#)을 참조하세요.
- 작업 상호 작용의 경우 [PutActionInteractions](#) API 작업을 통해 실시간 이벤트를 기록합니다. Amazon Personalize는 이 데이터를 데이터 세트 그룹의 [작업 상호 작용 데이터 세트](#)에 추가합니다. PERSONALIZED_ACTIONS 레시피만 작업 상호 작용 데이터를 사용합니다. 자세한 정보는 [작업 상호 작용 이벤트 기록](#)을 참조하세요.

주제

- [실시간 이벤트가 추천에 미치는 영향](#)
- [항목 상호 작용 이벤트 기록](#)
- [작업 상호 작용 이벤트 기록](#)
- [익명 사용자를 위한 이벤트 기록](#)
- [타사 이벤트 추적 서비스](#)
- [샘플 구현](#)

실시간 이벤트가 추천에 미치는 영향

레시피가 실시간 개인 맞춤을 지원하는 경우 추천자 또는 사용자 지정 캠페인을 생성한 후 Amazon Personalize는 가져온 후 몇 초 내에 기존 항목 또는 작업에 대해 기록된 새 이벤트 데이터를 사용합니다. 실시간 개인 맞춤을 지원하는 사용 사례 및 레시피는 다음과 같습니다.

- [추천 제품\(ECOMMERCE 사용 사례\)](#)
- [가장 적합한 추천 제품\(VIDEO_ON_DEMAND 사용 사례\)](#)
- [사용자 맞춤화-v2 레시피](#)
- [사용자-개인 맞춤 레시피](#)
- [맞춤형 랭킹-v2 레시피](#)
- [개인 맞춤형- 순위 레시피](#)
- [차선택-작업 레시피](#)

지금-유행 레시피를 사용하는 경우 Personalize는 구성 가능한 간격으로 새 이벤트 데이터의 항목을 자동으로 고려합니다. 새 솔루션 버전을 생성할 필요는 없습니다. 자세한 정보는 [지금-유행 레시피](#)를 참조하세요.

이벤트의 항목, 작업 또는 사용자가 신규인 경우 Amazon Personalize에서 데이터를 사용하는 방법은 사용 사례나 레시피에 따라 다릅니다. 자세한 정보는 [새 데이터가 실시간 추천에 영향을 미치는 방식을](#) 참조하세요.

항목 상호 작용 이벤트 기록

항목 상호 작용 이벤트는 카탈로그에 있는 항목과 사용자 간의 상호 작용입니다. 사용자가 신발을 구매하거나 영화를 보는 경우를 예로 들 수 있습니다.

고객에게 추천 항목을 보여주면서 실시간 항목 상호 작용 이벤트를 기록하세요. 이렇게 하면 상호 작용 데이터가 쌓이고 데이터를 최신으로 유지할 수 있습니다. 그리고 Amazon Personalize에 사용자의 현재 관심사를 알려주므로, 추천 관련성을 높일 수 있습니다.

[PutEvents](#) API 작업을 사용하여 항목 상호 작용 이벤트를 기록합니다. Amazon Personalize는 이 이벤트 데이터를 데이터 세트 그룹의 항목 상호 작용 데이터 세트에 추가합니다. 타임스탬프와 속성이 정확히 동일한 두 이벤트를 기록할 경우 Personalize는 이벤트 중 하나만 보관합니다. AWS SDK, AWS Amplify AWS Command Line Interface 또는 AWS CLI() 를 사용하여 항목 상호 작용 이벤트를 기록할 수 있습니다.

Apache Kafka를 사용하는 경우 Amazon Personalize용 Kafka 커넥터를 사용하여 Amazon Personalize로 항목 상호 작용을 실시간으로 스트리밍할 수 있습니다. 자세한 내용은 [personalize-kafka-connector Github 리포지토리의 Personalize용 Kafka 커넥터](#)를 참조하세요.

AWS Amplify에는 웹 클라이언트 애플리케이션의 항목 상호 작용 이벤트를 기록하기 위한 JavaScript 라이브러리와 서버 코드에 이벤트를 기록하는 라이브러리가 포함되어 있습니다. 자세한 내용은 [Amplify - 분석](#) 단원을 참조하세요.

주제

- [항목 상호 작용 이벤트 기록 및 모델 교육을 위한 요구 사항](#)
- [항목 상호 작용 이벤트 트래커 생성](#)
- [PutEvents 오퍼레이션 사용](#)
- [이벤트 지표 및 어트리뷰션 보고서](#)

항목 상호 작용 이벤트 기록 및 모델 교육을 위한 요구 사항

항목 상호 작용 이벤트를 기록하려면 다음이 필요합니다.

- Item interactions 데이터세트를 포함하는 데이터세트 그룹(비워 둘 수 있음) [시작하기](#) 안내서를 살펴보았으면 자신이 만든 동일한 데이터세트 그룹 및 데이터세트를 사용할 수 있습니다. 데이터세트 그룹 및 데이터세트 생성에 대한 자세한 내용은 [단계 2: 데이터 준비 및 가져오기](#) 단원을 참조하세요.
- 이벤트 트래커.
- [PutEvents](#) API 작업에 대한 호출.
- AWS Lambda 함수를 사용하여 PutEvents 작업을 호출하는 경우 함수의 실행 역할에 요소에 와 일드카드가 * 포함된 `personalize:PutEvents` 작업을 수행할 수 있는 권한이 있어야 합니다.

Resource

빈 항목 상호 작용 데이터 세트로 시작해서 데이터를 충분히 기록했으면 새로 기록된 이벤트만 사용하여 모델을 교육할 수 있습니다. 모든 사용 사례(도메인 데이터세트 그룹) 및 레시피(사용자 지정 데이터세트 그룹)에 대해 학습 전에 상호작용 데이터에는 다음이 포함되어야 합니다.

- 카탈로그의 항목과 상호 작용하는 사용자의 1,000개 이상의 항목 상호 작용 기록. 이러한 상호작용은 대량 가져오기, 스트리밍된 이벤트 또는 둘 다에서 발생할 수 있습니다.
- 각각 2번 이상의 항목 상호 작용을 갖는 25개 이상의 고유 사용자 ID.

품질 추천의 경우, 각 2번 이상의 항목 상호 작용을 갖는 1,000명 이상의 사용자로부터 50,000건 이상의 항목 상호 작용을 갖는 것이 좋습니다.

항목 상호 작용 이벤트 트래커 생성

항목 상호 작용 이벤트를 기록하려면 먼저 항목 상호 작용 이벤트 트래커를 생성해야 합니다. 이벤트 트래커는 새 이벤트 데이터를 데이터 세트 그룹의 항목 상호 작용 데이터 세트에 전달합니다.

Personalize 콘솔 또는 [CreateEventTracker](#) API 작업을 사용하여 이벤트 트래커를 생성합니다. 대상 항목 상호 작용 데이터 세트가 포함된 데이터 세트 그룹의 Amazon 리소스 이름(ARN)을 파라미터로 전달합니다. Personalize 콘솔을 사용하여 이벤트 트래커를 생성하는 방법에 대한 지침은 [이벤트 트래커 생성\(콘솔\)](#) 단원을 참조하세요.

이벤트 트래커에는 [PutEvents](#) 작업 사용 시 파라미터로 전달하는 추적 ID가 포함되어 있습니다. 그러면 Amazon Personalize는 이벤트 트래커에서 지정한 데이터 세트 그룹의 항목 상호 작용 데이터 세트에 새 이벤트 데이터를 추가합니다.

Note

데이터 세트 그룹에 대해 항목 상호 작용 이벤트 트래커를 하나만 생성할 수 있습니다.

Python

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_event_tracker(
    name='MovieClickTracker',
    datasetGroupArn='arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieClickGroup'
)
print(response['eventTrackerArn'])
print(response['trackingId'])
```

이벤트 트래커 ARN과 추적 ID가 표시됩니다. 예:

```
{
```

```

    "eventTrackerArn": "arn:aws:personalize:us-west-2:acct-id:event-tracker/
MovieClickTracker",
    "trackingId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx"
}

```

AWS CLI

```

aws personalize create-event-tracker \
  --name MovieClickTracker \
  --dataset-group-arn arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieClickGroup

```

이벤트 트래커 ARN과 추적 ID가 표시됩니다. 예:

```

{
  "eventTrackerArn": "arn:aws:personalize:us-west-2:acct-id:event-tracker/
MovieClickTracker",
  "trackingId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx"
}

```

SDK for Java 2.x

```

public static String createEventTracker(PersonalizeClient personalizeClient,
                                       String eventTrackerName,
                                       String datasetGroupArn) {

    String eventTrackerId = null;
    String eventTrackerArn = null;
    long maxTime = 3 * 60 * 60;
    long waitInMilliseconds = 30 * 1000;
    String status;

    try {
        CreateEventTrackerRequest createEventTrackerRequest =
        CreateEventTrackerRequest.builder()
            .name(eventTrackerName)
            .datasetGroupArn(datasetGroupArn)
            .build();

        CreateEventTrackerResponse createEventTrackerResponse =
            personalizeClient.createEventTracker(createEventTrackerRequest);

        eventTrackerArn = createEventTrackerResponse.eventTrackerArn();
    }
}

```

```

    eventTrackerId = createEventTrackerResponse.trackingId();

    System.out.println("Event tracker ARN: " + eventTrackerArn);
    System.out.println("Event tracker ID: " + eventTrackerId);

    maxTime = Instant.now().getEpochSecond() + maxTime;

    DescribeEventTrackerRequest describeRequest =
DescribeEventTrackerRequest.builder()
    .eventTrackerArn(eventTrackerArn)
    .build();

    while (Instant.now().getEpochSecond() < maxTime) {

        status =
personalizeClient.describeEventTracker(describeRequest).eventTracker().status();
        System.out.println("EventTracker status: " + status);

        if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
            break;
        }
        try {
            Thread.sleep(waitInMilliseconds);
        } catch (InterruptedException e) {
            System.out.println(e.getMessage());
        }
    }
    return eventTrackerId;
}
catch (PersonalizeException e){
    System.out.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
return eventTrackerId;
}
}

```

PutEvents 오퍼레이션 사용

데이터 세트 그룹의 항목 상호 작용 데이터 세트 및 [이벤트 트래커](#)를 생성하면 항목 상호 작용 이벤트를 기록할 준비가 된 것입니다. 항목 상호 작용 이벤트를 기록하려면 [PutEvents](#) API 작업을 사용합니다. 다음 섹션에서는 단일 이벤트를 기록하는 방법, 이벤트 값 데이터로 여러 이벤트를 기록하는 방법, 이벤트에 노출 데이터를 포함하는 방법을 보여줍니다.

익명 사용자를 위한 이벤트를 기록하는 자세한 방법은 [익명 사용자를 위한 이벤트 기록](#) 섹션을 참조하세요.

주제

- [단일 항목 상호 작용 이벤트 기록](#)
- [이벤트 값 데이터로 여러 항목 상호 작용 이벤트 기록](#)
- [노출 데이터 기록](#)

단일 항목 상호 작용 이벤트 기록

다음 예제는 하나의 항목 상호 작용 이벤트를 전달하는 PutEvents 작업을 보여줍니다. 항목 상호 작용 데이터 세트의 예제 행과 함께 해당하는 스키마가 표시됩니다.

애플리케이션은 사용자가 웹 사이트를 처음 방문하거나 애플리케이션을 사용할 때 고유한 sessionId을 생성합니다. 세션 전체의 모든 이벤트에서 동일한 sessionId을 사용해야 합니다. Personalize는 사용자가 로그인하기 전에 sessionId를 사용하여 이벤트를 사용자와 연결합니다(익명 처리). 자세한 내용은 [익명 사용자를 위한 이벤트 기록](#) 단원을 참조하세요.

이벤트 목록은 [Event](#) 객체의 어레이입니다. 각 이벤트에는 eventType이 필요하지만 이 예시에서는 eventType데이터가 스키마에 포함되어 있지 않기 때문에 학습에 사용되지 않습니다. 요구 사항을 충족하기 위해 자리 표시자 값을 제공할 수 있습니다.

trackingId은 [항목 상호 작용 이벤트 트래커 생성](#)에서 생성한 이벤트 트래커에서 가져온 것입니다. userId, itemId 및 sentAt 파라미터는 해당 과거 Interactions 데이터 세트의 USER_ID, ITEM_ID 및 TIMESTAMP 필드에 매핑됩니다. 자세한 정보는 [스키마](#)를 참조하세요.

해당 데이터 세트 열

```
Dataset columns: USER_ID, ITEM_ID, TIMESTAMP
Item interactions dataset data: user123, item-xyz, 1543631760
```

코드 예제

Python

```
import boto3

personalize_events = boto3.client(service_name='personalize-events')

personalize_events.put_events(
```



```

trackingId = 'tracking_id',
userId= 'USER_ID',
sessionId = 'session_id',
eventList = [{
    'sentAt': TIMESTAMP,
    'eventType': 'eventTypePlaceholder',
    'itemId': 'ITEM_ID'
}]
)

```

AWS CLI

```

aws personalize-events put-events \
  --tracking-id tracking_id \
  --user-id USER_ID \
  --session-id session_id \
  --event-list '[{
    "sentAt": TIMESTAMP,
    "eventType": "eventTypePlaceholder",
    "itemId": "ITEM_ID"
}]'

```

SDK for Java 2.x

```

public static void putEvents(PersonalizeEventsClient personalizeEventsClient,
                             String trackingId,
                             String sessionId,
                             String userId,
                             String itemId) {

    try {
        Event event = Event.builder()
            .sentAt(Instant.ofEpochMilli(System.currentTimeMillis() + 10 * 60 *
1000))
            .itemId(itemId)
            .eventType("typePlaceholder")
            .build();

        PutEventsRequest putEventsRequest = PutEventsRequest.builder()
            .trackingId(trackingId)
            .userId(userId)
            .sessionId(sessionId)
            .eventList(event)

```

```

        .build();

    int responseCode = personalizeEventsClient.putEvents(putEventsRequest)
        .sdkHttpResponse()
        .statusCode();
    System.out.println("Response code: " + responseCode);

} catch (PersonalizeEventsException e) {
    System.out.println(e.awsErrorDetails().errorMessage());
}
}

```

이 예제 이후에는 필수 속성만 사용하여 모델 학습으로 진행합니다.

이벤트 값 데이터로 여러 항목 상호 작용 이벤트 기록

다음 예제는 이벤트 유형과 이벤트 값이 서로 다른 여러 항목 상호 작용 이벤트를 기록하는 방법을 보여줍니다.

솔루션을 구성할 때 항목 상호 작용 데이터 세트에 EVENT_TYPE 및 EVENT_VALUE 필드가 포함되어 있는 경우, 특정 값을 임계값으로 설정하여 교육에서 레코드를 제외할 수 있습니다. 자세한 내용은 [교육에 사용할 항목 상호 작용 데이터 선택](#) 섹션을 참조하세요.

특정 레시피에 의해 메타데이터로 사용되는 추가 속성 numRatings의 기록도 보여줍니다.

```

Dataset columns: USER_ID, ITEM_ID, TIMESTAMP, EVENT_TYPE, EVENT_VALUE, NUM_RATINGS
Item interactions dataset: user123, movie_xyz, 1543531139, rating, 5, 12
                        user321, choc-ghana, 1543531760, like, 4
                        user111, choc-fake, 1543557118, like, 3

```

Python

```

import boto3
import json

personalize_events = boto3.client(service_name='personalize-events')

personalize_events.put_events(
    trackingId = 'tracking_id',
    userId= 'user555',
    sessionId = 'session1',
    eventList = [{

```

```

    'eventId': 'event1',
    'sentAt': 1553631760,
    'eventType': 'like',
    'properties': json.dumps({
        'itemId': 'choc-panama',
        'eventValue': 4,
        'numRatings': 0
    })
  }, {
    'eventId': 'event2',
    'sentAt': 1553631782,
    'eventType': 'rating',
    'properties': json.dumps({
        'itemId': 'movie_ten',
        'eventValue': 3,
        'numRatings': 13
    })
  ]
}
)

```

AWS CLI

```

aws personalize-events put-events \
  --tracking-id tracking_id \
  --user-id user555 \
  --session-id session1 \
  --event-list '[{
    "eventId": "event1",
    "sentAt": 1553631760,
    "eventType": "like",
    "properties": "{\"itemId\": \"choc-panama\", \"eventValue\": \"true\"}"
  }, {
    "eventId": "event2",
    "sentAt": 1553631782,
    "eventType": "rating",
    "properties": "{\"itemId\": \"movie_ten\", \"eventValue\": \"4\",
    \"numRatings\": \"13\"}"
  }]'

```

SDK for Java 2.x

```

public static void putMultipleEvents(PersonalizeEventsClient
  personalizeEventsClient,

```

```
        String trackingId,
        String sessionId,
        String userId,
        String event1Type,
        Float event1Value,
        String event1ItemId,
        int event1NumRatings,
        String event2Type,
        Float event2Value,
        String event2ItemId,
        int event2NumRatings) {

    ArrayList<Event> eventList = new ArrayList<Event>();

    try {
        Event event1 = Event.builder()
            .eventType(event1Type)
            .sentAt(Instant.ofEpochMilli(System.currentTimeMillis() + 10 * 60 *
1000))
            .itemId(event1ItemId)
            .eventValue(event1Value)
            .properties("{\"numRatings\": " + event1NumRatings + "\"}")
            .build();

        eventList.add(event1);

        Event event2 = Event.builder()
            .eventType(event2Type)
            .sentAt(Instant.ofEpochMilli(System.currentTimeMillis() + 10 * 60 *
1000))
            .itemId(event2ItemId)
            .eventValue(event2Value)
            .properties("{\"numRatings\": " + event2NumRatings + "\"}")
            .build();

        eventList.add(event2);

        PutEventsRequest putEventsRequest = PutEventsRequest.builder()
            .trackingId(trackingId)
            .userId(userId)
            .sessionId(sessionId)
            .eventList(eventList)
            .build();
```

```

        int responseCode = personalizeEventsClient.putEvents(putEventsRequest)
            .sdkHttpResponse()
            .statusCode();

        System.out.println("Response code: " + responseCode);

    } catch (PersonalizeEventsException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
}

```

Note

속성 키는 상호작용 스키마의 필드와 일치하는 낙타 대문자 이름을 사용합니다. 예를 들어 상호작용 스키마에서 'NUM_RATINGS' 필드를 정의한 경우 속성 키는 numRatings이 되어야 합니다.

노출 데이터 기록

[사용자-개인 맞춤](#) 레시피를 사용하거나 도메인 데이터세트 그룹의 데이터세트에 대한 스키마에 IMPRESSIONS 필드를 추가하면 작업에서 노출 데이터를 기록할 수 있습니다. PutEvents 노출은 사용자가 특정 항목과 상호작용(예: 클릭 또는 시청)했을 때 볼 수 있었던 항목의 목록입니다. Personalize 는 노출 데이터를 사용하여 탐색을 안내합니다. 이때 상호작용 데이터 또는 관련성이 낮은 항목이 추천에 포함됩니다. Personalize가 모델링할 수 있는 암시적 및 명시적 노출에 대한 자세한 내용은 [노출 데이터](#) 단원을 참조하세요.

Important

PutEvents 요청에서 암시적 및 명시적 노출 데이터가 충돌하는 경우 Personalize는 기본적으로 명시적 노출을 사용합니다.

Personalize 추천을 기록하려면 사용자에게 노출 데이터로 표시하고 [PutEvents](#) 요청에 recommendationId를 포함시킵니다. 그러면 Personalize가 추천 데이터를 기반으로 암시적 노출을 도출합니다.

이벤트의 노출 데이터를 수동으로 기록하려면 [PutEvents](#) 명령의 impression 입력 파라미터에 노출을 나열합니다. 다음 코드 샘플은 Python용 SDK (Boto3) 또는 Java 2.x용 SDK를 사용하여 PutEvents 연

산에 recommendationId a와 impression an을 포함하는 방법을 보여줍니다. 둘 다 포함시킬 경우 Personalize는 기본적으로 명시적 노출을 사용합니다.

SDK for Python (Boto3)

```
import boto3

personalize_events = boto3.client(service_name='personalize-events')

personalize_events.put_events(
    trackingId = 'tracking_id',
    userId= 'userId',
    sessionId = 'sessionId',
    eventList = [{
        'eventId': 'event1',
        'eventType': 'rating',
        'sentAt': 1553631760,
        'itemId': 'item id',
        'recommendationId': 'recommendation id',
        'impression': ['itemId1', 'itemId2', 'itemId3']
    }]
)
```

SDK for Java 2.x

노출 데이터 및 추천 ID를 사용하여 이벤트를 기록하려면 다음 putEvents메서드를 사용합니다. 노출 매개 변수의 경우 ItemID 목록을 로 전달하십시오. ArrayList

```
public static void putEvents(PersonalizeEventsClient personalizeEventsClient,
                            String trackingId,
                            String sessionId,
                            String userId,
                            String eventType,
                            Float eventValue,
                            String itemId,
                            ArrayList<String> impressions,
                            String recommendationId) {

    try {
        Event event = Event.builder()
            .eventType(eventType)
            .sentAt(Instant.ofEpochMilli(System.currentTimeMillis() + 10 * 60 *
1000))
```

```

        .itemId(itemId)
        .eventValue(eventValue)
        .impression(impressions)
        .recommendationId(recommendationId)
        .build();

    PutEventsRequest putEventsRequest = PutEventsRequest.builder()
        .trackingId(trackingId)
        .userId(userId)
        .sessionId(sessionId)
        .eventList(event)
        .build();

    int responseCode = personalizeEventsClient.putEvents(putEventsRequest)
        .sdkHttpResponse()
        .statusCode();
    System.out.println("Response code: " + responseCode);

} catch (PersonalizeEventsException e) {
    System.out.println(e.awsErrorDetails().errorMessage());
}
}

```

이벤트 지표 및 어트리뷰션 보고서

Amazon Personalize로 전송되는 이벤트의 유형과 수를 모니터링하려면 Amazon CloudWatch 지표를 사용하십시오. 자세한 정보는 [Personalize 모니터링](#)을 참조하세요.

추천의 영향을 보여주는 CloudWatch 보고서를 생성하려면 지표 어트리뷰션을 생성하고 실시간 추천으로 사용자 상호 작용을 기록하십시오. 지표 어트리뷰션 생성에 대한 자세한 내용은 [추천의 영향 측정 단원](#)을 참조하세요.

각 이벤트에 대해 사용자에게 보여준 추천의 추천 ID를 포함시킵니다. 또는 이벤트 소스(예: 타사)를 포함시킵니다. 이 데이터를 가져와서 다른 캠페인, 추천자, 타사와 비교해 봅니다. 최대 100개의 이벤트 어트리뷰션 소스를 가져올 수 있습니다.

- recommendationId를 입력하면 Personalize는 자동으로 소스 캠페인 또는 추천자를 결정하고 EVENT_ATTRIBUTION_SOURCE 열의 보고서에서 이를 식별합니다.
- 두 속성을 모두 제공하는 경우 Personalize는 eventAttributionSource만 사용합니다.

- 소스를 제공하지 않는 경우 Personalize는 보고서에서 소스 SOURCE_NAME_UNDEFINED에 레이블을 지정합니다.

다음 코드는 PutEvents 작업 중에 eventAttributionSource for 이벤트를 제공하는 방법을 보여줍니다.

```
response = personalize_events.put_events(  
    trackingId = 'eventTrackerId',  
    userId= 'userId',  
    sessionId = 'sessionId123',  
    eventList = [{  
        'eventId': 'event1',  
        'eventType': 'watch',  
        'sentAt': '1667260945',  
        'itemId': '123',  
        'metricAttribution': {  
            'eventAttributionSource': 'thirdPartyServiceXYZ'  
        }  
    }]  
)  
statusCode = response['ResponseMetadata']['HTTPStatusCode']  
print(statusCode)
```

다음 코드는 PutEvents 작업에서 recommendationId for 이벤트를 제공하는 방법을 보여줍니다.

```
response = personalize_events.put_events(  
    trackingId = 'eventTrackerId',  
    userId= 'userId',  
    sessionId = 'sessionId123',  
    eventList = [{  
        'eventId': 'event1',  
        'eventType': 'watch',  
        'sentAt': '1667260945',  
        'itemId': '123',  
        'recommendationId': 'RID-12345678-1234-1234-1234-abcdefghijkl'  
    }]  
)  
statusCode = response['ResponseMetadata']['HTTPStatusCode']  
print(statusCode)
```


작업 상호 작용 이벤트 기록

작업 상호 작용 이벤트는 사용자와 작업 간의 상호 작용입니다. 멤버십 프로그램에 등록하거나 신용카드를 신청하는 사용자를 예로 들 수 있습니다.

PERSONALIZED_ACTIONS 사용자 지정 레시피를 사용하는 경우 고객이 작업 추천에 반응할 때 실시간 작업 상호 작용 이벤트를 기록하세요. 이렇게 하면 상호 작용 데이터가 쌓이고 데이터를 최신으로 유지할 수 있습니다. 또한 Amazon Personalize에 사용자의 현재 관심사를 알려주므로, 추천 관련성을 높일 수 있습니다. PERSONALIZED_ACTIONS 사용자 지정 레시피만 작업 상호 작용 데이터를 사용합니다.

[PutActionInteractions](#) API 작업을 사용하여 작업 상호 작용 이벤트를 기록합니다. Amazon Personalize는 이 데이터를 데이터 세트 그룹의 [작업 상호 작용 데이터 세트](#)에 추가합니다.

작업 상호 작용 이벤트에는 이벤트 유형 속성이 있어야 하며, 이벤트 유형 속성은 다음 중 하나일 수 있습니다.

- Taken – 사용자가 추천 작업을 실행할 때 Taken 이벤트가 기록됩니다.
- Not Taken – 사용자가 작업을 본 후 작업을 실행하지 않기로 의도적으로 선택한 경우 Not Taken 이벤트가 기록됩니다. 작업을 보여줄 때 아니요를 선택하는 경우를 예로 들 수 있습니다. Not Taken 이벤트는 고객이 해당 작업에 관심이 없음을 의미할 수 있습니다.
- Viewed – 작업을 수행할지 여부를 결정하기 전에 사용자에게 작업을 보여주면 Viewed 이벤트가 기록됩니다. Amazon Personalize는 View 이벤트를 통해 사용자의 관심도를 파악합니다. 예를 들어, 사용자가 작업을 보고 실행하지 않는 경우 이 사용자는 앞으로 해당 작업에 관심이 없을 수 있습니다.

AWS SDK 또는 AWS Command Line Interface (AWS CLI) 를 사용하여 실시간 이벤트를 기록할 수 있습니다. 타임스탬프와 속성이 정확히 동일한 두 이벤트를 기록할 경우 Personalize는 이벤트 중 하나만 보관합니다.

주제

- [작업 상호 작용 이벤트를 기록하기 위한 요구 사항](#)
- [작업 상호 작용 이벤트 트래커 ID 찾기](#)
- [오퍼레이션 사용 PutActionInteractions](#)

작업 상호 작용 이벤트를 기록하기 위한 요구 사항

실시간 작업 상호 작용 이벤트를 기록하려면 다음이 필요합니다.

- Action interactions dataset를 포함하는 데이터 세트 그룹(비워 둘 수 있음). 데이터세트 그룹 및 데이터세트 생성에 대한 자세한 내용은 [단계 2: 데이터 준비 및 가져오기](#) 단원을 참조하세요.
- 이벤트 트래커의 ID. PutActionInteractions 작업 시 이 ID를 지정합니다. 작업 상호 작용 데이터 세트를 생성하면 Amazon Personalize가 자동으로 작업 상호 작용 이벤트 트래커를 생성합니다. 자세한 정보는 [작업 상호 작용 이벤트 트래커 ID 찾기](#)을 참조하세요.
- [PutActionInteractions](#) 작업에 대한 호출.

작업 상호 작용 이벤트 트래커 ID 찾기

작업 상호 작용 데이터 세트를 생성하면 Amazon Personalize가 자동으로 작업 상호 작용 이벤트 트래커를 생성합니다. PutActionInteractions API 작업에서 트래커 ID를 지정합니다. Amazon Personalize는 이를 사용하여 새 데이터를 데이터 세트 그룹의 작업 상호 작용 데이터 세트로 전달합니다.

Amazon Personalize 콘솔에 있는 작업 상호 작용 데이터 세트의 세부 정보 페이지에서 이벤트 트래커의 ID를 찾을 수 있습니다. 그리고 DescribeDataset API 작업을 호출하여 ID를 찾을 수 있습니다. 다음 Python 코드는 Action 상호 작용 데이터세트의 추적 ID를 인쇄합니다.

```
import boto3

personalize = boto3.client(service_name='personalize')

response = personalize.describe_dataset(
    datasetArn="Action interactions dataset ARN"
)

print(response['trackingId'])
```

오퍼레이션 사용 PutActionInteractions

작업 상호 작용 데이터 세트를 만들었으면 [PutActionInteractions](#) 작업과 함께 작업 상호 작용 이벤트를 기록할 준비가 된 것입니다. 다음 섹션에서는 단일 이벤트를 기록하는 방법과 이벤트 값 데이터로 여러 이벤트를 기록하는 방법을 보여줍니다.

주제

- [단일 작업 상호 작용 이벤트 기록](#)
- [여러 작업 상호 작용 이벤트 기록](#)

단일 작업 상호 작용 이벤트 기록

다음 코드는 하나의 TAKEN 이벤트를 전달하는 PutActionInteractions 작업을 보여줍니다. Amazon Personalize의 사용자 추천을 보여주고 사용자가 신용카드 신청과 같은 작업을 행하면, 이 이벤트를 기록할 수 있습니다.

객체 actionInteractions 배열입니다. ActionInteraction trackingId는 작업 상호 작용 데이터 세트를 생성할 때 Amazon Personalize가 생성한 이벤트 트래커에서 가져옵니다. 자세한 정보는 [작업 상호 작용 이벤트 트래커 ID 찾기](#)을 참조하세요.

애플리케이션은 사용자가 웹 사이트를 처음 방문하거나 애플리케이션을 사용할 때 고유한 sessionId을 생성합니다. 세션 전체의 모든 이벤트에서 동일한 sessionId을 사용해야 합니다. Personalize는 사용자가 로그인하기 전에 sessionId를 사용하여 이벤트를 사용자와 연결합니다(익명 처리). 자세한 정보는 [익명 사용자를 위한 이벤트 기록](#)을 참조하세요.

userId, actionId, sentAt 파라미터는 작업 상호 작용 데이터 세트의 USER_ID, ACTION_ID, EVENT_TYPE, TIMESTAMP 필드에 매핑됩니다.

해당 작업 상호 작용 데이터 세트

```
USER_ID, ACTION_ID, TIMESTAMP, EVENT_TYPE
user123, action-xyz, 1543631760, TAKEN
```

코드 예제

AWS CLI

```
aws personalize-events put-action-interactions \
--tracking-id 12345678-xxxx-xxxx-xxxx-xxxxxxxxxxxx \
--action-interactions '[{
  "userId": "user123",
  "sessionId": "abcdefg",
  "timestamp": 1543631760,
  "eventType": "TAKEN",
  "actionId": "action-xyz"}]'
```

SDK for Python (Boto3)

```
import boto3
```

```

personalize_events = boto3.client(service_name='personalize-events')

response = personalize_events.put_action_interactions(
    trackingId='12345678-xxxx-xxxx-xxxx-xxxxxxxxxxxx',
    actionInteractions=[{
        'userId': 'user123',
        'sessionId': 'abcdefg',
        'timestamp': 1543631760,
        'eventType': 'Taken',
        'actionId': 'action-xyz'
    }]
)

```

여러 작업 상호 작용 이벤트 기록

다음 코드는 동일한 sessionId를 가진 같은 사용자에게 대한 여러 작업 상호 작용 이벤트를 기록하는 방법을 보여줍니다.

해당 작업 상호 작용 데이터 세트

```

USER_ID, ACTION_ID, EVENT_TYPE, TIMESTAMP
user123, action123, Taken, 1543531139
user123, action345, Not Taken, 1543531139

```

AWS CLI

```

aws personalize-events put-action-interactions \
--tracking-id 6ddfe6b7-cd83-4dd4-b09d-4c35ecbacfe1 \
--action-interactions '[{
  "userId": "user123",
  "sessionId": "abcdefg",
  "timestamp": 1543531139,
  "eventType": "Taken",
  "actionId": "action123"
},
{
  "userId": "user123",
  "sessionId": "abcdefg",
  "timestamp": 1543531139,
  "eventType": "Not Taken",
  "actionId": "action345"}]'

```

SDK for Python (Boto3)

```
import boto3

personalize_events = boto3.client(service_name='personalize-events')

response = personalize_events.put_action_interactions(
    trackingId='12345678-xxxx-xxxx-xxxx-xxxxxxxxxxxx',
    actionInteractions=[
        {
            'userId': 'user123',
            'sessionId': 'abcdefg',
            'timestamp': 1697848587,
            'eventType': 'Taken',
            'actionId': 'action123'
        },
        {
            'userId': 'user123',
            'sessionId': 'abcdefg',
            'timestamp': 1697848622,
            'eventType': 'Not Taken',
            'actionId': 'action345'
        }
    ]
)
```

익명 사용자를 위한 이벤트 기록

Important

사용자에 대해 sessionId 및 userId가 포함된 이벤트를 최소 하나 이상 기록하지 않는 경우 Personalize는 학습 시 sessionId에 대해서만 추적되는 활동을 사용하지 않습니다. 또한 학습이 완료된 후에는 더 이상 sessionId에 대해 추적된 활동을 기반으로 추천이 제공되지 않습니다.

사용자가 계정을 생성하기 전에 항목 상호 작용 또는 작업 상호 작용 이벤트를 기록할 수 있습니다. 익명 사용자의 이벤트를 기록하여 로그인 전후의 이벤트로 지속적인 이벤트 기록을 작성할 수 있습니다. 이를 통해 Personalize는 사용자에 대한 더 많은 상호 작용 데이터를 제공하므로 관련성이 더 높은 추천을 생성하는 데 도움이 될 수 있습니다.

익명 사용자(로그인한 적이 없는 사용자)에 대한 이벤트를 기록하려면 각 이벤트에 `sessionId`만 지정합니다. 애플리케이션은 사용자가 웹 사이트를 처음 방문하거나 애플리케이션을 사용할 때 고유한 `sessionId`을 생성합니다. 세션 전체의 모든 이벤트에서 동일한 `sessionId`을 사용해야 합니다. Personalize는 사용자가 로그인하기 전에 `sessionId`를 사용하여 이벤트를 해당 사용자와 연결합니다.

Personalize는 익명 사용자의 이벤트를 `userId`와 연결할 때까지 익명 사용자의 이벤트를 사용하지 않습니다. 자세한 정보는 [익명 사용자를 위한 지속적인 이벤트 이력 작성](#)을 참조하세요.

익명 사용자에게 [실시간 개인화](#)를 제공하려면 [GetRecommendations](#) 또는 `userId` [GetActionRecommendations](#) 요청에서 `sessionId`로 지정하십시오.

- `PutEvents` 작업과 `SessionID` 및 `UserID`를 사용하여 항목 상호 작용 이벤트를 기록하는 방법을 보여주는 코드 샘플은 [여기](#)를 참조하십시오. [PutEvents 오퍼레이션 사용](#)
- `PutActionInteractions` 작업과 `SessionID` 및 `UserID`를 사용하여 작업 상호 작용 이벤트를 기록하는 방법을 보여주는 코드 샘플은 [여기](#)를 참조하십시오. [오퍼레이션 사용 PutActionInteractions](#)

익명 사용자를 위한 지속적인 이벤트 이력 작성

익명 사용자를 위한 이벤트 기록을 작성하고 Personalize가 학습 시 해당 이벤트를 사용하도록 하려면 `sessionId`와 `userId`가 모두 포함된 이벤트를 최소 하나 이상 기록합니다. 그러면 `userId`에 대한 이벤트를 원하는 수만큼 기록할 수 있습니다. `userId`를 제공하기 시작한 후 `sessionId`가 변경될 수 있습니다. 다음 전체 재학습 시 Personalize는 `userId`를 `sessionId`원본에 대해 추적된 익명 사용자 기록과 연결합니다.

재학습이 완료되면 익명 이벤트의 `sessionId`와 해당 `userId`에 대해 추적된 모든 이벤트 모두에 대해 추적된 활동을 기반으로 추천이 제공됩니다.

Note

사용자가 계정을 생성하지 않은 상태에서 Personalize가 학습 시 데이터를 사용하도록 하려는 경우 이벤트에 `sessionId`를 `userId`로 사용할 수 있습니다. 단, 사용자가 결국 계정을 만들면 익명 브라우저에서 발생한 이벤트를 새 `userId`와 연결할 수 없습니다.

타사 이벤트 추적 서비스

다음 고객 데이터 플랫폼(CDP)을 사용하면 애플리케이션에서 이벤트 데이터를 수집하고 Personalize로 보내는 데 도움이 될 수 있습니다.

- Amplitude — Amplitude를 사용하면 사용자 행동을 추적하여 사용자의 행동을 이해할 수 있습니다. Amplitude 및 Personalize를 사용하는 방법에 대한 자세한 내용은 다음 AWS 파트너 네트워크(APN) 블로그 게시물: [Amplitude와 Personalize를 통한 개인 맞춤의 효율성 측정](#)을 참조하세요.
- mParticle — mParticle을 사용하면 앱에서 이벤트 데이터를 수집할 수 있습니다. mParticle과 Personalize를 사용하여 개인 맞춤형 제품 추천을 구현하는 방법을 보여주는 예제는 [기계 학습에 CDP의 성능을 활용하는 방법: 2부](#)를 참조하세요.
- 세그먼트 — 세그먼트를 사용하면 Personalize로 데이터를 보낼 수 있습니다. 세그먼트를 Personalize와 통합하는 방법에 대한 자세한 내용은 [Personalize 목적지](#) 단원을 참조하세요.

샘플 구현

Amazon Personalize를 사용하여 이벤트 추적기 및 [PutEvents](#) 작업을 사용하여 사용자의 실시간 행동에 대응하는 방법을 보여주는 샘플 Jupyter 노트북은 리포지토리의 `getting_started` 폴더에 있는 `2.view_Campaign_and_Interactions.iPynb`를 [참조하십시오](#). [amazon-personalize-samples](#) GitHub

권장 사항과 상호 작용하는 사용자의 이벤트를 스트리밍하는 방법을 보여주는 예제는 Amazon Personalize 샘플 [리포지토리의 streaming_events](#)를 참조하십시오. GitHub

Amazon Personalize 리소스와 클라이언트 애플리케이션 사이에 있는 실시간 API를 배포하기 위한 소스 코드 및 지원 파일이 포함된 전체 예제는 샘플 리포지토리의 [실시간 개인화 API를 참조하십시오](#). AWS GitHub 이 프로젝트에는 다음을 구현하는 방법이 포함되어 있습니다.

- 사용자 컨텍스트 및 사용자 이벤트 컬렉션
- 응답 캐싱
- 항목 메타데이터를 기반으로 추천 꾸미기
- A/B 테스트
- API 인증

데이터셋의 훈련 데이터 관리

데이터를 데이터세트로 가져온 후 다음 작업을 수행할 수 있습니다.

- 카탈로그가 커지면 데이터셋의 데이터를 업데이트하세요. 이를 통해 Amazon Personalize 권장 사항의 관련성을 유지하고 개선할 수 있습니다. 대량 또는 개별 데이터 가져오기 작업을 통해 더 많은 데이터를 가져올 수 있습니다. 자세한 정보는 [데이터세트로 더 많은 훈련 데이터 가져오기](#)을 참조하세요.
- 데이터셋의 훈련 데이터를 분석합니다. 데이터 인사이트와 열 및 행 통계를 통해 데이터에 대해 학습할 수 있습니다. 또한 데이터를 개선하기 위해 어떤 조치를 취할 수 있는지 학습할 수 있습니다. 이러한 조치는 모델 교육 요구 사항과 같은 Personalize 리소스 요구 사항을 충족하는 데 도움을 줄 수 있거나 추천을 개선할 수 있습니다. 자세한 정보는 [데이터셋의 데이터 품질 및 수량 분석](#)을 참조하세요.
- Amazon S3 버킷으로 데이터를 내보냅니다. Amazon Personalize가 추천을 생성하는 데 사용하는 데이터를 확인 및 검사하거나, 이전에 실시간으로 기록한 항목 상호 작용 이벤트를 보거나, 데이터에 대한 오프라인 분석을 수행하기 위해 데이터를 내보낼 수 있습니다. 자세한 정보는 [데이터셋의 교육 데이터를 Amazon S3로 내보내기](#)을 참조하세요.
- 항목 및 사용자 데이터셋의 경우 데이터셋의 스키마를 교체하여 새 데이터 열을 추가할 수 있습니다. 데이터셋을 생성한 후 데이터 구조가 변경된 경우 데이터셋의 스키마를 교체할 수 있습니다. 자세한 정보는 [데이터셋 스키마를 교체하여 새 열 추가](#)을 참조하세요.
- 데이터셋의 모든 데이터를 삭제할 수 있습니다. 또는 데이터셋 그룹에서 사용자 및 해당 데이터 (메타데이터 및 상호작용 데이터 포함) 를 삭제할 수 있습니다. 자세한 내용은 [데이터 삭제 작업을 통한 사용자 및 데이터 삭제 및 데이터셋을 삭제하여 해당 데이터를 모두 삭제합니다.](#) 단원을 참조하세요.

주제

- [새 데이터가 실시간 추천에 영향을 미치는 방식](#)
- [데이터세트로 더 많은 훈련 데이터 가져오기](#)
- [데이터셋의 데이터 품질 및 수량 분석](#)
- [데이터셋의 교육 데이터를 Amazon S3로 내보내기](#)
- [데이터셋 스키마를 교체하여 새 열 추가](#)
- [데이터 삭제 작업을 통한 사용자 및 데이터 삭제](#)
- [데이터셋을 삭제하여 해당 데이터를 모두 삭제합니다.](#)

새 데이터가 실시간 추천에 영향을 미치는 방식

추천 또는 사용자 지정 솔루션 버전을 만든 후 새 데이터가 실시간 권장 사항에 미치는 영향은 데이터 유형, 가져오기 방법, 사용하는 도메인 사용 사례 또는 사용자 지정 레시피에 따라 달라집니다. 다음 섹션에서는 다음 교육 전에 새 데이터가 실시간 권장 사항에 미치는 영향을 설명합니다.

교육은 추천자의 주간 자동 교육 또는 자동 또는 수동 솔루션 버전 생성이 될 수 있습니다. User-
Personalization을 통한 수동 교육의 경우 로 설정해야 합니다. trainingMode FULL

새 레코드가 배치 추천에 영향을 미치는 방식에 대한 자세한 내용은 [배치 추천 받기](#) 단원을 참조하세요. 새 레코드가 배치 세그먼트 작업에 영향을 미치는 방식에 대한 자세한 내용은 [사용자 세그먼트 가져오기](#) 단원을 참조하세요.

주제

- [새 상호작용](#)
- [새 항목](#)
- [신규 사용자](#)
- [새로운 작업](#)

새 상호작용

새 상호 작용은 최신 교육 이후에 가져오는 항목 또는 작업 상호 작용입니다.

실시간 및 대량 데이터 모두에서, 상호 작용에 새 항목 또는 작업이 포함되는 경우 Amazon Personalize는 교육 없이 이를 권장 사항으로 고려할 수 있습니다. 자세한 내용은 [새 항목](#) 또는 [새로운 작업](#)을 참조하세요.

실시간 이벤트

실시간 개인화를 특징으로 하는 사용 사례 및 레시피의 경우 Amazon Personalize는 동일한 사용자에게 대한 추천을 생성할 때 사용자와 기존 항목 또는 작업 (최신 교육에 있는 기록) 간의 실시간 상호 작용을 즉시 사용합니다. 실시간 개인화에 대한 자세한 내용은 [실시간 개인 맞춤](#) 단원을 참조하세요.

유사한 항목을 추천하는 것과 같이 실시간 개인화를 지원하지 않는 도메인 사용 사례 및 사용자 지정 레시피의 경우 모델은 학습 후에만 실시간 상호 작용 데이터에서 학습합니다.

대량 상호작용

대량 상호 작용의 경우 증분 및 전체 데이터세트 가져오기 작업 모두에 대해 모델은 다음 학습 후에만 대량 항목 상호 작용 또는 작업 상호 작용 데이터에서 학습합니다. 대량 데이터는 실시간 개인 맞춤을 위한 추천을 업데이트하는 데는 사용되지 않습니다.

기존 대량 데이터 업데이트에 대한 자세한 내용은 [기존 대량 데이터 업데이트](#) 단원을 참조하세요.

새 항목

새 항목은 최신 교육 이후에 가져오는 항목입니다. 항목 데이터세트의 상호작용 데이터 또는 항목 메타 데이터에서 가져올 수도 있습니다.

새 항목은 다음과 같이 추천 대상으로 고려됩니다.

- 추천 상품 및 추천 도메인 사례 또는 사용자 개인화-v2, 사용자 맞춤화 또는 차세대 베스트 액션 레시피의 경우 Amazon Personalize는 2시간마다 모델을 자동으로 업데이트합니다. Amazon Personalize는 각 업데이트 이후 탐색의 일환으로 새 항목을 추천 대상으로 간주합니다. Amazon Personalize는 새 항목을 고려할 때 해당 항목에 대한 모든 메타데이터를 고려합니다. 그러나 이 데이터는 항목에 대한 상호 작용을 기록하고 새 모델을 학습시킨 후에야 권장 사항에 더 큰 영향을 미칩니다. 업데이트에 대한 자세한 내용은 [자동 업데이트](#) 단원을 참조하세요.
- 지금 유행 사용 사례를 사용하는 경우 Personalize는 2시간마다 상호작용 데이터를 자동으로 평가하고 유행 항목을 식별합니다. 추천자가 훈련할 때까지 기다릴 필요가 없습니다. Trending-Now 레시피를 사용하는 경우 Amazon Personalize는 교육 없이 구성 가능한 간격으로 모든 새 항목을 자동으로 고려합니다. 간격 구성에 대한 자세한 내용은 [지금-유행 레시피](#) 단원을 참조하세요.
- Trending-Now 레시피를 사용하지 않거나 사용 사례 또는 레시피가 자동 업데이트를 지원하지 않는 경우 Amazon Personalize는 다음 교육 후에만 새 항목을 고려합니다.

신규 사용자

신규 사용자는 최신 교육을 받은 후 가져온 사용자입니다. 사용자 데이터세트의 상호작용 데이터 또는 항목 메타데이터에서 가져올 수도 있습니다. 신규 익명 사용자(사용자 ID가 없는 사용자)의 경우 sessionId를 사용하여 해당 사용자의 이벤트를 기록할 수 있으며 Personalize는 사용자가 로그인하기 전에 해당 사용자에게 이벤트를 연결합니다. 자세한 내용은 [익명 사용자를 위한 이벤트 기록](#) 단원을 참조하세요.

Personalize는 다음과 같이 신규 사용자를 위한 추천을 생성합니다.

- 지금 유행 도메인 사용 사례 또는 지금-유행 사용자 지정 레시피를 사용하면 신규 사용자에게 가장 유행하는 항목에 대한 추천이 즉시 제공됩니다. 인기도-집계 레시피를 사용하면 신규 사용자는 상호 작용이 가장 많은 항목에 대한 추천을 즉시 받게 됩니다.

- 사용자에게 개인 맞춤형 추천을 제공하는 레시피나 사용 사례의 경우 기존 사용자의 초기 상호작용 이력을 기반으로 신규 사용자를 위한 추천이 제공됩니다. 기존 사용자가 가장 먼저 상호 작용한 항목 또는 작업은 신규 사용자에게 추천될 가능성이 높습니다. 사용자-개인 맞춤 또는 개인 맞춤형-순위 레시피의 경우, `recency_mask`을 `true`로 설정하면 상호작용 데이터의 최신 인기 트렌드를 기반으로 한 항목도 추천에 포함됩니다.

다음과 같이 하면 신규 사용자에 대한 추천 관련성을 높일 수 있습니다.

- 상호작용 데이터 — 새 사용자의 추천 관련성을 높이는 주요 방법은 항목과의 상호작용에서 데이터를 가져오는 것입니다. 새로운 상호작용 데이터가 추천에 영향을 미치는 방식에 대한 자세한 내용은 [새 상호작용](#) 단원을 참조하세요.
- 사용자 메타데이터 - GENDER 또는 MEMBERSHIP_STATUS와 같은 사용자 메타데이터를 가져오면 추천을 개선할 수 있습니다. 메타데이터가 추천에 영향을 미치려면 도메인 추천자의 주간 자동 재학습이 완료될 때까지 기다려야 합니다. 또는 새 솔루션 버전을 수동으로 만들어야 합니다.
- 상황별 메타데이터 - 사용 사례 또는 레시피가 상황별 메타데이터를 지원하고 항목 상호 작용 데이터 세트에 상황별 데이터를 위한 메타데이터 필드가 있는 경우 추천 요청에 사용자의 컨텍스트를 제공할 수 있습니다. 여기에는 재학습이 필요하지 않습니다. 자세한 정보는 [컨텍스트 메타데이터로 추천 관련성 높이기](#)를 참조하세요.

새로운 작업

새 액션은 최신 교육 이후 가져온 액션입니다. 작업 상호 작용 데이터나 작업 데이터 세트의 작업에서 가져올 수 있습니다.

차선책-작업 레시피를 사용하면 Amazon Personalize는 2시간마다 솔루션 버전을 자동으로 업데이트합니다. 각 업데이트 이후 Amazon Personalize는 탐색의 일환으로 새 작업을 추천 대상으로 고려합니다. Amazon Personalize는 새 작업을 고려할 때 해당 작업에 대한 모든 메타데이터를 고려합니다. 그러나 이 데이터는 작업에 대한 작업 상호 작용을 기록하고 완전히 재교육한 후에만 추천에 더 큰 영향을 미칩니다. 업데이트에 대한 자세한 내용은 [자동 업데이트](#) 섹션을 참조하세요.

데이터세트로 더 많은 훈련 데이터 가져오기

카탈로그가 커지면 추가 훈련 데이터를 데이터세트로 가져오세요. 이를 통해 Amazon Personalize 권장 사항의 관련성을 유지하고 개선할 수 있습니다. 대량 또는 개별 데이터 가져오기 작업을 통해 더 많은 데이터를 가져올 수 있습니다.

기존 데이터 세트를 업데이트하여 데이터 열을 더 추가하려면 데이터 세트의 스키마를 열어 추가된 새 스키마로 바꿀 수 있습니다. 그런 다음 새 데이터 열을 가져오면 됩니다. 자세한 정보는 [데이터세트 스키마를 교체하여 새 열 추가](#)를 참조하세요.

주제

- [개별 가져오기 작업을 통한 데이터 가져오기](#)
- [기존 대량 데이터 업데이트](#)

개별 가져오기 작업을 통한 데이터 가져오기

Amazon Personalize 데이터 세트로 데이터를 가져온 후 항목 상호 작용, 작업 상호 작용, 사용자, 항목 또는 작업을 포함한 추가 개별 레코드를 가져와 업데이트할 수 있습니다. 데이터를 개별적으로 가져오면 카탈로그가 커짐에 따라 Personalize 데이터세트에 소량의 레코드를 추가할 수 있습니다.

레코드를 개별적으로 가져오는 경우 Personalize는 새 레코드를 데이터세트에 추가합니다. 개별 항목, 사용자 또는 작업을 업데이트하려면 ID는 같아도 속성이 수정된 레코드를 가져오면 됩니다. 개별 가져오기 작업당 최대 10개의 레코드를 가져올 수 있습니다.

레코드를 개별적으로 가져오는 방법에 대한 자세한 내용은 [개별 레코드 가져오기](#) 단원을 참조하세요. 실시간 이벤트 기록에 대한 자세한 내용은 [이벤트 기록](#)을 참조하십시오.

기존 대량 데이터 업데이트

이전에 데이터세트에 대한 데이터세트 가져오기 작업을 만든 경우 [다른 가져오기 작업을 만들어](#) 대량 데이터를 추가하거나 대체합니다. 기본 설정으로 데이터세트 가져오기 작업은 대량으로 가져온 데이터세트의 기존 데이터를 대체합니다. 대신 작업의 [가져오기 모드](#)를 변경하여 기존 데이터에 새 레코드를 추가할 수 있습니다.

다음은 대량 데이터 업데이트에 대한 지침 및 요구 사항입니다.

- 데이터 세트 가져오기 작업과 함께 항목 상호 작용 데이터 세트 또는 작업 상호 작용 데이터 세트에 데이터를 추가하려면, 새 항목 상호 작용 또는 작업 상호 작용 레코드가 1,000개 이상 있어야 합니다.
- 이미 추천자를 생성했거나 캠페인과 함께 사용자 지정 솔루션 버전을 배포한 경우, 새 대량 레코드가 추천에 미치는 영향은 사용하는 도메인 사용 사례 또는 레시피에 따라 달라집니다. 자세한 정보는 [새 데이터가 실시간 추천에 영향을 미치는 방식](#)을 참조하세요.
- 대량 가져오기를 완료한 후 20분 이내에 Amazon Personalize는 데이터 세트 그룹에서 생성한 모든 필터를 새로운 대량 데이터로 업데이트합니다. 이 업데이트를 통해 Personalize는 사용자에게 대한 추천을 필터링할 때 최신 데이터를 사용할 수 있습니다.

데이터세트 가져오기 작업 생성에 대한 자세한 내용은 [데이터세트 가져오기 작업으로 대량 레코드 가져오기](#).

데이터셋의 데이터 품질 및 수량 분석

항목 상호 작용, 사용자 또는 항목 데이터 세트로 데이터를 가져온 후 Amazon Personalize 콘솔을 사용하여 데이터를 분석할 수 있습니다. 데이터 인사이트와 열 및 행 통계를 통해 데이터에 대해 학습할 수 있습니다. 또한 데이터를 개선하기 위해 어떤 조치를 취할 수 있는지 학습할 수 있습니다. 이러한 조치는 모델 교육 요구 사항과 같은 Personalize 리소스 요구 사항을 충족하는 데 도움을 줄 수 있거나 추천을 개선할 수 있습니다.

Important

Amazon Personalize 콘솔을 사용하여 작업 상호 작용 또는 작업 데이터 세트의 데이터를 분석할 수 없습니다.

권장 변경을 실시한 후에는 데이터를 다시 가져와서 문제가 해결되었는지 또는 데이터세트 통계가 개선되었는지 확인할 수 있습니다. 데이터 업데이트에 대한 자세한 내용은 [데이터세트에 더 많은 훈련 데이터 가져오기](#) 섹션을 참조하세요.

인사이트가 보이지 않으면 데이터가 Personalize 데이터 기대치에 부합하는 것입니다. 도메인 데이터 세트 그룹 또는 사용자 지정 데이터세트 그룹에서 데이터를 분석할 수 있습니다.

인사이트를 생성하고 통계를 계산할 때 Personalize는 익명이 아닌 사용자의 모든 대량 및 스트리밍 데이터를 고려합니다. 익명 사용자의 이벤트는 userId와 연결할 때까지 고려되지 않습니다. 자세한 내용은 [익명 사용자를 위한 이벤트 기록](#) 단원을 참조하세요.

주제

- [데이터 분석을 위한 필수 권한](#)
- [데이터 인사이트](#)
- [데이터세트 인사이트 및 통계 보기](#)

데이터 분석을 위한 필수 권한

사용자에게 Personalize에 대한 전체 액세스 권한을 부여하는 경우, 권한을 변경할 필요가 없습니다. Amazon Personalize에서 작업을 수행하는 데 필요한 권한만 사용자에게 부여하는 경우 AWS Identity and Access Management (IAM) 정책에 다음과 같은 추가 데이터 인사이트 작업이 포함되어야 합니다.

- 개인화: CreateData InsightsJob
- 개인화: ListData InsightsJob
- 개인화: DescribeData InsightsJob
- 개인화: 인사이트 GetData

데이터 인사이트

다음은 Personalize에서 생성할 수 있는 가능한 데이터 인사이트입니다.

인사이트	작업	관련 데이터 세트
상호작용 데이터 세트에는 X개의 상호작용만 있습니다. 모델 학습에는 최소 1,000개의 상호작용이 필요합니다. 50,000개 이상을 권장합니다.	모델을 학습시키기 전에 Y개의 고유한 상호작용 레코드를 추가로 가져오세요.	항목 상호작용
상호작용 데이터 세트에는 상호작용이 두 개 이상 있는 X명의 고유 사용자만 있습니다. 모델 교육에는 최소 25명의 사용자가 필요합니다. 1,000명 이상을 권장합니다.	Y명의 추가 사용자에 대해 각각 2개 이상의 상호작용 레코드를 가져오세요.	항목 상호작용
항목 데이터 세트에 있는 X%의 항목에 상호작용 데이터 세트에 상호작용이 없으므로 권장되지 않을 수 있습니다.	모든 상호작용 데이터를 가져와서 항목과 상호작용 데이터 세트 간 ID가 일치하지 않는지 확인하세요. 아래 데이터 세트 통계에서 항목 및 상호작용 데이터 세트를 확인하여 예상되는 행의 수를 가져왔는지 확인하세요. 사용 사례 또는 레시피에서 탐색을 사용하는 경우, 상호작용 데이터가 없는 더 많은 항목을 추천하도록 탐색 구성을 수정하세요.	항목 상호작용 및 항목
사용자 데이터 세트에 있는 X%의 사용자는 상호작용 데이터 세트에 상호작용이 없습니다. 이러한 사용자는 인기 항목에 대한 추천을 수신하게 됩니다.	모든 상호작용 데이터를 가져와서 사용자와 상호작용 데이터 세트 간 ID가 일치하지 않는지 확인하세요. 아래 데이터 세트 통계에서 사용자 및 상호작용 데이터 세트를 확	항목 상호작용 및 사용자

인사이트	작업	관련 데이터 세트
	<p>인하여 예상되는 행의 수를 가져왔는지 확인하세요. 더 많은 사용자가 상호작용 데이터를 확보할 수 있도록 추가 상호작용을 가져오세요.</p>	
<p><Users or Items or Interactions> 데이터 세트에 누락된 값을 포함한 X%의 행이 있습니다. 이는 추천에 부정적인 영향을 미칠 수 있습니다. 모든 필수 및 선택 필드를 70% 이상 작성하는 것이 좋습니다.</p>	<p>전체 기록을 추가로 가져오거나, 작성되지 않은 행이 없이 데이터를 다시 가져오거나, 누락된 값을 대체 데이터(예: 숫자 열의 평균 또는 범주형 열의 가장 일반적인 값)로 대체하여 데이터를 다시 가져옵니다.</p>	<p>모두 해당</p>
<p><datasetType>데이터 세트의 다음 열은 70% 미만 완료되었습니다. <ColumnName, ColumnName... >. 이 데이터를 교육에 포함할 경우, 추천에 부정적인 영향을 미칠 수 있습니다. null 값을 허용하는 열은 70% 이상 작성하는 것이 좋습니다.</p>	<p>전체 기록을 추가로 가져오거나, 작성되지 않은 행이 없이 데이터를 다시 가져오거나, 누락된 값을 대체 데이터(예: 숫자 열의 평균 또는 범주형 열의 가장 일반적인 값)로 대체하여 데이터를 다시 가져옵니다.</p>	<p>모두 해당</p>
<p>다음 (숫자) 열에는 이상값이 있습니다. <,... >. ColumnName ColumnName 특이값이 항상 문제가 되는 것은 아니지만 추천에 부정적인 영향을 미치는 경우도 있습니다.</p>	<p>아래 열 통계를 사용하여 해당 열의 최소값과 최대값이 예상과 일치하는지 확인하세요. 이러한 값이 예상과 일치하지 않는 경우, 해당 열의 데이터가 부정확하지 않은지 확인하고 데이터 수집 및 데이터 처리에 문제가 있는지 검토하세요.</p>	<p>모두</p>
<p>다음 열에는 1000개가 넘는 범주가 있습니다. <ColumnName,... >. ColumnName 이 데이터를 학습에 포함하면 권장 사항에 부정적인 영향을 미칠 수 있습니다: <ColumnName, ColumnName... >.</p>	<p>범주형 데이터에 철자 차이로 인한 중복 카테고리 등의 문제가 없는지 확인하세요. 부정확성을 모두 해결하고 데이터를 다시 가져오세요.</p>	<p>모두</p>

인사이트	작업	관련 데이터세트
다음 텍스트 메타데이터 열은 85% 미만으로 완료되었으며 모델 학습에 사용되지 않습니다. <ColumnName, ColumnName...>.	행을 추가로 가져오거나 해당 열의 텍스트 데이터와 함께 행을 다시 가져오세요.	Items
상호작용 데이터세트에는 10개가 넘는 고유한 이벤트 유형이 있으며, 이로 인해 모델 학습이 실패할 수 있습니다.	이벤트 유형 열에 철자 차이로 인한 이벤트 유형 중복 등 부정확한 부분이 없는지 확인하세요. 불필수 이벤트 유형을 제거하고 데이터를 다시 가져오세요.	항목 상호 작용
상호작용 데이터세트에는 모든 기록에 대한 동일한 타임스탬프가 있습니다. 사용자_세분화 레시피를 사용하고 모든 기록의 타임스탬프가 동일하면, 모델 학습은 실패합니다.	데이터에 타임스탬프 문제가 있는지 확인하고, 중복된 타임스탬프를 고유한 타임스탬프로 바꾸세요.	항목 상호 작용

데이터세트 인사이트 및 통계 보기

Personalize 데이터세트의 데이터에 대한 인사이트와 통계를 보려면 Personalize 콘솔의 데이터세트로 이동한 다음 분석 실행을 선택합니다.

인사이트와 통계를 보려면

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터세트 그룹 페이지에서 데이터세트 그룹을 선택합니다.
3. 탐색 창의 데이터 세트에서 데이터 분석을 선택합니다.
4. 오른쪽 상단에서 분석 실행을 선택합니다. Personalize에서 데이터 분석을 시작합니다. 이 프로세스는 최대 15분이 걸릴 수 있습니다. 성공하면 결과가 이 페이지에 표시됩니다.
5. 인사이트에서 다음을 사용하여 나타나는 인사이트를 필터링합니다.

- 특정 언어가 포함된 인사이트를 찾으려면 인사이트 찾기 에 기준을 입력합니다. 텍스트를 입력하면 인사이트 또는 권장 조치에 정확한 문자열이 포함된 인사이트만 포함되도록 목록이 업데이트됩니다.
 - 데이터세트 유형별로 통찰력을 필터링하려면 모든 데이터세트를 특정 데이터세트 유형으로 변경하세요. 이 데이터세트와 관련된 인사이트만 포함하도록 목록이 업데이트됩니다.
6. 데이터세트에 대한 데이터세트 통계를 보려면 다음과 같이 하세요.
- 상호작용 데이터세트의 행 수, 고유 사용자, 고유 항목 등 데이터세트에 대한 일반 세부 정보 및 통계를 보려면, 데이터세트 섹션을 펼칩니다.
 - 열에 대한 자세한 통계를 보려면 데이터세트 섹션을 펼치고 열 수준 통계를 선택한 다음 해당 열의 라디오 버튼을 선택합니다.
7. 데이터의 모든 문제를 수정하고 다시 가져온 다음, 다른 분석을 실행하여 확인합니다. 다시 데이터 가져오기에 대한 자세한 내용은 [데이터세트로 더 많은 훈련 데이터 가져오기](#) 섹션을 참조하세요.

데이터세트의 교육 데이터를 Amazon S3로 내보내기

Personalize 데이터세트로 데이터를 가져온 후 데이터를 S3 버킷으로 내보낼 수 있습니다. Amazon Personalize가 추천을 생성하는 데 사용하는 데이터를 확인 및 검사하거나, 이전에 실시간으로 기록한 항목 상호 작용 이벤트를 보거나, 데이터에 대한 오프라인 분석을 수행하기 위해 데이터를 내보낼 수 있습니다.

대량으로 가져온 데이터(Personalize 데이터세트 가져오기 작업을 사용하여 가져온 데이터)만 내보내거나, 개별적으로 가져온 데이터(콘솔이나, PutEvents, PutUsers 또는 PutItems 작업을 사용하여 가져온 레코드)만 내보내거나, 둘 다 내보내도록 선택할 수 있습니다.

Note

작업 상호 작용 데이터 세트 또는 작업 데이터 세트의 데이터는 내보낼 수 없습니다.

모든 필드와 정확히 일치하는 레코드의 경우 Personalize는 레코드 하나만 내보냅니다. 두 레코드의 ID가 동일하지만 하나 이상의 필드가 다른 경우 Personalize는 내보내기로 선택한 데이터에 따라 레코드를 포함시키거나 제거합니다.

- 대량 및 증분 데이터를 모두 내보내는 경우 Personalize는 동일한 ID를 가진 최신 항목만 내보내고 (항목 데이터세트 내보내기에서), 동일한 ID를 가진 사용자만 내보냅니다(사용자 데이터세트 내보내

기에서). 항목 상호 작용 데이터 세트의 경우 Amazon Personalize는 모든 항목 상호 작용 데이터를 내보냅니다.

- **중분 데이터만 내보내는 경우** Amazon Personalize는 ID가 동일한 항목 또는 사용자를 포함하여 개별적으로 가져온 모든 항목, 사용자 또는 항목 상호 작용 데이터를 내보냅니다. 모든 필드와 정확히 일치하는 레코드만 제외됩니다.
- **대량 데이터만 내보내는 경우** Amazon Personalize는 동일한 ID를 가진 항목 또는 사용자를 포함하여 대량으로 가져온 모든 항목, 사용자 또는 항목 상호 작용 데이터를 포함시킵니다. 모든 필드와 정확히 일치하는 레코드만 제외됩니다.

데이터세트를 내보내려면 데이터세트 내보내기 작업을 생성합니다. 데이터세트 내보내기 작업은 데이터세트의 레코드를 S3 버킷에 있는 하나 이상의 CSV 파일로 출력하는 레코드 내보내기 도구입니다. 출력 CSV 파일에는 데이터세트 스키마의 필드와 일치하는 열 이름을 가진 헤더 행이 포함됩니다.

주제

- [데이터세트 내보내기 작업 권한 요구 사항](#)
- [데이터세트 내보내기 작업 생성](#)

데이터세트 내보내기 작업 권한 요구 사항

데이터세트를 내보내려면 S3 버킷에 파일을 추가할 수 있는 권한이 Personalize에 필요합니다. 권한을 부여하려면 버킷에서 PutObject 및 ListBucket 작업을 사용할 수 있는 권한을 역할에 부여하는 새 AWS Identity and Access Management (IAM) 정책을 Personalize 서비스 역할에 연결하고, 출력 S3 버킷에 PutObject 및 ListBucket 작업을 사용할 수 있는 Personalize 원칙 권한을 부여하는 버킷 정책을 연결합니다.

암호화에 AWS Key Management Service (AWS KMS) 를 사용하는 경우 Amazon Personalize와 Amazon Personalize IAM 서비스 역할에 키를 사용할 권한을 부여해야 합니다. 자세한 정보는 [Amazon Personalize에 AWS KMS 키 사용 권한 부여](#)를 참조하세요.

데이터세트 내보내기를 위한 서비스 역할 정책

다음 예제 정책은 Personalize 서비스 역할에 PutObject 및 ListBucket 작업을 사용할 수 있는 권한을 부여합니다. bucket-name을 출력 버킷의 이름으로 바꿉니다. IAM 서비스 역할에 정책을 연결하는 방법에 대한 자세한 내용은 [S3 정책을 Personalize 서비스 역할에 연결](#) 단원을 참조하세요.

```
{
  "Version": "2012-10-17",
```

```

    "Id": "PersonalizeS3BucketAccessPolicy",
    "Statement": [
      {
        "Sid": "PersonalizeS3BucketAccessPolicy",
        "Effect": "Allow",
        "Action": [
          "s3:PutObject",
          "s3:ListBucket"
        ],
        "Resource": [
          "arn:aws:s3:::bucket-name",
          "arn:aws:s3:::bucket-name/*"
        ]
      }
    ]
  }
}

```

데이터세트를 내보내는 데 사용되는 S3 버킷 정책

다음 예제 정책은 S3 버킷에서 PutObject 및 ListBucket 작업을 사용할 수 있는 권한을 Personalize에 부여합니다. *bucket-name*을 사용 중인 버킷의 이름으로 바꿉니다. Amazon S3 버킷 정책을 버킷에 [추가하는 방법에 대한 자세한 내용은 Amazon 심플 스토리지 서비스 사용 설명서의 Amazon S3 콘솔을 사용하여 버킷 정책 추가를 참조하십시오.](#)

```

{
  "Version": "2012-10-17",
  "Id": "PersonalizeS3BucketAccessPolicy",
  "Statement": [
    {
      "Sid": "PersonalizeS3BucketAccessPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "personalize.amazonaws.com"
      },
      "Action": [
        "s3:PutObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}

```

```
]
}
```

데이터세트 내보내기 작업 생성

Amazon Personalize 콘솔 AWS Command Line Interface ,AWS CLI() 또는 SDK를 사용하여 데이터세트 내보내기 작업을 생성할 수 있습니다. AWS

데이터세트 내보내기 작업 생성(콘솔)

데이터를 데이터세트로 가져오고 출력 S3 버킷을 생성한 후, 분석을 위해 데이터를 버킷으로 내보낼 수 있습니다. Personalize 콘솔을 사용하여 데이터세트를 내보내려면 데이터세트 내보내기 작업을 생성합니다. S3 버킷 생성에 대한 자세한 내용은 Simple Storage Service 사용 설명서의 [버킷 생성](#) 단원을 참조하세요.

데이터세트를 내보내기 전에 Personalize 서비스 역할이 출력 S3 버킷에 액세스하고 여기에 기록할 수 있는지 확인합니다. [데이터세트 내보내기 작업 권한 요구 사항](#) 단원을 참조하세요.

데이터세트 내보내기 작업을 생성하려면(콘솔)

1. <https://console.aws.amazon.com/personalize/home> 에서 Personalize 콘솔을 엽니다.
2. 탐색 창에서 데이터세트 그룹을 선택합니다.
3. 데이터세트 그룹 페이지에서 데이터세트 그룹을 선택합니다.
4. 탐색 창에서 데이터세트를 선택합니다.
5. S3 버킷으로 내보내려는 데이터세트를 선택합니다.
6. 데이터세트 내보내기 작업에서 데이터세트 내보내기 작업 생성을 선택합니다.
7. 데이터세트 내보내기 작업 세부 정보에서 데이터세트 내보내기 작업 이름에 내보내기 작업의 이름을 입력합니다.
8. IAM 서비스 역할의 경우 [Personalize에 대한 IAM 역할 생성](#)에서 생성한 Personalize 서비스 역할을 선택합니다.
9. S3 데이터 출력 경로에서 대상 S3 버킷을 입력합니다. 다음 구문을 사용합니다.

s3://<name of your S3 bucket>/<folder path>

10. AWS KMS 암호화에 사용하는 경우 KMS 키 ARN에 키의 Amazon 리소스 이름 (ARN) 을 입력합니다. AWS KMS
11. 내보내기 데이터 유형에서 해당 데이터를 처음 가져온 방식에 따라 내보낼 데이터 유형을 선택합니다.

- 데이터세트 가져오기 작업을 사용하여 대량으로 가져온 데이터만 내보내려면 대량을 선택합니다.
 - 콘솔이나 PutEvents, PutUsers 또는 PutItems 작업을 사용하여 개별적으로 가져온 데이터만 내보내려면 증분을 선택합니다.
 - 데이터세트의 모든 데이터를 내보내려면 둘 다를 선택합니다.
12. 태그의 경우, 원하는 태그를 추가할 수 있습니다. Personalize 리소스 태그 지정에 대한 자세한 내용은 [Personalize 리소스에 태그 지정](#) 단원을 참조하세요.
13. 데이터세트 내보내기 작업 생성을 선택합니다.

데이터세트 개요 페이지의 데이터세트 내보내기 작업에서 작업이 내보내기 작업 상태와 함께 나열됩니다. 상태가 ACTIVE로 표시되면 해당 데이터세트 내보내기 작업은 완료된 것입니다. 그러면 출력 S3 버킷에서 데이터를 다운로드할 수 있습니다. S3 버킷에서 객체를 다운로드하는 방법에 대한 자세한 내용은 Simple Storage Service 사용 설명서의 [객체 다운로드](#) 단원을 참조하세요.

데이터세트 내보내기 작업 생성(AWS CLI)

데이터를 데이터세트로 가져오고 출력 S3 버킷을 생성한 후, 데이터세트를 분석을 위해 버킷으로 내보낼 수 있습니다. 를 사용하여 데이터세트를 내보내려면 명령어를 사용하여 데이터세트 내보내기 작업을 생성하십시오. AWS CLI create-dataset-export-job AWS CLI S3 버킷 생성에 대한 자세한 내용은 Simple Storage Service 사용 설명서의 [버킷 생성](#) 단원을 참조하세요.

데이터세트를 내보내기 전에 Personalize 서비스 역할이 출력 S3 버킷에 액세스하고 여기에 기록할 수 있는지 확인합니다. [데이터세트 내보내기 작업 권한 요구 사항](#) 섹션을 참조하십시오.

다음은 create-dataset-export-job AWS CLI 명령의 예입니다. 작업에 이름을 부여하고, dataset arn을 내보내려는 데이터세트의 리소스 이름(ARN)으로 바꾸고, role ARN를 [Personalize에 대한 IAM 역할 생성](#)에서 생성했던 Personalize 서비스 역할의 ARN으로 바꿉니다. s3DataDestinationkmsKeyArn에서는 선택적으로 AWS KMS 키에 대한 ARN을 제공하고 출력의 경우 Amazon S3 버킷의 경로를 path 제공하십시오.

ingestion-mode의 경우 다음 옵션 중에서 내보낼 데이터를 지정합니다.

- 데이터세트 가져오기 작업을 사용하여 대량으로 가져온 데이터만 내보내려면 BULK를 지정합니다.
- 콘솔 또는 PutEvents PutUsers, 또는 PutItems 작업을 사용하여 개별적으로 가져온 데이터만 PUT 내보내도록 지정합니다.
- 데이터세트의 모든 데이터를 내보내려면 ALL를 지정합니다.

자세한 내용은 [CreateDatasetExportJob](#) 단원을 참조하세요.

```
aws personalize create-dataset-export-job \
  --job-name job name \
  --dataset-arn dataset ARN \
  --job-output "{\"s3DataDestination\":{\"kmsKeyArn\":\"kms key ARN\",\"path\":
  \"s3://bucket-name/folder-name/\"}}\" \
  --role-arn role ARN \
  --ingestion-mode PUT
```

데이터세트 내보내기 작업 ARN이 표시됩니다.

```
{
  "datasetExportJobArn": "arn:aws:personalize:us-west-2:acct-id:dataset-export-job/
  DatasetExportJobName"
}
```

DescribeDatasetExportJob 작업을 사용하여 상태를 확인합니다.

```
aws personalize describe-dataset-export-job \
  --dataset-export-job-arn dataset export job ARN
```

데이터세트 내보내기 작업 (AWS SDK) 생성

데이터를 데이터세트로 가져오고 출력 S3 버킷을 생성한 후, 데이터세트를 분석을 위해 버킷으로 내보낼 수 있습니다. AWS SDK를 사용하여 데이터세트를 내보내려면 작업을 사용하여 데이터세트 내보내기 작업을 만드세요. [CreateDatasetExportJob](#) S3 버킷 생성에 대한 자세한 내용은 Simple Storage Service 사용 설명서의 [버킷 생성](#) 단원을 참조하세요.

다음 코드는 Python용 SDK(Boto3) 또는 Java 2.x용 SDK를 사용하여 데이터세트 내보내기 작업을 생성하는 방법을 보여줍니다.

데이터세트를 내보내기 전에 Personalize 서비스 역할이 출력 S3 버킷에 액세스하고 여기에 기록할 수 있는지 확인합니다. [데이터세트 내보내기 작업 권한 요구 사항](#) 섹션을 참조하십시오.

SDK for Python (Boto3)

데이터세트의 데이터를 S3 버킷으로 내보내려면 다음 `create_dataset_export_job`을 사용합니다. 작업에 이름을 부여하고, `dataset_arn`을 내보내려는 데이터세트의 리소스 이름(ARN)으로 바꾸고, `role_arn`를 [Personalize에 대한 IAM 역할 생성](#)에서 생성했던 Personalize 서비스 역할의

ARN으로 바꿉니다. `s3DataDestinationkmsKeyArn`에서는 선택적으로 AWS KMS 키에 대한 ARN을 제공하고 출력의 경우 Amazon S3 버킷의 경로를 `path` 제공하십시오.

`ingestionMode`의 경우 다음 옵션 중에서 내보낼 데이터를 지정합니다.

- 데이터세트 가져오기 작업을 사용하여 대량으로 가져온 데이터만 내보내려면 `BULK`을 지정합니다.
- 콘솔 또는 `PutEvents` `PutUsers`, 또는 `PutItems` 작업을 사용하여 개별적으로 가져온 데이터만 `PUT` 내보내도록 지정합니다.
- 데이터세트의 모든 데이터를 내보내려면 `ALL`를 지정합니다.

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_dataset_export_job(
    jobName = 'job name',
    datasetArn = 'dataset ARN',
    jobOutput = {
        "s3DataDestination": {
            "kmsKeyArn": "kms key ARN",
            "path": "s3://bucket-name/folder-name/"
        }
    },
    roleArn = 'role ARN',
    ingestionMode = 'PUT'
)

dsej_arn = response['datasetExportJobArn']

print ('Dataset Export Job arn: ' + dsej_arn)

description = personalize.describe_dataset_export_job(
    datasetExportJobArn = dsej_arn)['datasetExportJob']

print('Name: ' + description['jobName'])
print('ARN: ' + description['datasetExportJobArn'])
print('Status: ' + description['status'])
```

SDK for Java 2.x

데이터세트 내보내기 작업을 생성하려면 다음 `createDatasetExportJob` 메서드를 사용합니다. `a` PersonalizeClient, 내보내기 작업의 이름, 내보내려는 데이터세트의 ARN, 수집 모드, 출력 Amazon S3 버킷의 경로, 키의 ARN을 파라미터로 전달합니다. AWS KMS

`ingestionMode`은 다음 옵션 중 하나일 수 있습니다.

- 데이터세트 가져오기 작업을 사용하여 대량으로 가져온 데이터만 내보내려면 `IngestionMode.BULK`을 사용합니다.
- 콘솔 또는, 또는 작업을 사용하여 개별적으로 가져온 데이터만 내보내는 데 사용합니다 `IngestionMode.PUT`. `PutEvents` `PutUsers` `PutItems`
- 데이터세트의 모든 데이터를 내보내려면 `IngestionMode.ALL`를 사용합니다.

```
public static void createDatasetExportJob(PersonalizeClient personalizeClient,
                                         String jobName,
                                         String datasetArn,
                                         IngestionMode ingestionMode,
                                         String roleArn,
                                         String s3BucketPath,
                                         String kmsKeyArn) {

    long waitInMilliseconds = 30 * 1000; // 30 seconds
    String status = null;

    try {
        S3DataConfig exportS3DataConfig = S3DataConfig.builder()
            .path(s3BucketPath)
            .kmsKeyArn(kmsKeyArn)
            .build();

        DatasetExportJobOutput jobOutput = DatasetExportJobOutput.builder()
            .s3DataDestination(exportS3DataConfig)
            .build();

        CreateDatasetExportJobRequest createRequest =
        CreateDatasetExportJobRequest.builder()
            .jobName(jobName)
            .datasetArn(datasetArn)
            .ingestionMode(ingestionMode)
            .jobOutput(jobOutput)
```



```
        .roleArn(roleArn)
        .build();

    String datasetExportJobArn =
personalizeClient.createDatasetExportJob(createRequest).datasetExportJobArn();

    DescribeDatasetExportJobRequest describeDatasetExportJobRequest =
DescribeDatasetExportJobRequest.builder()
        .datasetExportJobArn(datasetExportJobArn)
        .build();

    long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

    while (Instant.now().getEpochSecond() < maxTime) {

        DatasetExportJob datasetExportJob =
personalizeClient.describeDatasetExportJob(describeDatasetExportJobRequest)
            .datasetExportJob();

        status = datasetExportJob.status();
        System.out.println("Export job status: " + status);

        if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
            break;
        }
        try {
            Thread.sleep(waitInMilliseconds);
        } catch (InterruptedException e) {
            System.out.println(e.getMessage());
        }
    }
} catch (PersonalizeException e) {
    System.out.println(e.awsErrorDetails().errorMessage());
}
}
```

데이터세트 스키마를 교체하여 새 열 추가

항목 또는 사용자 데이터 세트를 만든 후 해당 스키마를 새 스키마 또는 기존 스키마로 바꿀 수 있습니다. 데이터세트를 생성한 후 데이터 구조가 변경된 경우 데이터세트의 스키마를 교체할 수 있습니다.

예를 들어 항목 메타데이터의 새 열이 있어서 학습 중에 Personalize에서 고려하길 원할 수 있습니다. 또는 추천을 필터링할 때만 사용할 데이터 열을 추가할 수도 있습니다.

데이터세트의 스키마를 바꿀 때는 이전 스키마에 모든 필드를 유지해야 하며 데이터 유형이나 속성은 변경할 수 없습니다. 데이터세트의 스키마를 교체하면 Personalize는 기존 추천자 또는 사용자 지정 솔루션에 대한 학습에서 새 열을 자동으로 제외합니다. 자세한 지침 및 요구 사항은 [지침 및 요구 사항](#) 단원을 참조하세요.

데이터세트의 스키마를 Amazon Personalize 콘솔 AWS Command Line Interface (AWS CLI) 및 SDK로 대체할 수 있습니다. AWS

주제

- [지침 및 요구 사항](#)
- [데이터세트 스키마 교체\(콘솔\)](#)
- [데이터세트의 스키마 교체\(AWS CLI\)](#)
- [데이터세트의 스키마 \(AWS SDK\) 바꾸기](#)

지침 및 요구 사항

데이터세트의 스키마를 교체하기 전에 다음 지침 및 요구 사항을 알고 있어야 합니다.

- 항목 상호 작용 데이터 세트, 작업 상호 작용 데이터 세트 또는 작업 데이터 세트의 스키마는 대체할 수 없습니다.
- 교체 스키마에 새 필드를 추가할 수 있지만 이전 스키마의 모든 필드를 유지해야 합니다. 데이터 유형이나 속성은 변경할 수 없습니다. 예를 들어 이전 스키마에 범주형 문자열 데이터용 MEMBERSHIP_STATUS 필드가 포함된 경우 사용하는 새 스키마에는 이러한 속성 및 데이터 유형의 MEMBERSHIP_STATUS 필드가 포함되어야 합니다.
- 현재 스키마에 이름을 바꾸려는 필드가 있거나 스키마의 데이터 유형 또는 속성을 변경하려는 경우 새 이름과 수정된 유형 또는 속성을 가진 새 필드를 추가할 수 있습니다. 그런 다음 새 필드를 학습에 포함시키고 이전 필드는 제외시킵니다. 모든 새 필드는 null 데이터를 지원해야 합니다. 이전 필드가 Null 데이터를 지원하지 않는 경우 데이터를 가져올 때 자리 표시자 데이터를 사용하여 가져오기가 스키마와 일치하는지 확인할 수 있습니다. 추천자가 사용하는 열을 구성하는 방법에 대한 자세한 내용은 [추천자 업데이트](#) 단원을 참조하세요. 솔루션이 사용하는 열을 구성하는 방법에 대한 자세한 내용은 [교육 시 사용되는 열 구성](#) 단원을 참조하세요.
- 모든 새 필드는 null 데이터를 지원해야 합니다. 필드에 Null 유형을 추가하는 방법에 대한 자세한 내용은 [스키마 데이터 유형](#) 단원을 참조하세요.

- 데이터세트의 스키마를 교체하면 Personalize는 기존 추천자 또는 사용자 지정 솔루션에 대한 학습에서 새 열을 자동으로 제외합니다. 수정된 데이터세트를 사용하려면 다음과 같은 작업이 필요합니다.
- 학습에 새 열을 사용하려면 새 스키마에 맞는 데이터를 가져옵니다. 그런 다음 새 열을 사용하도록 추천자를 업데이트하거나, 새 사용자 지정 솔루션을 만들고 학습 시 사용할 열을 구성합니다.

추천자가 사용하는 열을 업데이트하는 방법에 대한 자세한 내용은 [추천자 업데이트](#) 단원을 참조하세요. 솔루션이 사용하는 열을 구성하는 방법에 대한 자세한 내용은 [교육 시 사용되는 열 구성](#) 단원을 참조하세요.

- 필터링할 때만 열을 사용하려면 새 스키마에 맞는 데이터를 가져오고 새 데이터를 사용하는 필터를 만든 다음 필터를 추천 요청에 적용합니다. 추천자를 업데이트하거나 사용자 지정 리소스를 만들거나 업데이트할 필요가 없습니다.

데이터세트 스키마 교체(콘솔)

Personalize 콘솔에서 데이터세트의 스키마를 교체하려면 수정할 데이터세트를 선택하고 새 스키마로 바꾸거나 기존 스키마를 사용하도록 선택합니다.

데이터세트의 스키마를 교체하려면

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터세트 그룹 페이지에서 데이터세트 그룹을 선택합니다.
3. 탐색 창에서 Datasets를 선택하고 수정하려는 데이터세트의 라디오 버튼을 선택합니다.
4. 작업을 선택하고 스키마 교체를 선택합니다.
5. 스키마 세부 정보에서 새 스키마로 교체할지 또는 이전에 만든 스키마로 교체할지 선택합니다.
6. 사용할 새 스키마를 지정합니다. 다음을 선택한 경우:
 - 새 스키마로 교체하도록 선택한 경우, 스키마에 이름을 지정한 다음 스키마 정의에서 스키마 JSON을 변경합니다.
 - 이전에 생성한 스키마를 사용하도록 선택한 경우, 이전에 생성한 스키마에 대해 사용할 스키마를 선택합니다. 적합한 스키마만 나열됩니다. 스키마 요구 사항에 대한 자세한 내용은 [지침 및 요구 사항](#) 단원을 참조하세요.
7. 바꾸기를 선택합니다. 데이터세트가 활성화되면 새 스키마에 맞는 데이터 가져오기를 시작할 수 있습니다. 자세한 내용은 [단계 2: 데이터 준비 및 가져오기](#) 단원을 참조하세요.

데이터세트의 스키마 교체(AWS CLI)

데이터세트의 스키마를 로 바꾸려면 `update-dataset` 명령을 사용하여 업데이트할 데이터세트의 Amazon 리소스 이름 (ARN) 과 사용할 새 스키마의 ARN을 지정합니다. AWS CLI항목 상호 작용 데이터 세트, 작업 상호 작용 데이터 세트 또는 작업 데이터 세트의 스키마는 업데이트할 수 없습니다.

다음 코드는 AWS CLI에서 데이터세트의 스키마를 업데이트하는 방법을 보여줍니다. 데이터세트의 스키마를 새 스키마로 바꾸려면 먼저 `create-schema`명령을 사용합니다. 그런 후 다음 코드를 사용하여 현재 스키마를 새 스키마로 교체합니다. 를 사용하여 스키마를 생성하는 방법에 대한 자세한 내용은 참조하십시오. AWS CLI [데이터세트 및 스키마 생성\(AWS CLI\)](#) 데이터세트 및 스키마 요구 사항에 대한 자세한 내용은 [스키마](#) 단원을 참조하십시오.

```
aws personalize update-dataset \
--dataset-arn Dataset ARN \
--schema-arn New schema ARN
```

데이터세트가 활성화되면 새 스키마에 맞는 데이터 가져오기를 시작할 수 있습니다. 자세한 내용은 [단계 2: 데이터 준비 및 가져오기](#) 단원을 참조하십시오. 데이터세트의 최신 업데이트에 대한 자세한 내용은 [DescribeDataset](#) 작업을 사용할 수 있습니다.

데이터세트의 스키마 (AWS SDK) 바꾸기

데이터세트의 스키마를 AWS SDK로 바꾸려면 API 작업을 사용합니다. `UpdateDataset` 업데이트할 데이터세트의 리소스 이름(ARN)과 사용할 새 스키마를 지정합니다. 항목 상호 작용 데이터 세트, 작업 상호 작용 데이터 세트 또는 작업 데이터 세트의 스키마는 업데이트할 수 없습니다.

다음 코드는 Python용 SDK(Boto3)에서 데이터세트의 스키마를 교체하는 방법을 보여줍니다. 데이터 세트의 스키마를 새 스키마로 바꾸려면 먼저 [CreateSchema](#) 작업을 사용합니다. 그런 후 다음 코드를 사용하여 현재 스키마를 새 스키마로 교체합니다. AWS SDK로 스키마를 만드는 방법에 대한 자세한 내용은 참조하십시오. [데이터세트 및 스키마 \(SDK\) 생성AWS](#) 데이터세트 및 스키마 요구 사항에 대한 자세한 내용은 [스키마](#) 단원을 참조하십시오.

```
import boto3

personalize = boto3.client('personalize')

update_dataset_response = personalize.update_dataset(
    datasetArn = 'dataset_arn',
    schemaArn = 'new_schema_arn'
```

```
)

print(update_dataset_response)
```

데이터세트가 활성화되면 새 스키마에 맞는 데이터 가져오기를 시작할 수 있습니다. 자세한 내용은 [단계 2: 데이터 준비 및 가져오기](#) 단원을 참조하세요. 데이터세트의 최신 업데이트에 대한 자세한 내용은 [DescribeDataset](#) 작업을 사용할 수 있습니다.

데이터 삭제 작업을 통한 사용자 및 데이터 삭제

데이터를 가져온 후 데이터세트 그룹에서 메타데이터 및 상호작용 데이터를 포함한 사용자 및 해당 데이터를 삭제할 수 있습니다. 규정 준수 프로그램의 일환으로, 사용자 삭제 요청을 처리하기 위해 또는 사용자 기반 변화에 따라 데이터를 최신 상태로 유지하기 위해 사용자 데이터를 삭제할 수 있습니다.

사용자를 삭제하면 Amazon Personalize는 더 이상 사용자 데이터를 트레이닝하지 않으며 사용자 세그먼트를 생성할 때 사용자를 고려하지 않습니다.

Amazon Personalize 데이터 세트 및 데이터 세트 그룹의 모델에 있는 사용자에 대한 참조를 삭제하려면 다음과 같이 하십시오.

1. USER_ID 열에 삭제할 사용자의 사용자 ID를 나열하는 CSV 파일을 준비하십시오.
2. CSV 파일을 Amazon S3 버킷에 업로드합니다. Amazon Personalize 서비스 역할에는 이 버킷에 액세스할 수 있는 권한이 있어야 합니다.
3. 데이터 삭제 작업을 생성하십시오. 데이터 삭제 작업은 데이터세트 그룹의 모델 및 데이터세트에서 사용자와 해당 데이터를 삭제하는 일괄 작업입니다.

주제

- [지침 및 요구 사항](#)
- [삭제할 사용자 목록 준비](#)
- [데이터 삭제 작업 생성](#)

지침 및 요구 사항

다음은 사용자 삭제에 대한 지침 및 요구 사항입니다.

- 데이터 삭제 작업을 생성하기 전에 데이터세트를 사용하는 작업 (예: 교육 작업, 일괄 작업, 대량 또는 개별 가져오기 작업) 이 진행 중이지 않은지 확인하세요. 그리고 데이터 삭제 작업이 진행 중인 동

안에는 이러한 작업을 만들지 마세요. 교육 또는 가져오기가 발생하는 경우 사용자 데이터가 모델에서 삭제된다고 보장할 수 없으므로 추가 데이터 삭제 작업을 생성하는 것이 좋습니다.

- 데이터 삭제 작업은 Amazon Personalize 외부 사용자에게 대한 참조를 삭제하지 않습니다. 예를 들어 Amazon S3 버킷의 배치 추천에서 해당 사용자의 UserID를 삭제하지는 않습니다. 이러한 레코드는 수동으로 삭제해야 합니다.
- PENDING 상태인 데이터세트 그룹당 최대 5개의 삭제 작업을 수행할 수 있습니다.
- 데이터 삭제 입력 파일 또는 파일의 최대 총 크기는 50MB입니다. 삭제 작업을 생성할 때 동일한 입력 파일을 재사용할 수 있습니다.
- 각 데이터 삭제 작업은 데이터세트 그룹에서 사용자와 사용자의 상호작용 데이터를 삭제합니다. 모든 데이터세트 그룹에서 데이터를 삭제하려면 각 데이터세트 그룹에 대한 데이터 삭제 작업을 만들어야 합니다.
- 작업을 생성한 후 데이터세트와 모델에서 사용자 데이터를 삭제하는 데 최대 하루가 걸릴 수 있습니다.
- 작업이 완료된 후에는 모든 커스텀 리소스를 업데이트해야 합니다. 새 솔루션 버전을 만들고 필요한 경우 캠페인을 업데이트하세요. 자동 교육을 사용하는 경우에도 새 솔루션 버전을 수동으로 만들 수 있습니다.
- Amazon Personalize 서비스 역할에는 삭제할 사용자 목록이 있는 Amazon S3 버킷에 액세스할 수 있는 권한이 있어야 합니다. GetObject버킷과 ListBucket 해당 콘텐츠에 대한 필요 및 권한이 필요합니다. 이러한 권한은 데이터 가져오기와 동일합니다. 권한 부여 및 정책 예제에 대한 자세한 내용은 [Personalize에 S3 리소스에 대한 액세스 권한 부여](#)를 참조하십시오.
- 삭제할 사용자의 사용자 ID 목록을 저장하는 Amazon S3 버킷에서는 자체 AWS Key Management Service 키를 사용할 수 없습니다.
- 항목이 항목 상호 작용 데이터 세트 데이터 세트에만 나타나고 삭제하려는 사용자만 이 항목과 상호 작용한 경우 이 항목은 더 이상 권장 사항에 표시되지 않습니다.

삭제할 사용자 목록 준비

Amazon Personalize에서 사용자를 삭제하기 전에 삭제할 사용자 목록을 CSV 파일로 준비하여 Amazon S3에 업로드해야 합니다.

삭제하고 업로드할 사용자 목록을 준비하려면

1. 삭제할 사용자의 사용자 ID가 나열된 CSV 파일을 생성합니다. 다음은 CSV 파일의 형식을 지정하는 방법을 보여줍니다.

```

USER_ID
abc
2a
5basc
ab35
123f
a55d
0v22
441fa
efg

```

2. CSV 파일을 아마존 심플 스토리지 서비스 (Amazon S3) 버킷에 업로드합니다. Amazon S3에 파일을 업로드하는 방법에 대한 자세한 내용은 Amazon Simple Storage 서비스 사용 설명서의 [드래그 앤 드롭을 사용한 파일 및 폴더 업로드](#)를 참조하십시오.
3. Amazon Personalize에 버킷과 CSV 파일에 대한 액세스 권한을 부여하십시오. Amazon Personalize에는 버킷 GetObject 및 해당 콘텐츠에 대해 ListBucket 작업을 수행할 권한이 있어야 합니다. 이러한 권한은 데이터 가져오기와 동일합니다. 권한 부여 및 정책 예제에 대한 자세한 내용은 [Personalize에 S3 리소스에 대한 액세스 권한 부여](#)를 참조하십시오.

데이터 삭제 작업 생성

작업을 [삭제할 사용자 목록 준비](#) 완료하고 나면 데이터 삭제 작업으로 사용자를 삭제할 준비가 된 것입니다.

데이터 삭제 작업은 데이터세트 그룹의 모델 및 데이터세트에서 사용자와 해당 데이터를 삭제하는 일괄 작업입니다. 사용자를 삭제하면 Amazon Personalize는 더 이상 사용자 데이터를 트레이닝하지 않으며 사용자 세그먼트를 생성할 때 사용자를 고려하지 않습니다.

데이터 삭제 작업을 생성할 때 삭제할 사용자 목록의 Amazon S3 위치를 지정합니다.

- 데이터가 단일 파일에 있는 경우 Amazon S3 위치에 다음 구문을 사용하십시오.

s3://<name of your S3 bucket>/<folder path>/<CSV filename>.csv

- CSV 파일이 Amazon S3 버킷의 폴더에 있는 경우 폴더 경로를 지정할 수 있습니다. 데이터 삭제 작업에서 Amazon Personalize는 폴더 및 하위 폴더에서 .csv 파일 확장자가 있는 모든 파일을 사용합니다. 다른 유형의 파일은 무시합니다. 폴더 이름 뒤에 /가 있는 다음 구문을 사용하세요.

s3://<name of your S3 bucket>/<folder path>/

사용하는 역할에는 Amazon S3 버킷 GetObject 및 해당 콘텐츠에서 및 ListBucket 작업을 수행할 수 있는 권한이 있어야 합니다. 권한 부여 및 정책 예제에 대한 자세한 내용은 [을 참조하십시오](#) [Personalize에 S3 리소스에 대한 액세스 권한 부여](#).

Amazon Personalize 콘솔, AWS Command Line Interface (AWS CLI) 또는 AWS SDK를 사용하여 데이터 삭제 작업을 생성할 수 있습니다.

데이터 삭제 작업 생성 (콘솔)

Amazon Personalize 콘솔에서 사용자를 삭제하려면 데이터의 이름, IAM 서비스 역할 및 Amazon S3 위치를 사용하여 데이터 삭제 작업을 생성하십시오.

레코드를 삭제하려면 (콘솔)

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터세트 그룹 페이지에서 데이터세트 그룹을 선택합니다. 데이터세트 그룹 개요가 표시됩니다.
3. 탐색 창에서 데이터세트를 선택합니다.
4. 데이터 삭제 작업에서 작업 생성을 선택합니다.
5. Job details (Job details) 에서 작업 이름을 지정합니다.
6. S3 입력 소스에서 S3 위치에 대해 삭제할 사용자의 사용자 ID 목록을 저장하는 CSV 파일의 Amazon S3 위치를 지정합니다. 에서 이 파일을 준비했습니다. [삭제할 사용자 목록 준비](#)
7. IAM 역할에서 새 역할 생성 또는 기존 역할 사용 중 하나를 선택합니다. Amazon Personalize용 역할을 생성하기 위한 사전 요구 사항을 완료하고 이 역할에 Amazon S3 버킷에 대한 액세스 권한을 부여한 경우, 기존 서비스 역할 사용을 선택하고 생성한 역할을 지정합니다. [Personalize에 대한 IAM 역할 생성](#)

사용하는 역할에는 Amazon S3 버킷 GetObject 및 해당 콘텐츠에서 및 ListBucket 작업을 수행할 수 있는 권한이 있어야 합니다. 이러한 권한은 데이터 가져오기와 동일합니다. 권한 부여 및 정책 예제에 대한 자세한 내용은 [을 참조하십시오](#) [Personalize에 S3 리소스에 대한 액세스 권한 부여](#)

8. 태그의 경우, 원하는 태그를 추가할 수 있습니다. Personalize 리소스 태그 지정에 대한 자세한 내용은 [Personalize 리소스에 태그 지정](#) 단원을 참조하세요.
9. 작업 생성을 선택합니다. 작업이 시작되고 세부 정보 페이지가 표시됩니다.

작업을 만든 후 데이터세트와 모델에서 사용자 데이터를 삭제하는 데 하루 정도 걸릴 수 있습니다. 작업이 완료될 때까지 Amazon Personalize는 교육 시 데이터를 계속 사용합니다. 또한 사용자가 사용자 세그먼트에 나타날 수도 있습니다.

상태가 COMPLETED로 표시되면 데이터 삭제가 완료된 것입니다. 어떤 이유로든 작업이 실패할 경우 다른 데이터 삭제 작업을 생성하는 것이 좋습니다. 작업이 완료된 후에는 모든 사용자 지정 리소스를 업데이트해야 합니다. 새 솔루션 버전을 만들고 필요한 경우 캠페인을 업데이트하세요. 자동 교육을 사용하는 경우에도 새 솔루션 버전을 수동으로 만들 수 있습니다.

데이터 삭제 작업 생성 (AWS CLI)

를 AWS CLI 사용하여 사용자를 삭제하려면 `create-data-deletion-job` 명령을 사용합니다. 이 명령은 `CreateDataDeletion` API 작업을 사용합니다. 다음 코드는 데이터 삭제 작업을 생성하는 방법을 보여줍니다. 코드를 사용하려면 코드를 업데이트하여 작업 이름, 생성한 IAM 역할 [Personalize에 대한 IAM 역할 생성](#), 데이터의 Amazon S3 위치를 지정하십시오. 에서 이 파일을 준비했습니다 [삭제할 사용자 목록 준비](#).

```
aws personalize create-data-deletion-job \
--job-name deletion job name \
--dataset-group-arn dataset group ARN \
--data-source dataLocation=s3://bucketname/filename.csv \
--role-arn roleArn
```

작업을 만든 후 데이터세트와 모델에서 사용자 데이터를 삭제하는 데 하루 정도 걸릴 수 있습니다. 작업이 완료될 때까지 Amazon Personalize는 교육 시 데이터를 계속 사용합니다. 또한 사용자가 사용자 세그먼트에 나타날 수도 있습니다.

상태가 완료되면 작업이 완료된 것입니다. `describe-data-deletion-job` 명령을 사용하여 상태를 확인하고 데이터 삭제 작업 ARN을 지정합니다. API 작업에 대한 자세한 내용은 을 참조하십시오 [DescribeDataDeletionJob](#). 생성 시간별로 정렬된 데이터 삭제 작업 기록을 보려면 [ListDataDeletionJobs](#) API 작업을 사용하십시오.

어떤 이유로든 작업이 실패할 경우 다른 데이터 삭제 작업을 생성하는 것이 좋습니다. 작업이 완료된 후에는 모든 사용자 지정 리소스를 업데이트해야 합니다. 새 솔루션 버전을 만들고 필요한 경우 캠페인을 업데이트하세요. 자동 교육을 사용하는 경우에도 새 솔루션 버전을 수동으로 만들 수 있습니다.

데이터 삭제 작업 (AWS SDK) 생성

AWS SDK를 사용하여 사용자를 삭제하려면 [CreateDataDeletionJob](#) API 작업을 사용하십시오. 다음 코드는 데이터 삭제 작업을 생성하는 방법을 보여줍니다. 코드를 사용하려면 코드를 업데이트하여 작업 이름, 생성한 IAM 역할 [Personalize에 대한 IAM 역할 생성](#), 데이터의 Amazon S3 위치를 지정하십시오. 에서 이 파일을 준비했습니다 [삭제할 사용자 목록 준비](#).

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_data_deletion_job(
    jobName = 'Deletion job name',
    datasetGroupArn = 'Dataset Group ARN',
    dataSource = {'dataLocation': 's3://bucket/file.csv'},
    roleArn = 'role_arn'
)

deletion_job_arn = response['dataDeletionJobArn']

print ('Deletion Job arn: ' + deletion_job_arn)

description = personalize.describe_data_deletion_job(
    dataDeletionJobArn = deletion_job_arn)['dataDeletionJob']

print('Name: ' + description['jobName'])
print('ARN: ' + description['dataDeletionJobArn'])
print('Status: ' + description['status'])
```

작업을 만든 후 데이터세트와 모델에서 사용자 데이터를 삭제하는 데 하루 정도 걸릴 수 있습니다. 작업이 완료될 때까지 Amazon Personalize는 교육 시 데이터를 계속 사용합니다. 또한 사용자가 사용자 세그먼트에 나타날 수도 있습니다.

상태가 완료되면 작업이 완료된 것입니다. [DescribeDataDeletionJob](#) 작업을 사용하여 상태를 확인하고 데이터 삭제 작업 ARN을 지정합니다. 생성 시간별로 정렬된 데이터 삭제 작업 기록을 보려면 [ListDataDeletionJobs](#) API 작업을 사용하십시오.

어떤 이유로든 작업이 실패할 경우 다른 데이터 삭제 작업을 생성하는 것이 좋습니다. 작업이 완료된 후에는 모든 사용자 지정 리소스를 업데이트해야 합니다. 새 솔루션 버전을 만들고 필요한 경우 캠페인을 업데이트하세요. 자동 교육을 사용하는 경우에도 새 솔루션 버전을 수동으로 만들 수 있습니다.

데이터세트를 삭제하여 해당 데이터를 모두 삭제합니다.

데이터세트의 모든 데이터를 삭제하려면 데이터세트를 삭제합니다. 데이터세트 가져오기 작업 또는 솔루션 버전이 CREATE PENDING 또는 IN PROGRESS 상태인 경우, 데이터세트를 삭제할 수 없습니다. 사용자 맞춤설정-v2, 사용자 맞춤화 또는 차세대 베스트 액션 레시피나 사용 사례에 맞는 추천 제품을 사용하는 경우, 데이터세트를 삭제하면 관련 솔루션 버전 또는 추천에 대한 자동 업데이트가 중단됩니다.

Amazon Personalize 콘솔 AWS Command Line Interface (AWS CLI) 또는 SDK를 사용하여 데이터세트를 삭제할 수 있습니다. AWS

주제

- [데이터세트 삭제\(콘솔\)](#)
- [데이터세트 삭제\(AWS CLI\)](#)
- [데이터세트 삭제\(AWS SDK\)](#)

데이터세트 삭제(콘솔)

Personalize 콘솔로 데이터세트를 삭제하려면 데이터세트 세부 정보 페이지로 이동하여 삭제를 선택합니다.

데이터세트를 삭제하려면

1. <https://console.aws.amazon.com/personalize/home>에서 Personalize 콘솔을 엽니다.
2. 탐색 창에서 데이터세트 그룹을 선택합니다.
3. 데이터세트 그룹 페이지에서 데이터세트 그룹을 선택합니다.
4. 탐색 창에서 데이터세트를 선택합니다.
5. 데이터세트를 선택하여 해당 세부 정보 페이지를 엽니다.
6. 데이터세트의 세부정보 페이지에서 삭제를 선택하고 데이터세트 삭제를 확인합니다.

데이터세트 삭제(AWS CLI)

다음 코드는 AWS CLI 및 작업을 사용하여 데이터세트를 삭제하는 방법을 보여줍니다. [DeleteDataset](#)

```
aws personalize delete-dataset --dataset-arn dataset-arn
```

데이터세트 삭제(AWS SDK)

다음 코드는 AWS SDK로 데이터세트를 삭제하는 방법과 작업을 보여줍니다. [DeleteDataset](#)

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.delete_dataset(
    datasetArn = 'dataset ARN'
)
```

SDK for Java 2.x

```
public static void deleteDataset(PersonalizeClient personalizeClient,
                                String datasetArn) {

    try {
        DeleteDatasetRequest deleteRequest = DeleteDatasetRequest.builder()
            .datasetArn(datasetArn)
            .build();

        int responseCode =
personalizeClient.deleteDataset(deleteRequest).sdkHttpResponse().statusCode();
        System.out.println(responseCode);
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
}
```

추천 및 사용자 세그먼트 필터링

도메인 추천자 또는 사용자 지정 캠페인을 사용하여 추천을 받는 경우 사용자 지정 기준에 따라 결과를 필터링할 수 있습니다. 예를 들어 사용자가 이미 구매한 제품은 추천하지 않거나 특정 연령대에 맞는 제품만 추천하고 싶을 수 있습니다.

마찬가지로 사용자_세분화 레시피를 사용하면 사용자 세그먼트에 특정 유형의 사용자를 포함시키고 싶지 않을 수 있습니다. 결과를 필터링하여 사용자에게 추천할 항목이나 사용자 세그먼트에 포함시킬 사용자를 제어할 수 있습니다.

Amazon Personalize 콘솔, AWS Command Line Interface (AWS CLI) 및 SDK를 사용하여 필터를 생성, 편집, 삭제 및 적용할 수 있습니다. AWS

- 실시간 추천의 경우 GetRecommendations GetActionRecommendations, 또는 GetPersonalizedRanking 작업을 호출할 때 필터를 적용하고 필터 파라미터 값을 지정합니다. 콘솔에서 캠페인 또는 추천자로부터 추천을 받을 때도 필터를 적용할 수 있습니다.

개인 맞춤형 또는 관련 항목 레시피 또는 사용 사례가 포함된 실시간 항목 추천을 받으면 요청에 홍보를 지정할 수 있습니다. 홍보는 필터를 사용하여 추천 항목의 구성 가능한 하위 집합에 적용되는 추가 비즈니스 규칙을 정의합니다. 자세한 내용은 [추천의 항목 홍보](#) 단원을 참조하세요.

- 배치 워크플로우의 경우 입력 JSON에 모든 필터 파라미터 값을 포함시킬 수 있습니다. 그런 다음 배치 추론 작업 또는 배치 세그먼트 작업을 생성할 때 필터의 리소스 이름(ARN)을 지정합니다. 자세한 내용은 [배치 추천 및 사용자 세그먼트\(사용자 지정 리소스\) 필터링](#) 단원을 참조하세요.

새 레코드에 대한 필터 업데이트

PutEvents 또는 PutActionInteractions 작업을 통해 가져오는 데이터의 경우 Amazon Personalize는 가져온 후 몇 초 내에 데이터세트 그룹의 모든 필터를 새 데이터로 업데이트합니다. 예를 들어 필터가 추천에서 구매한 항목을 제거하고 해당 PutEvents 작업을 수행하는 사용자에게 대한 구매 이벤트를 기록하는 경우 이 항목은 이벤트를 기록한 후 몇 초 내에 이 사용자에게 대한 향후 권장 사항에서 제거됩니다.

대량으로 또는 개별적으로 가져온 다른 모든 데이터에 대해 Amazon Personalize는 데이터 세트 그룹의 필터를 마지막으로 가져온 후 20분 이내에 새 데이터로 업데이트합니다.

주제

- [필터 표현식](#)
- [실시간 추천 필터링](#)

- [배치 추천 및 사용자 세그먼트\(사용자 지정 리소스\) 필터링](#)

필터 표현식

필터를 구성하기 위해서는 올바른 형식의 필터 표현식을 사용해야 합니다. 필터 표현식은 `dataset.field` 형식의 데이터셋 및 필드 식별자와 함께 논리 연산자, 키워드, 값으로 구성됩니다. 값의 경우 고정 값을 지정하거나 자리 표시자 파라미터를 추가하고 추천을 받을 때의 필터 기준을 설정할 수 있습니다.

필터 표현식을 사용하여 다음 데이터셋의 데이터를 기반으로 추천의 항목, 사용자 또는 작업을 필터링할 수 있습니다.

- **항목 상호 작용:** 필터 식을 사용하여 상호 작용 데이터를 기반으로 항목 또는 사용자를 포함하거나 제외할 수 있습니다. 예를 들어 항목 추천의 경우 사용자가 클릭한 항목을 제외하거나 (항목 추천의 경우) 항목에 등급을 매긴 사용자만 포함할 수 있습니다 (항목-선호도 레시피의 경우). 모든 레시피 유형에서 이벤트 유형으로만 필터링할 수 있습니다. 상황별 메타데이터와 같은 다른 상호작용 메타데이터를 기반으로 필터링할 수는 없습니다. 항목 상호 작용 필터는 와 함께 사용할 수 없습니다. [항목-속성-친화도 레시피](#)

Personalize는 이벤트 유형별로 사용자당 가장 최근의 상호작용을 최대 100건까지 고려합니다. 이 할당량은 조정 가능한 할당량입니다. [Service Quotas 콘솔을 사용하여 IAM 할당량 증가를](#) 요청할 수 있습니다.

- **작업 상호 작용:** 필터 식을 사용하여 이벤트 유형에 따라 사용자가 상호 작용한 작업을 포함하거나 제외할 수 있습니다. 예를 들어, 사용자가 이미 수행한 작업을 제외할 수 있습니다. 다른 동작 상호 작용 메타데이터를 기반으로 필터링할 수는 없습니다.

Amazon Personalize는 이벤트 유형별로 사용자당 가장 최근의 작업 상호 작용을 최대 300건까지 고려합니다. 이 할당량은 조정 가능한 할당량입니다. [Service Quotas 콘솔을 사용하여 IAM 할당량 증가를](#) 요청할 수 있습니다.

- **항목:** 필터 표현식을 사용하여 특정 항목 조건에 따라 항목을 포함하거나 제외할 수 있습니다. 필터를 사용해도 제품 설명과 같은 구조화되지 않은 텍스트 항목 메타데이터를 기반으로 항목을 포함하거나 제외할 수 없습니다. 도메인 사용 사례 또는 사용자 지정 레시피가 유사-항목 레시피 또는 X보다 좋은 제품 도메인 사용 사례와 같은 관련 항목 추천을 생성하는 경우 필터 표현식을 사용하여 추천 요청에서 지정한 항목의 속성을 기반으로 항목을 포함하거나 제외할 수 있습니다.
- **사용자:** 항목 및 작업 추천의 경우 사용자 데이터 세트가 있으면 `CurrentUser`를 기반으로 항목 또는 작업을 제외하거나 포함할 수 있습니다. 개인 맞춤형 추천, 인기 항목, 작업 추천의 경우 추천을 받

는 사용자가 바로 이 사용자입니다. 관련 항목의 경우 추천 요청에 지정할 수 있는 선택형 사용자입니다.

사용자 세그먼트의 경우 필터 표현식을 사용하여 속성(예: Users.MEMBERSHIP_STATUS)에 따라 사용자 세그먼트에 사용자를 포함하거나 제외할 수 있습니다.

- 작업: 필터 표현식을 사용하여 특정 작업 조건에 따라 작업을 포함하거나 제외할 수 있습니다. Amazon Personalize는 Action expiration timestamp 및 Repeat frequency 데이터에 기반한 작업을 자동으로 제외합니다. 이 데이터를 기반으로 필터링하는 추가 사용자 지정 필터를 생성할 수 없습니다.

필터 표현식 요소의 전체 목록은 [필터 표현식 요소](#) 단원을 참조하세요. 필터 표현식의 예는 [필터 표현식 예제](#) 단원을 참조하세요.

주제

- [지침 및 요구 사항](#)
- [필터 표현식 구조 및 요소](#)
- [필터 표현식 예제](#)

지침 및 요구 사항

필터 표현식을 만들 경우 다음 지침 및 요구 사항을 참고하세요.

- 필터를 사용해도 제품 설명과 같은 구조화되지 않은 텍스트 항목 메타데이터를 기반으로 항목을 포함하거나 제외할 수 없습니다.
- 항목 또는 동작 상호 작용 데이터를 기반으로 필터링하는 경우 이벤트 유형을 기반으로 필터링만 할 수 있습니다. 다른 상호 작용 메타데이터 (예: 상황별 메타데이터) 를 기반으로 필터링할 수 없습니다.
- Personalize는 이벤트 유형을 일치시켰을 때만 대소문자를 무시합니다.
- 항목 상호 작용 및 항목 데이터 세트를 하나의 표현식에 사용할 수는 없습니다. 상호작용을 기준으로 필터링한 다음 항목 데이터 세트를 기준으로 필터링(또는 그 반대로)하는 필터를 만들려면 둘 이상의 표현식을 함께 연결해야 합니다. 자세한 정보는 [여러 표현식 결합](#)을 참조하세요.
- 항목 상호 작용 및 작업 데이터 세트를 하나의 표현식에 사용할 수는 없습니다. 항목 상호 작용을 기준으로 필터링한 다음 작업 데이터 세트를 기준으로 필터링(또는 그 반대로)하는 필터를 만들려면 둘 이상의 표현식을 함께 연결해야 합니다. 자세한 정보는 [여러 표현식 결합](#)을 참조하세요.
- 항목 상호 작용 필터는 와 함께 사용할 수 없습니다. [항목-속성-친화도 레시피](#)

- 스키마에서 부울 형식의 값을 사용하여 필터링하는 필터 표현식은 만들 수 없습니다. 부울 값을 기준으로 필터링하려면 필드가 문자열 유형인 스키마를 사용하고 데이터에 값 "True" 및 "False"를 사용합니다. 또는 int 또는 long 유형과 0 및 1 값을 사용할 수 있습니다.
- 필터, 한 표현식 또는 함께 연결된 여러 표현식에 사용할 수 있는 개별 데이터세트 필드의 최대 수는 5개입니다. 데이터세트 그룹의 모든 필터를 통틀어 고유한 데이터세트 필드의 최대 수는 10개입니다.
- 도메인 사용 사례 또는 사용자 지정 레시피가 Similar-Items 레시피 또는 More Like X 도메인 사용 사례와 같은 관련 항목 추천을 생성하는 경우에만 CurrentItem 요소와 함께 필터를 적용할 수 있습니다.
- NOT_IN 연산자를 사용하는 필터 표현식에는 자리 표시자 파라미터를 사용할 수 없습니다. 대신 IN 연산자를 사용하고 반대의 작업을 사용합니다. 예를 들면 Exclude 대신 Include를 사용합니다(또는 그 반대).
- Action expiration timestamp 및 Repeat frequency 데이터를 기준으로 필터링하는 필터는 만들 수 없습니다. Amazon Personalize는 이 데이터를 기반으로 작업 추천을 자동으로 필터링합니다.

필터 표현식 구조 및 요소

이 섹션에는 필터 표현식의 구조와 요소에 대한 정보가 포함되어 있습니다.

주제

- [필터 표현식 구조](#)
- [필터 표현식 요소](#)

필터 표현식 구조

필터 표현식의 일반적인 구조는 다음과 같습니다.

```
EXCLUDE/INCLUDE ItemID/ActionID/UserID WHERE dataset type.field IN/NOT IN (value/parameter)
```

수동으로 필터 표현식을 만들거나 콘솔의 [표현식 빌더](#)를 사용하여 표현식 구문 및 구조에 대한 도움을 받을 수 있습니다.

필터 표현식 요소

필터 표현식을 만들려면 다음 요소를 사용합니다.

INCLUDE 또는 EXCLUDE

필터 기준을 충족하는 항목으로만 추천을 제한하려면 INCLUDE를 사용하고 또는 필터 기준을 충족하는 모든 항목을 제거하려면 EXCLUDE를 사용합니다.

ItemID/ActionID/UserID

INCLUDE 또는 EXCLUDE 요소 뒤에 다음 요소 중 하나를 사용합니다. 항목(항목 추천용), 작업(작업 추천용) 또는 사용자(사용자 세그먼트용)를 필터링하는지에 따라 사용하는 요소가 달라집니다.

WHERE

WHERE는 항목, 작업 또는 사용자의 조건을 확인하는 데 사용합니다. WHERE 요소는 ItemID, ActionID 또는 UserID 뒤에 사용해야 합니다.

AND/OR

동일한 필터 표현식 내에서 여러 조건을 함께 연결하려면 AND또는 OR를 사용합니다. AND 또는 OR를 사용하여 서로 연결된 조건은 첫 번째 조건에 사용된 데이터세트의 필드에만 영향을 줄 수 있습니다.

Dataset.field

추천 필터링 기준으로 사용할 데이터세트와 메타데이터 필드를 dataset.field의 형식으로 제공합니다. 예를 들어, 항목 데이터 세트의 장르 필드를 기반으로 항목 상호 작용을 필터링하려면 필터 표현식에 Items.genres를 사용합니다.

IF 조건

IF 조건은 CurrentUser의 조건을 확인할 때만 사용하고 표현식 끝에서 한 번만 사용합니다. 하지만 AND를 사용하여 IF조건을 확장할 수 있습니다.

CurrentUser.attribute

추천을 받는 사용자를 기준으로 항목 추천을 필터링하려면 IF 조건에서만 CurrentUser를 사용하고 사용자 필드를 제공합니다. 예를 들어 CurrentUser.AGE입니다.

CurrentItem. 속성

관련 항목 레시피 및 사용 사례의 경우에만 관련 항목 추천 요청에서 지정한 항목의 속성을 기준으로 항목을 필터링하려면 CurrentItem.attribute를 사용합니다. 예: CurrentItem.GENRE또는 CurrentItem.PRICE.

도메인 사용 사례 또는 사용자 지정 레시피가 Similar-Items 레시피 또는 More Like X 도메인 사용 사례와 같은 관련 항목 추천을 생성하는 경우에만 CurrentItem 요소와 함께 필터를 적용할 수 있습니다. CurrentItem 요소를 사용하여 필터를 처음 만들 때는 필터 생성에 몇 분 정도 걸릴 수 있습니다. 암호화에 사용하는 경우 필터 AWS KMS 생성에 최대 15분이 소요될 수 있습니다.

IN/NOT IN

하나 이상의 문자열 값이 일치하거나 일치하지 않는 것을 기준으로 필터링하려면 IN 또는 NOT IN을 비교 연산자로 사용합니다. Personalize는 정확한 문자열만 필터링합니다.

비교 연산자

=, <, <=, >, >=, != 를 사용하십시오. = 자리 표시자 매개 변수로 전달된 데이터를 비롯한 수치 데이터가 같은지 테스트하는 연산자입니다.

별표(*) 문자

*는 모든 유형의 상호작용을 포함하거나 제외하는 데 사용합니다. *는 Interactions 데이터 세트의 EVENT_TYPE 필드를 사용하는 필터 표현식에만 사용합니다.

파이프 구분자

파이프 구분자(|)를 사용하여 여러 표현식을 연결할 수 있습니다. 자세한 내용은 [여러 표현식 결합 단원을 참조](#)하세요.

파라미터

비교 연산자 또는 IN 연산자를 사용하는 표현식의 경우 자리 표시자 파라미터를 값으로 추가하려면 달러 기호(\$)와 파라미터 이름을 사용합니다. 예를 들어 \$GENRES입니다. 이 예제에서는 추천을 받을 때 필터링 기준으로 삼을 장르를 입력합니다.

Note

표현식에 파라미터를 추가할 때 파라미터 이름을 정의합니다. 파라미터 이름이 필드 이름과 일치하지 않아도 됩니다. 필드 이름과 비슷하고 기억하기 쉬운 파라미터 이름을 사용하는 것이 좋습니다. 추천 요청에 필터를 적용할 때 파라미터 이름(대소문자 구분)을 사용합니다. SDK를 사용할 때 자리 표시자 매개 변수와 함께 필터를 적용하는 방법을 보여주는 예제를 참조하십시오. AWS [필터 적용 \(AWS SDK\)](#)

필터 표현식 예제

다음에 이어지는 단원의 필터 표현식을 사용하여 고유한 필터 표현식을 작성하는 방법을 알아봅니다.

주제

- [항목 추천 필터 표현식 예제](#)
- [사용자 세그먼트 필터 표현식](#)
- [작업 추천 필터 표현식 예제](#)
- [여러 표현식 결합](#)

항목 추천 필터 표현식 예제

다음 필터 표현식은 항목 상호 작용, 항목 메타데이터, 사용자 메타데이터를 기반으로 항목 추천을 필터링하는 방법을 보여줍니다. 데이터 유형별로 정리되어 있습니다.

주제

- [항목 상호 작용 데이터](#)
- [항목 데이터](#)
- [사용자 데이터](#)

항목 상호 작용 데이터

다음 표현식은 \$EVENT_TYPE 파라미터를 사용하여 추천을 받을 때 지정하는 이벤트 유형(예: 클릭)에 따라 항목을 제외합니다.

```
EXCLUDE ItemID WHERE Interactions.EVENT_TYPE IN ($EVENT_TYPE)
```

다음 표현식은 사용자가 클릭하거나 스트리밍한 항목을 제외합니다.

```
EXCLUDE ItemID WHERE Interactions.EVENT_TYPE IN ("click", "stream")
```

다음 표현식은 사용자가 클릭한 항목만 포함시킵니다.

```
INCLUDE ItemID WHERE Interactions.EVENT_TYPE IN ("click")
```

항목 데이터

다음 표현식은 \$CATEGORY 파라미터를 사용하여 추천을 받을 때 지정한 범주를 기준으로 항목을 제외합니다.

```
EXCLUDE ItemID WHERE Items.CATEGORY IN ($CATEGORY)
```

다음 표현식은 현재 항목(관련 항목 추천 요청에서 지정한 항목) 보다 저렴하고 현재 항목과 동일한 스튜디오에서 만든 항목만 포함시킵니다. 도메인 사용 사례 또는 사용자 지정 레시피가 관련 항목 추천을 생성하는 경우에만 CurrentItem 요소와 함께 필터를 적용할 수 있습니다.

```
INCLUDE ItemID WHERE Items.PRICE < CurrentItem.PRICE AND Items.GENRE IN
CurrentItem.GENRE
```

다음 표현식은 여러 수준의 범주형 필드를 기준으로 항목을 제외합니다. 이는 boot의 CATEGORY_L2 값이 없는 shoe의 CATEGORY_L1 값을 가진 항목을 제외합니다.

```
EXCLUDE ItemID WHERE Items.CATEGORY_L1 IN ("shoe") AND Items.CATEGORY_L2 NOT IN
("boot")
```

다음 표현식은 \$PRICE파라미터를 사용하여 추천을 받을 때 지정한 가격보다 낮거나 같은 가격의 항목만 포함합니다.

```
INCLUDE ItemID WHERE Items.PRICE <= $PRICE
```

다음 표현식은 추천을 받을 때 지정한 타임스탬프(Unix epoch 시간) 이전에 생성된 항목만 포함합니다.

```
INCLUDE ItemID WHERE Items.CREATION_TIMESTAMP < $DATE
```

다음 표현식은 \$GENRE파라미터를 사용하여 추천을 받을 때 지정한 장르의 항목만 포함합니다.

```
INCLUDE ItemID WHERE Items.GENRE IN ($GENRE)
```

다음 표현식은 현재 항목보다 비싸고 및 지정한 타임스탬프(Unix epoch 시간)보다 최근에 생성된 항목만 포함합니다. 관련 항목 추천을 받고 가격 및 다양한 생성 날짜를 기반으로 특정 비즈니스 규칙을 적용하려는 경우 이 필터를 사용할 수 있습니다.

```
INCLUDE ItemID WHERE Items.PRICE < CurrentItem.PRICE AND Items.CREATION_TIMESTAMP >
$DATE
```

사용자 데이터

다음 표현식은 \$GENRE파라미터를 사용하여 추천을 받을 때 지정한 장르의 항목을 제외하되, 현재 사용자의 연령이 \$AGE파라미터를 사용하여 추천을 받을 때 지정한 값과 같은 경우에만 제외합니다.

```
EXCLUDE ItemID WHERE Items.GENRE IN ($GENRE) IF CurrentUser.AGE = $AGE
```

다음 표현식은 현재 사용자의 연령이 18을 초과하는 경우 CATEGORY_L1의 경우 watch 그리고 CATEGORY_L2의 경우 luxury가 있는 항목만 포함합니다.

```
INCLUDE ItemID WHERE Items.CATEGORY_L1 IN ("watch") AND Items.CATEGORY_L2 IN ("luxury")
IF CurrentUser.AGE > 18
```

사용자 세그먼트 필터 표현식

다음 필터 표현식은 항목 상호 작용 데이터 및 사용자 메타데이터를 기반으로 사용자 세그먼트를 필터링하는 방법을 보여줍니다. 데이터 유형별로 정리되어 있습니다.

사용자 데이터

다음 필터 표현식은 사용자 세그먼트를 가져올 때 지정한 값과 동일한 멤버십 상태를 가진 사용자만 포함합니다.

```
INCLUDE UserID WHERE Users.MEMBERSHIP_STATUS IN ($MEMBERSHIP)
```

다음 필터 표현식은 사용자 세그먼트를 가져올 때 지정한 값보다 작은 AGE의 사용자를 제외합니다.

```
EXCLUDE UserID WHERE Users.AGE < $AGE
```

항목 상호 작용 데이터

다음 필터 표현식은 항목을 클릭하거나 등급을 매긴 사용자만 포함합니다.

```
INCLUDE UserID WHERE Interactions.EVENT_TYPE IN ("click", "rating")
```

다음 필터 표현식은 사용자 세그먼트를 가져올 때 지정한 이벤트 유형과 항목 상호 작용하는 사용자를 사용자 세그먼트에서 제외합니다.

```
EXCLUDE UserID WHERE Interactions.EVENT_TYPE IN ($EVENT_TYPE)
```

작업 추천 필터 표현식 예제

다음 필터 표현식 예제는 작업 상호 작용 데이터, 작업 데이터, 사용자 데이터를 기반으로 작업을 필터링하는 방법을 보여줍니다. 데이터 유형별로 정리되어 있습니다.

주제

- [작업 상호 작용 데이터](#)
- [작업 데이터](#)
- [사용자 데이터](#)

작업 상호 작용 데이터

다음 필터 표현식에는 상호 작용에 추천을 받을 때 지정한 이벤트 유형이 있는 경우 사용자가 상호 작용한 추천의 작업만 포함됩니다.

```
INCLUDE ActionID WHERE Action_Interactions.EVENT_TYPE IN ($EVENT_TYPE)
```

다음 필터 표현식에서는 이벤트 유형에 따라 사용자가 수행하지 않은 작업은 제외합니다.

```
EXCLUDE ActionID WHERE Action_Interactions.EVENT_TYPE IN ("NOT_TAKEN")
```

작업 데이터

다음 표현식은 \$CATEGORY 파라미터를 사용하여 추천을 받을 때 지정한 범주를 기준으로 작업을 제외합니다.

```
EXCLUDE ActionID WHERE Actions.CATEGORY IN ($CATEGORY)
```

다음 표현식에는 추천을 받을 때 지정한 값보다 큰 값을 가진 작업만 포함됩니다.

```
INCLUDE ActionID WHERE Actions.VALUE > ($VALUE)
```

사용자 데이터

다음 표현식에는 현재 사용자가 프리미엄 멤버십을 보유한 경우 프리미엄 회원을 위한 작업만 포함됩니다.

```
INCLUDE ActionID WHERE Action.MEMBERSHIP_LEVEL IN ("Premium") IF CurrentUser.MEMBERSHIP = $PREMIUM
```

다음 표현식에는 현재 사용자가 프리미엄 회원인 경우 추천을 받을 때 지정하는 값보다 적은 VALUE의 작업은 제외됩니다.

```
EXCLUDE ActionID WHERE Actions.VALUE < ($VALUE) IF CurrentUser.MEMBERSHIP = $PREMIUM
```

여러 표현식 결합

여러 표현식을 함께 결합하려면 파이프 구분자(|)를 사용합니다. 항목 및 항목 상호 작용 데이터 세트 또는 작업 및 작업 상호 작용 데이터 세트에 하나 이상의 필터를 사용하려면 표현식을 조합하여 사용하세요. 각 표현식이 먼저 독립적으로 평가되며 결과는 두 결과의 합집합 또는 교차가 됩니다. 다음 예제는 항목 및 항목 상호 작용 데이터 세트에 대한 표현식을 만드는 방법을 보여주지만, 작업 및 작업 상호 작용을 사용할 때도 동일한 규칙이 적용됩니다.

매칭 표현식 예제

두 표현식이 모두 EXCLUDE를 사용하거나 두 표현식이 모두 INCLUDE를 사용하는 경우 결과는 다음과 같이 두 결과를 합친 것이 됩니다(A와 B가 다른 표현식임).

- Exclude A | Exclude B은 Exclude result from A or result from B와 같습니다
- Include A | Include B은 Include result from A or result from B와 같습니다

다음 예제에서는 INCLUDE를 사용하는 두 개의 표현식을 결합하는 방법을 보여줍니다. 첫 번째 표현식은 \$CATEGORY파라미터를 사용하여 추천을 받을 때 지정한 범주의 항목만 포함합니다. 두 번째 표현식은 사용자에게 favorite로 표시된 항목을 포함합니다. 추천에는 사용자가 즐겨찾기로 표시한 항목과 함께 지정한 범주의 항목만 포함될 것입니다.

```
INCLUDE ItemID WHERE Items.CATEGORY IN ($CATEGORY) | INCLUDE ItemID WHERE
Interactions.EVENT_TYPE IN ("favorite")
```

INCLUDE 및 EXCLUDE 예제

하나 이상의 표현식이 INCLUDE를 사용하고 하나 이상의 표현식이 EXCLUDE를 사용하는 경우 결과는 다음과 같이 INCLUDE표현식 결과에서 EXCLUDE표현식 결과를 뺀 것이 됩니다(A, B, C, D가 다른 표현식임).

- Include A | Exclude B은 Include result from A - result from B와 같습니다
- Include A | Include B | Exclude C | Exclude D은 Include (A or B) - (C or D)와 같습니다

표현식 순서는 중요하지 않습니다. EXCLUDE 표현식이 INCLUDE 표현식 앞에 오면 결과는 동일합니다.

다음 예제에서는 INCLUDE표현식과 EXCLUDE표현식을 결합하는 방법을 보여줍니다. 첫 번째 표현식은 \$GENRE파라미터를 사용하여 추천을 받을 때 지정한 장르의 항목만 포함합니다. 두 번째 표현식은

사용자가 클릭하거나 스트리밍한 항목을 제외합니다. 추천에는 장르를 지정하고 클릭하거나 스트리밍한 적이 없는 항목만 포함될 것입니다.

```
INCLUDE ItemID WHERE Items.GENRE IN ($GENRE) | EXCLUDE ItemID WHERE
Interactions.EVENT_TYPE IN ("click", "stream")
```

실시간 추천 필터링

Amazon Personalize 콘솔, AWS Command Line Interface (AWS CLI) 또는 SDK를 사용하여 실시간 추천을 필터링할 수 있습니다. AWS

개인 맞춤형 항목 추천이나 유사한 항목을 받으면 요청에 홍보를 지정할 수 있습니다. 홍보는 필터를 사용하여 추천 항목의 구성 가능한 하위 집합에 적용되는 추가 비즈니스 규칙을 정의합니다. 자세한 내용은 [추천의 항목 홍보](#) 단원을 참조하세요.

주제

- [실시간 추천 필터링\(콘솔\)](#)
- [실시간 추천 필터링\(AWS CLI\)](#)
- [실시간 추천 \(AWS SDK\) 필터링](#)

실시간 추천 필터링(콘솔)

콘솔을 사용하여 실시간 추천을 필터링하려면 필터를 생성한 다음, 추천 요청에 필터를 적용합니다.

Note

파라미터가 포함된 필터와 2020년 11월 10일 이전에 배포한 캠페인을 사용하여 추천을 필터링하려면 [UpdateCampaign](#) 작업을 사용하여 캠페인을 재배포하거나 캠페인을 새로 생성해야 합니다.

필터 생성(콘솔)

콘솔에서 필터를 만들려면 필터링된 추천을 받는 데 사용할 캠페인 또는 추천자가 포함된 데이터세트 그룹을 선택합니다. 그런 다음 필터 이름과 필터 표현식을 제공합니다.

필터를 생성하려면(콘솔)

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 필터링된 추천을 받는 데 사용할 캠페인 또는 추천자가 포함된 데이터세트 그룹을 선택합니다.
3. 탐색 창에서 필터를 선택한 다음 새 필터 생성을 선택합니다. 필터 생성 페이지가 표시됩니다.

Create filter Info

Use filters to include or exclude items from Amazon Personalize recommendations. To create a filter, provide a filter name and filter expression.

Filter configuration

Filter name
The filter name that you enter here can help you distinguish this filter from others.

The filter name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and _ - (hyphen).

Expression
The expressions specify what to include or exclude from your recommendations.

Build expression
Select this option to build an expression using the expression builder tool.

Input expression
Select this option if you have an existing expression or prefer to input text.

Build expression Info
Build an expression using the fields below. For the value, enter a value (or comma separated values) to set the filter criteria, or enter "\$*" + parameter name to add a placeholder parameter. When you get recommendations, you'll pass a value to this parameter to set the filter criteria.

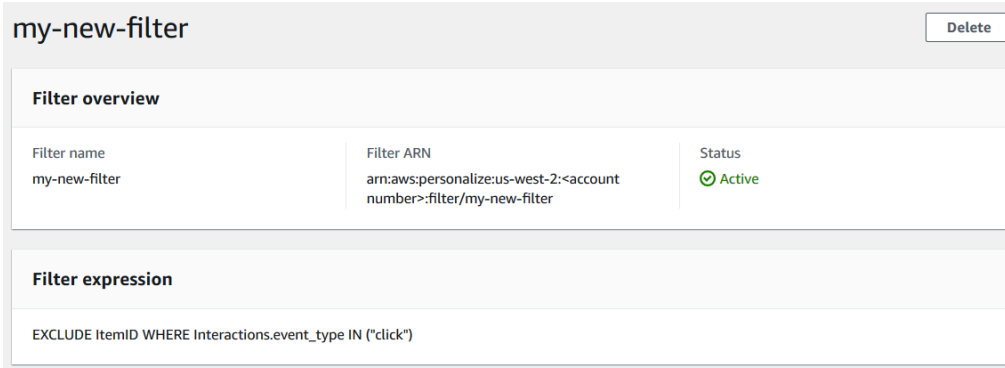
Action	ID	Property	Operator	Value
Exclude ▼	ItemID ▼	WHERE	Interactions.event_type ▼	IN ▼ Value or \$PARAMETER
				+

► **Tags - optional (0) Info**

A tag is an administrative label that you assign to AWS resources to make it easier to manage them. Each tag consists of a key and an optional value. Use tags to search and filter your resources or track your AWS costs.

4. 필터 이름에 필터 이름을 입력합니다. 추천 요청에 필터를 적용할 때 이 이름을 기준으로 하는 필터를 선택합니다.
5. 표현식에 대해 표현식 작성 또는 표현식 수동 추가를 선택하고 표현식을 작성하거나 삽입합니다.
 - 표현식 빌더를 사용하려면 표현식 작성을 선택합니다. 표현식 빌더는 올바른 형식의 필터 표현식을 작성하기 위한 구조, 필드 및 지침을 제공합니다. 자세한 내용은 [필터 표현식 빌더 사용](#) 단원을 참조하세요.

- 표현식을 직접 입력하려면 표현식 수동 추가를 선택합니다. 자세한 내용은 [필터 표현식 요소단원](#)을 참조하세요.
6. 마침을 클릭합니다. 필터의 개요 페이지에 필터의 리소스 이름(ARN), 상태 및 전체 필터 표현식이 표시됩니다. 필터를 삭제하려면 삭제를 선택합니다. 개요 페이지를 떠난 후 필터를 찾고 삭제하는 방법에 대한 자세한 내용은 [필터 삭제\(콘솔\)](#) 단원을 참조하세요.



필터 적용(콘솔)

필터를 적용하려면 추천자 테스트(추천자의 경우) 또는 캠페인 결과 테스트(사용자 지정 캠페인의 경우)에서 필터를 선택하고 필터 파라미터 값을 입력합니다. 그런 다음 사용자를 위한 추천을 받습니다.

⚠ Important

INCLUDE 요소를 사용하는 필터 표현식의 경우 표현식에 정의되어 있는 모든 파라미터의 값을 제공해야 합니다. EXCLUDE 요소를 사용하는 표현식이 포함된 필터의 경우 filter-values를 생략할 수 있습니다. 이 경우 Personalize는 표현식의 해당 부분을 추천을 필터링하는 데 사용하지 않습니다.

필터를 적용하려면(콘솔)

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 필터링된 추천을 받는 데 사용할 캠페인 또는 추천자가 포함된 데이터세트 그룹을 선택합니다.
3. 데이터 세트 그룹 유형 또는 리소스 유형에 따라 다음 중 하나를 수행합니다.
 - a. 도메인 데이터세트 그룹의 경우 탐색 창에서 추천자를 선택합니다.

- b. 사용자 지정 데이터 세트 그룹 또는 사용자 지정 리소스의 경우 탐색 창에서 사용자 지정 리소스를 선택한 다음 캠페인을 선택합니다.
4. 추천자 또는 캠페인 페이지에서 대상 추천자 또는 캠페인을 선택합니다.
5. 비교를 위해 필터를 적용하지 않고 추천을 가져오기 시작합니다. 추천자 테스트 / 캠페인 결과 테스트에서 추천을 받으려는 사용자의 ID 또는 관련 항목의 항목 ID를 입력하고 추천 받기를 선택합니다. 최상위 추천이 포함된 표가 나타납니다.

Test campaign results

User ID Info

This is the user ID of the user you want to see campaign results for. This user ID needs to be obtained from your user-interactions or user dataset.

Filter name

Select the filter you want to apply to this recommendation.

 Refresh

To find a filter, [go to the filter page](#).

Get recommendations

Item ID	Score
3948	0.0107270
1676	0.0069995
2657	0.0064348
2985	0.0055178
2081	0.0054022

6. 필터 이름에서, 생성한 필터를 선택합니다. 필터에 자리 표시자 파라미터가 있는 경우 각 파라미터의 관련 필드가 나타납니다.
7. 자리 표시자 파라미터가 있는 필터를 사용하는 경우 각 파라미터에 대해 값을 입력하여 필터 기준을 설정합니다. 한 파라미터에 여러 값을 사용하려면 각 값을 쉼표로 구분합니다.
8. 이전 단계에서와 동일한 User ID 또는 Item ID를 사용하여 추천 받기를 선택합니다. 추천 표가 나타납니다.

예를 들어, 사용자가 이미 추천 항목을 구매한 경우 필터는 추천 목록에서 해당 항목을 제거합니다. 이 예시에서는 2657, 2985 항목이 사용자가 구매하지 않은 가장 적합한 항목(2641 및 1573 항목)으로 대체되었습니다.

필터 표현식 빌더 사용

필터 만들기 페이지의 표현식 빌더는 올바른 형식의 필터를 작성하기 위한 구조, 필드 및 지침을 제공합니다.

Select this option to build an expression using the expression builder tool.

Select this option if you have an existing expression or prefer to input text.

Build expression Info
Build an expression using the fields below. For the value, enter a value (or comma separated values) to set the filter criteria, or enter "\$" + parameter name to add a placeholder parameter. When you get recommendations, you'll pass a value to this parameter to set the filter criteria.

Action	ID		Property	Operator	Value
Exclude ▼	ItemID ▼	WHERE	Interactions.event_type ▼	IN ▼	Value or \$PARAMETER
		AND ▼	Interactions.event_type ▼	IN ▼	Value or \$PARAMETER

+

Add expression

필터 표현식을 작성하려면:

- 유형, 작업, 속성, 연산자, 값 필드를 사용하여 표현식을 생성합니다.

값에 고정 값을 입력하거나, 추천을 받을 때 필터 기준을 설정하려면 \$ + 파라미터 이름을 입력합니다. 예를 들어 \$GENRES입니다. 추천을 받을 때 필터링 기준으로 사용할 값을 하나 또는 여러 개 제공합니다. 이 예시에서는 추천을 받을 때 장르 또는 장르 목록을 제공하게 될 것입니다.

파라미터가 아닌 여러 값은 쉼표로 구분합니다. 쉼표로 구분된 파라미터는 필터에 추가할 수 없습니다.

i Note

속성(dataset.field 형식)을 선택한 후에는 AND또는 OR조건으로 연결된 모든 후속 행의 속성 값도 동일한 dataset을 사용해야 합니다.

- + 및 X 버튼을 사용하여 표현식에 행을 추가하거나 삭제할 수 있습니다. 첫 번째 행은 삭제할 수 없습니다.
- 새 행의 경우 AND 메뉴에서 AND, IF또는 OR연산자를 사용하여 일련의 조건을 만들 수 있습니다.

IF 조건의 경우:

- 표현식마다 IF항목을 하나만 포함할 수 있습니다. IF 조건을 제거하면 표현식 빌더가 그 뒤에 오는 모든 AND조건을 제거합니다.
- IF 조건은 CurrentUser를 기준으로 필터링하는 표현식에만 사용할 수 있습니다.

- 표현식 추가 버튼을 선택하여 보다 정밀한 필터링을 위해 필터 표현식을 추가합니다. 먼저 각 표현식을 독립적으로 평가한 다음 두 결과를 합친 결과를 얻습니다.

Note

항목 및 항목 상호 작용 데이터 세트 또는 작업 및 작업 상호 작용 데이터 세트를 모두 사용하는 필터를 만들려면 여러 표현식을 사용해야 합니다.

표현식 빌더 예제

다음 예제에서는 추천을 받을 때 지정한 장르를 가진 항목을 제외하는 필터를 작성하는 방법을 보여줍니다(\$GENRES 자리 표시자 파라미터 참고). 또한 필터는 DOWNLOAD_COUNT이 200이상인 항목을 제외하지만 현재 사용자의 연령이 17을 초과하는 경우에만 제외합니다.

Build expression Info

Build an expression using the fields below. For the value, enter a value (or comma separated values) to set the filter criteria, or enter "\$" + parameter name to add a placeholder parameter. When you get recommendations, you'll pass a value to this parameter to set the filter criteria.

Action	Property	Operator	Value
Exclude	ItemID WHERE Items.GENRES	IN	\$GENRES
AND	Items.DOWNLOAD_COUNT	>	200
IF	currentUser.AGE	>	17

Add expression

필터 삭제(콘솔)

필터를 삭제하면 데이터세트 그룹의 필터 목록에서 해당 필터가 제거됩니다.

Important

배치 추론 작업이 진행 중인 동안에는 필터를 삭제할 수 없습니다.

필터를 삭제하려면(콘솔)

- <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
- 데이터세트 그룹 목록에서 삭제하려는 필터가 포함된 데이터세트 그룹을 선택합니다.
- 탐색 창에서 필터를 선택합니다.

4. 필터 목록에서 삭제하려는 필터를 선택하고 세부 정보 보기를 선택합니다. 필터 세부 정보 페이지가 나타납니다.
5. 삭제를 선택하고 확인 대화 상자에서 삭제를 확정합니다.

실시간 추천 필터링(AWS CLI)

를 사용하여 권장 사항을 필터링하려면 필터를 만든 다음 [GetRecommendations](#) or [GetPersonalizedRanking](#) 요청에서 필터 ARN을 지정하여 필터를 적용합니다. AWS CLI

⚠ Important

파라미터가 포함된 필터와 2020년 11월 10일 이전에 배포한 캠페인을 사용하여 추천을 필터링하려면 [UpdateCampaign](#) 호출을 사용하여 캠페인을 다시 배포하거나 새 캠페인을 생성해야 합니다.

필터 생성(AWS CLI)

다음과 같은 `create-filter` 작업을 사용하여 필터를 만들고 필터 표현식을 지정합니다.

`Filter name`을 필터 이름으로 바꾸고 `Dataset group ARN`를 데이터세트 그룹의 리소스 이름 (ARN)으로 바꿉니다. 샘플 `filter-expression`을 자체 필터 표현식으로 바꿉니다.

```
aws personalize create-filter \
  --name Filter name \
  --dataset-group-arn dataset group arn \
  --filter-expression "EXCLUDE ItemID WHERE Items.CATEGORY IN (\\"$CATEGORY\\")"
```

성공하면 필터 ARN이 표시됩니다. 나중에 사용할 수 있도록 기록해 둡니다. 필터가 활성 상태인지 확인하려면 [DescribeFilter](#) 작업을 사용한 후에 필터를 사용합니다.

API에 대한 자세한 내용은 [CreateFilter](#) 단원을 참조하세요. 필터 표현식에 대한 자세한 내용은 [필터 표현식 구조 및 요소](#) 단원을 참조하세요.

필터 적용(AWS CLI)

`get-recommendations`, `get-action-recommendations` 또는 `get-personalized-ranking` 작업을 사용하는 경우, `filter-arn` 및 모든 필터 값을 파라미터로 전달하여 필터를 적용합니다.

다음은 `get-recommendations` 작업의 예입니다. Campaign ARN을 캠페인의 리소스 이름(ARN)으로, User ID를 추천을 받는 사용자의 ID로 그리고 Filter ARN을 필터의 ARN으로 각각 바꿉니다. 캠페인 대신 추천자로부터 추천을 받는 경우 `--campaign-arn` 대신 `recommender-arn`을 사용하고 추천자의 ARN을 제공합니다.

표현식에 파라미터가 있는 경우 `filter-values` 객체를 포함시킵니다. 필터 표현식의 각 파라미터에 대해 파라미터 이름(대소문자 구분)과 값을 입력합니다. 예를 들어 필터 표현식에 `$GENRE` 파라미터가 있는 경우 "GENRE"를 키로 입력하고 값으로 장르(예: "Comedy")를 입력합니다. 값이 여러 개인 경우 쉼표로 구분합니다. 예를 들어 `"\"comedy\"\", \"drama\"\", \"horror\""`입니다.

⚠ Important

INCLUDE 요소를 사용하여 항목을 포함하는 필터 표현식의 경우 표현식에 정의되어 있는 모든 파라미터의 값을 제공해야 합니다. EXCLUDE 요소를 사용하여 항목을 제외하는 표현식이 포함된 필터의 경우 `filter-values`를 생략할 수 있습니다. 이 경우 Personalize는 표현식의 해당 부분을 추천을 필터링하는 데 사용하지 않습니다.

```
aws personalize-runtime get-recommendations \
  --campaign-arn Campaign ARN \
  --user-id User ID \
  --filter-arn Filter ARN \
  --filter-values '{
    "Parameter name": "\"value\\"",
    "Parameter name": "\"value1\", \"value2\", \"value3\\""
  }'
```

필터 삭제(AWS CLI)

필터를 삭제하려면 다음과 같은 `delete-filter` 작업을 사용합니다. filter ARN을 필터의 ARN으로 대체합니다.

```
aws personalize delete-filter --filter-arn Filter ARN
```

실시간 추천 (AWS SDK) 필터링

AWS SDK를 사용하여 권장 사항을 필터링하려면 필터를 만든 다음 or 요청에서 [GetRecommendations](#) 필터 ARN을 지정하여 필터를 적용합니다. [GetPersonalizedRanking](#)

⚠ Important

파라미터가 포함된 필터와 2020년 11월 10일 이전에 배포한 캠페인을 사용하여 추천을 필터링하려면 [UpdateCampaign](#) 호출을 사용하여 캠페인을 다시 배포하거나 새 캠페인을 생성해야 합니다.

필터 (AWS SDK) 생성

[CreateFilter](#) 작업을 사용하여 새 필터를 생성합니다. 다음 코드는 필터를 만드는 방법을 보여줍니다. 데이터세트 그룹의 필터 이름, 리소스 이름(ARN)을 지정하고 필터 표현식을 제공합니다.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_filter(
    name = 'Filter Name',
    datasetGroupArn = 'Dataset Group ARN',
    filterExpression = 'EXCLUDE ItemID WHERE Items.CATEGORY IN ($CATEGORY)'
)
filter_arn = response["filterArn"]
print("Filter ARN: " + filter_arn)
```

SDK for Java 2.x

```
public static String createFilter(PersonalizeClient personalizeClient,
                                String filterName,
                                String datasetGroupArn,
                                String filterExpression) {

    try {
        CreateFilterRequest request = CreateFilterRequest.builder()
            .name(filterName)
            .datasetGroupArn(datasetGroupArn)
            .filterExpression(filterExpression)
            .build();

        return personalizeClient.createFilter(request).filterArn();
    }
    catch(PersonalizeException e) {
```



```

        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}

```

SDK for JavaScript v3

```

// Get service clients module and commands using ES6 syntax.
import { CreateFilterCommand } from
    "@aws-sdk/client-personalize";
import { personalizeClient } from "./libs/personalizeClients.js";
// Or, create the client here.
// const personalizeClient = new PersonalizeClient({ region: "REGION"});

// Set the filter's parameters.
export const createFilterParam = {
    datasetGroupArn: 'DATASET_GROUP_ARN', /* required */
    name: 'NAME', /* required */
    filterExpression: 'FILTER_EXPRESSION' /*required */
}

export const run = async () => {
    try {
        const response = await personalizeClient.send(new
        CreateFilterCommand(createFilterParam));
        console.log("Success", response);
        return response; // For unit tests.
    } catch (err) {
        console.log("Error", err);
    }
};
run();

```

나중에 사용할 수 있도록 필터 ARN을 기록해 놓습니다. 필터가 활성 상태인지 확인하려면 [DescribeFilter](#) 작업을 사용한 후 필터를 사용합니다. API에 대한 자세한 내용은 [CreateFilter](#) 단원을 참조하세요. 필터 표현식에 대한 자세한 내용은 [필터 표현식 구조 및 요소](#) 단원을 참조하세요.

필터 적용 (AWS SDK)

GetRecommendations GetActionRecommendations, 또는 GetPersonalizedRanking 작업을 사용하는 경우 a filterArn 및 모든 필터 값을 매개 변수로 전달하여 필터를 적용합니다.

다음 코드는 사용자에게 대한 필터링된 Amazon Personalize 항목 추천을 가져오는 방법을 보여줍니다. 추천을 받고자 하는 사용자의 ID, 캠페인의 리소스 이름(ARN), 필터의 ARN을 지정합니다. 캠페인 대신 추천자로부터 추천을 받는 경우 `campaignArn` 대신 `recommenderArn`을 사용하고 추천자의 ARN을 제공합니다.

`filterValues`의 경우, 필터 표현식의 각 선택적 파라미터에 대해 파라미터 이름(대소문자 구분)과 값을 입력합니다. 예를 들어 필터 표현식에 `$GENRES` 파라미터가 있는 경우 "GENRES"를 키로 입력하고 값으로 장르(예: `"Comedy"`)를 입력합니다. 값이 여러 개인 경우 각 값을 쉼표로 구분합니다. 예를 들어 `"comedy", "drama", "horror"`입니다.

Important

INCLUDE 요소를 사용하여 항목을 포함하는 필터 표현식의 경우 표현식에 정의되어 있는 모든 파라미터의 값을 제공해야 합니다. EXCLUDE 요소를 사용하여 항목을 제외하는 표현식이 포함된 필터의 경우 `filter-values`를 생략할 수 있습니다. 이 경우 Personalize는 표현식의 해당 부분을 추천을 필터링하는 데 사용하지 않습니다.

SDK for Python (Boto3)

```
import boto3

personalize_runtime = boto3.client("personalize-runtime")

response = personalize_runtime.get_recommendations(
    campaignArn = "Campaign ARN",
    userId = "User ID",
    filterArn = "Filter ARN",
    filterValues = {
        "Parameter name": "\"value1\"",
        "Parameter name": "\"value1\", \"value2\", \"value3\""
        ....
    }
)
```

SDK for Java 2.x

다음 예제에서는 두 개의 값이 있는 파라미터와 값이 한 개인 파라미터를 사용합니다. 필터 표현식에 따라 코드를 수정하여 `ParameterName` 및 `ParameterValue` 필드를 추가하거나 제거합니다.

```
public static void getFilteredRecs(PersonalizeRuntimeClient
personalizeRuntimeClient,
                                String campaignArn,
                                String userId,
                                String filterArn,
                                String parameter1Name,
                                String parameter1Value1,
                                String parameter1Value2,
                                String parameter2Name,
                                String parameter2Value){

    try {

        Map<String, String> filterValues = new HashMap<>();

        filterValues.put(parameter1Name, String.format("\"%1$s\", \"%2$s\"",
            parameter1Value1, parameter1Value2));
        filterValues.put(parameter2Name, String.format("\"%1$s\"",
            parameter2Value));

        GetRecommendationsRequest recommendationsRequest =
        GetRecommendationsRequest.builder()
            .campaignArn(campaignArn)
            .numResults(20)
            .userId(userId)
            .filterArn(filterArn)
            .filterValues(filterValues)
            .build();

        GetRecommendationsResponse recommendationsResponse =
        personalizeRuntimeClient.getRecommendations(recommendationsRequest);
        List<PredictedItem> items = recommendationsResponse.itemList();

        for (PredictedItem item: items) {
            System.out.println("Item Id is : "+item.itemId());
            System.out.println("Item score is : "+item.score());
        }
    } catch (PersonalizeRuntimeException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

SDK for JavaScript v3

```
// Get service clients module and commands using ES6 syntax.
import { GetRecommendationsCommand } from
  "@aws-sdk/client-personalize-runtime";
import { personalizeRuntimeClient } from "./libs/personalizeClients.js";
// Or, create the client here:
// const personalizeRuntimeClient = new PersonalizeRuntimeClient({ region:
  "REGION"});

// Set recommendation request parameters.
export const getRecommendationsParam = {
  campaignArn: 'CAMPAIGN_ARN', /* required */
  userId: 'USER_ID',          /* required */
  numResults: 15,             /* optional */
  filterArn: 'FILTER_ARN',    /* required to filter recommendations */
  filterValues: {
    "PROPERTY": "\"VALUE\"" /* Only required if your filter has a placeholder
parameter */
  }
}

export const run = async () => {
  try {
    const response = await personalizeRuntimeClient.send(new
GetRecommendationsCommand(getRecommendationsParam));
    console.log("Success!", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

필터 삭제 (AWS Python SDK)

필터를 삭제하려면 다음과 같은 `delete_filter` 메서드를 사용합니다. `filter ARN`을 필터의 ARN으로 대체합니다.

```
import boto3
personalize = boto3.client("personalize")
```

```
response = personalize.delete_filter(
    filterArn = "filter ARN"
)
```

배치 추천 및 사용자 세그먼트(사용자 지정 리소스) 필터링

배치 추천 및 사용자 세그먼트 필터링은 실시간 추천 필터링과 거의 동일하게 작동합니다. [배치 권장 사항 및 사용자 세그먼트\(사용자 지정 리소스\)](#)에 설명되어 있는 것과 동일한 워크플로우를 따릅니다. 배치 추천 또는 사용자 세그먼트를 필터링하려면 다음과 같이 합니다.

1. 실시간 추천과 마찬가지로 필터를 만듭니다. 자세한 내용은 [실시간 추천 필터링](#) 단원을 참조하세요.
2. [배치 추천을 위한 입력 데이터 준비](#) 또는 [사용자 세그먼트를 위한 입력 데이터 준비](#)에 설명된 대로 입력 데이터를 준비하고 S3에 업로드합니다. 필터가 자리 표시자 파라미터를 사용하는 경우 `filterValues` 객체를 추가해야 합니다. 자세한 내용은 [입력 JSON에서 필터 값 제공](#) 단원을 참조하세요. 필터가 자리 표시자 파라미터를 사용하지 않는 경우 입력 데이터는 [배치 추론 작업 입력 및 출력 JSON 예제](#) 배치 세그먼트 작업 입력 및 출력 JSON 예제의 다음 예제를 따를 수 있습니다.
3. 출력 데이터를 저장할 별도의 위치(폴더 또는 다른 S3 버킷)를 만듭니다.
4. [배치 추론 작업](#) 또는 [배치 세그먼트 작업](#)을 만듭니다. 작업을 만들 때 필터의 리소스 이름(ARN)을 지정합니다.
5. 배치 추론 또는 배치 세그먼트 작업이 완료되면 S3의 출력 위치에서 추천 또는 사용자 세그먼트를 검색합니다.

주제

- [입력 JSON에서 필터 값 제공](#)
- [배치 워크플로우 필터링\(콘솔\)](#)
- [배치 워크플로우 필터링\(AWS SDK\)](#)

입력 JSON에서 필터 값 제공

자리 표시자 파라미터(예: \$GENRE)가 있는 필터의 경우 입력 JSON의 `filterValues` 객체에서 파라미터 값을 제공해야 합니다. `filterValues` 객체의 경우 각 키가 파라미터 이름에 해당합니다. 각 값은 파라미터로 전달하는 기준에 해당합니다. 각 값을 이스케이프 처리된 따옴표("filterValues": {"GENRES": "\"drama\""})로 묶습니다. 값이 여러 개인 경우 각 값을 쉼표("filterValues": {"GENRES": "\"horror\"", "\"comedy\"", "\"drama\""})로 구분합니다.

배치 추론 작업 입력 JSON 예제

다음은 배치 추론 작업에 대한 JSON 입력 파일의 처음 몇 줄의 예입니다. 예제에는 `filterValues` 객체가 포함되어 있습니다. GENRES 키는 필터 표현식의 \$GENRES 자리 표시자에 해당합니다. 이 예제의 작업에서는 사용자-개인 맞춤 레시피를 사용합니다. RELATED_ITEMS 레시피의 경우 `userID` 대신 `ItemID`를 제공합니다. PERSONALIZED_RANKING 레시피의 경우 `userID`와 `itemList`를 제공합니다.

```
{ "userId": "5", "filterValues": { "GENRES": "\"horror\", \"comedy\", \"drama\"" } }
{ "userId": "3", "filterValues": { "GENRES": "\"horror\", \"comedy\"" } }
{ "userId": "34", "filterValues": { "GENRES": "\"drama\"" } }
```

레시피별 배치 추론 작업 입력 데이터에 대한 자세한 예는 [배치 추론 작업 입력 및 출력 JSON 예제](#) 단원을 참조하세요. 이들 예제를 시작점으로 사용하여 위 예제의 `filterValues` 객체를 추가할 수 있습니다.

배치 세그먼트 작업 입력 JSON 예제

다음은 배치 세그먼트 작업에 대한 필터 값이 있는 JSON 입력 파일의 처음 몇 줄의 예입니다. GENRES 키는 필터 표현식의 \$GENRES 자리 표시자에 해당합니다.

```
{ "itemAttributes": "ITEMS.genres = \"Comedy\" AND ITEMS.genres = \"Action\"", "filterValues": { "COUNTRY": "\"Japan\"" } }
{ "itemAttributes": "ITEMS.genres = \"Horror\"", "filterValues": { "COUNTRY": "\"United States\"" } }
{ "itemAttributes": "ITEMS.genres = \"Action\" AND ITEMS.genres = \"Adventure\"", "filterValues": { "COUNTRY": "\"England\"" } }
```

레시피별 배치 추론 작업 입력 데이터에 대한 자세한 예는 [배치 세그먼트 작업 입력 및 출력 JSON 예제](#) 단원을 참조하세요. 이들 예제를 시작점으로 사용하여 위 예제의 `filterValues` 객체를 추가할 수 있습니다.

배치 워크플로우 필터링(콘솔)

Personalize 콘솔에서 배치 워크플로우를 필터링하려면 필터를 생성한 다음 배치 추론 작업 또는 배치 세그먼트 작업을 생성하고 필터를 선택합니다. 단계별 지침은 [배치 추론 작업 생성\(콘솔\)](#) 및 [배치 세그먼트 작업 생성\(콘솔\)](#) 단원을 참조하세요.

배치 워크플로우 필터링(AWS SDK)

AWS SDK로 일괄 추천을 필터링하려면 필터를 만들고 [CreateBatchInferenceJob](#) or [CreateBatchSegmentJob](#) 요청에 `FilterArn` 파라미터를 포함하세요.

다음 코드는 AWS SDK for Python (Boto3)을 사용하여 필터가 포함된 배치 추론 작업을 만드는 방법을 보여줍니다. 출력 데이터에 대한 다른 위치(폴더 또는 다른 S3 버킷)를 사용하는 것이 좋습니다. 모든 필드에 대한 자세한 설명은 [일괄 추론 작업 \(AWS SDK\) 생성단원](#)을 참조하세요.

```
import boto3

personalize = boto3.client("personalize")

personalize_rec.create_batch_inference_job (
    solutionVersionArn = "Solution version ARN",
    jobName = "Batch job name",
    roleArn = "IAM role ARN",
    filterArn = "Filter ARN",
    jobInput =
        {"s3DataSource": {"path": "S3 input path"}}},
    jobOutput =
        {"S3DataDestination": {"path": "S3 output path"}}
)
```

추천의 영향 측정

고객이 추천에 반응하면서 추천이 목표 달성에 얼마나 도움이 되는지 측정할 수 있습니다. 가장 많은 시청 시간 또는 가장 많은 클릭을 발생시키는 리소스 등 어떤 캠페인과 추천이 가장 큰 영향을 미치는지 파악할 수 있습니다. 또한 Amazon Personalize 추천의 성능을 타사 서비스에서 생성된 추천과 비교할 수 있습니다.

다음은 추천의 영향을 측정하는 데 도움이 될 수 있습니다.

- [지표 어트리뷰션](#): Amazon Personalize 지표 어트리뷰션은 사용자가 지정한 지표와 가져온 항목 상호 작용 및 항목 데이터를 기반으로 보고서를 생성합니다. 예를 들어, 사용자가 본 영화의 총 길이 또는 총 클릭 이벤트 수가 여기에 해당합니다.
- [A/B 테스트](#): A/B 테스트 수행은 여러 변형이 포함된 실험을 실행하고 결과를 비교하는 것으로 구성됩니다. A/B 테스트를 사용하면 다양한 추천 전략을 비교 및 평가하고 추천의 영향을 측정할 수 있습니다.

주제

- [지표 어트리뷰션을 통한 추천 영향력 측정](#)
- [A/B 테스트를 통한 권장 사항 영향 측정](#)

지표 어트리뷰션을 통한 추천 영향력 측정

항목 추천의 영향을 측정하기 위해 지표 어트리뷰션을 만들 수 있습니다. 지표 어트리뷰션은 가져온 항목 상호 작용 및 항목 데이터, 지정한 지표를 기반으로 보고서를 생성합니다. 예를 들어, 사용자가 본 영화의 총 길이 또는 총 클릭 이벤트 수가 여기에 해당합니다. Personalize는 15분 동안 계산을 집계합니다. 증분 대량 데이터의 경우 PutEvents Amazon Personalize는 자동으로 아마존에 지표 보고서를 전송합니다. CloudWatch 대량 데이터의 경우 S3 버킷에 보고서를 게시하도록 선택할 수 있습니다.

가져오는 각 상호작용에 대해 소스 데이터를 포함시켜서 다양한 캠페인, 추천자 및 타사와 비교합니다. 사용자에게 보여준 추천의 추천 ID나 이벤트 소스(예: 타사)를 포함시킬 수 있습니다.

예를 들어, 서로 다른 두 Personalize 추천자의 영화 추천을 보여주는 동영상 스트리밍 앱이 있을 수 있습니다. 어떤 추천자가 가장 많은 시청 이벤트를 생성하는지 확인하려면 총 시청 이벤트를 추적하는 지표 어트리뷰션을 만들 수 있습니다. 그런 다음 사용자가 추천과 상호작용하는 동안 시청 이벤트를 기록하고 각 이벤트에 recommendationId를 포함시킬 수 있습니다. Personalize는 recommendationId를 사용하여 각 추천자를 식별합니다. 이벤트를 기록할 때 양쪽 추천인에 대해

15분마다 집계된 시청 이벤트 총계를 볼 수 있습니다. CloudWatch 이벤트에 `recommendationId` 또는 `eventAttributionSource`을 포함시키는 방법을 보여주는 코드 샘플은 [이벤트 지표 및 어트리뷰션 보고서](#) 단원을 참조하세요.

주제

- [지침 및 요구 사항](#)
- [지표 어트리뷰션 생성](#)
- [지표 어트리뷰션 관리](#)
- [결과 게시 및 보기](#)

지침 및 요구 사항

Personalize는 지표 어트리뷰션을 생성한 후에야 추천의 영향을 계산하고 보고하기 시작합니다. 가장 완벽한 이력을 작성하려면 상호작용 데이터를 가져오기 전에 지표 어트리뷰션을 생성하는 것이 좋습니다. Amazon Personalize 콘솔을 사용하여 항목 상호 작용 데이터 세트에 대한 데이터 세트 가져오기 작업을 생성하는 경우, 옵션을 통해 새 탭에서 지표 어트리뷰션을 만들 수 있습니다. 그런 다음 가져오기 작업으로 돌아가 작업을 완료할 수 있습니다.

지표 어트리뷰션을 생성하고 이벤트를 기록하거나 증분 대량 데이터를 가져온 후에는 지표당 월별 비용이 발생합니다. CloudWatch CloudWatch 요금에 대한 자세한 내용은 [Amazon CloudWatch 가격 책정](#) 페이지를 참조하십시오. 메트릭 전송을 CloudWatch 중단하려면 지표 [어트리뷰션을 삭제하십시오](#).

시간 경과에 따른 추천의 영향을 확인하려면 고객이 추천과 상호작용하는 방식에 따른 데이터를 계속 가져옵니다. 데이터를 이미 가져왔더라도 지표 어트리뷰션을 생성하고 추천이 미치는 영향을 측정할 수 있습니다. 하지만 Personalize는 데이터를 생성하기 전에 가져온 데이터에 대해서는 보고하지 않습니다.

다음은 지표 어트리뷰션이 포함된 보고서를 생성하기 위한 지침 및 요구 사항입니다.

- Amazon Personalize에 데이터에 액세스하고 데이터를 입력할 수 있는 권한을 부여해야 합니다. CloudWatch 정책 예제는 [Amazon Personalize에 대한 액세스 권한 부여 CloudWatch](#) 단원을 참조하세요.
- S3에 지표를 게시하려면 Personalize에 버킷에 쓸 수 있는 권한을 부여합니다. 또한 지표 어트리뷰션에 버킷 경로를 제공해야 합니다. 정책 예제는 [Personalize에 S3 버킷에 대한 액세스 권한 부여](#) 단원을 참조하세요.
- 지표를 CloudWatch 게시하려면 레코드가 14일 미만이어야 합니다. 데이터가 이 보다 오래된 경우 이러한 레코드는 계산이나 보고서에 포함되지 않습니다.

- 중복된 이벤트(모든 속성이 정확히 일치하는 이벤트)를 가져오면 부정확한 지표를 비롯하여 예상치 못한 동작이 발생할 수 있습니다. 가져오기 전에 모든 대량 데이터에서 중복된 레코드를 제거하고 PutEvents 작업을 사용하여 중복 이벤트를 가져오지 않는 것이 좋습니다.
- 항목 상호 작용 데이터 세트에 EVENT_TYPE 열이 있어야 합니다.
- 작업 상호 작용 데이터 세트의 데이터에 대해서는 지표 보고서를 만들 수 없습니다.
- 데이터세트 그룹당 최대 하나의 지표 어트리뷰션을 만들 수 있습니다. 각 지표 어트리뷰션에는 최대 10개의 지표가 있을 수 있습니다.

소스를 비교하려면 각 상호작용 이벤트에 recommendationId 또는 eventAttributionSource가 포함되어 있어야 합니다. 최대 100개의 고유한 이벤트 어트리뷰션 소스를 제공할 수 있습니다.

PutEvents 코드 샘플은 [이벤트 지표 및 어트리뷰션 보고서](#) 단원을 참조하세요.

- recommendationId를 입력하면 Personalize는 자동으로 소스 캠페인 또는 추천자를 결정하고 EVENT_ATTRIBUTION_SOURCE 열의 보고서에서 이를 식별합니다.
- 두 속성을 모두 제공하는 경우 Personalize는 eventAttributionSource만 사용합니다.
- 소스를 제공하지 않는 경우 Personalize는 보고서에서 소스 SOURCE_NAME_UNDEFINED에 레이블을 지정합니다.

주제

- [Amazon Personalize에 대한 액세스 권한 부여 CloudWatch](#)
- [Personalize에 S3 버킷에 대한 액세스 권한 부여](#)

Amazon Personalize에 대한 액세스 권한 부여 CloudWatch

Important

권한을 부여하면 Amazon Personalize는 소량의 데이터를 배치하고 유효성을 검사합니다. CloudWatch 이모인해 0.30달러 미만의 일회성 비용이 발생합니다. CloudWatch 요금에 대한 자세한 내용은 [Amazon CloudWatch 가격 책정](#) 페이지를 참조하십시오.

Amazon Personalize에 대한 액세스 권한을 부여하려면 CloudWatch Amazon Personalize 서비스 역할에 작업을 사용할 수 있는 권한을 부여하는 새 AWS Identity and Access Management (IAM) 정책을 추가하십시오. PutMetricData CloudWatch 다음 예시와 같은 정책은 PutMetricData 권한을 부여합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Resource": "*"
    }
  ]
}
```

Personalize에 S3 버킷에 대한 액세스 권한 부여

Personalize에 S3 버킷에 대한 액세스 권한을 부여하려면:

- Personalize 서비스 역할에 IAM 정책을 연결하여 역할에 버킷에서 PutObject 작업을 사용할 수 있는 권한을 부여합니다.

```
{
  "Version": "2012-10-17",
  "Id": "PersonalizeS3BucketAccessPolicy",
  "Statement": [
    {
      "Sid": "PersonalizeS3BucketAccessPolicy",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}
```

- Personalize 원칙에 PutObject 작업을 사용할 수 있는 권한을 부여하는 버킷 정책을 출력 S3 버킷에 연결합니다.

암호화에 AWS Key Management Service (AWS KMS) 를 사용하는 경우 Amazon Personalize 와 Amazon Personalize IAM 서비스 역할에 키를 사용할 권한을 부여해야 합니다. 자세한 정보는 [Amazon Personalize에 AWS KMS 키 사용 권한 부여](#)을 참조하세요.

```
{
  "Version": "2012-10-17",
  "Id": "PersonalizeS3BucketAccessPolicy",
  "Statement": [
    {
      "Sid": "PersonalizeS3BucketAccessPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "personalize.amazonaws.com"
      },
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}
```

지표 어트리뷰션 생성

Important

지표 속성을 생성하고 이벤트를 기록하거나 증분 대량 데이터를 가져온 후에는 지표당 월별 CloudWatch 비용이 발생합니다. CloudWatch 요금에 대한 자세한 내용은 [Amazon CloudWatch 가격 책정](#) 페이지를 참조하십시오. 메트릭 전송을 CloudWatch 중단하려면 [지표 어트리뷰션을 삭제하십시오](#).

지표 보고서 생성을 시작하려면 지표 어트리뷰션을 생성하고 상호작용 데이터를 가져옵니다. 지표 어트리뷰션을 만들 때는 보고할 이벤트 유형 목록을 지정합니다. 각 이벤트 유형에 대해 Personalize가 데이터를 수집할 때 적용하는 함수를 지정합니다. 사용 가능한 함수에는 SUM(DatasetType.COLUMN_NAME) 및 SAMPLECOUNT()이 포함됩니다.

예를 들어 온라인 동영상 스트리밍 앱에서 추천 클릭률과 총 시청 영화 길이(항목 데이터세트의 각 동영상에 LENGTH속성이 포함된 경우)라는 두 가지 지표를 추적하고 싶을 수 있습니다. 지표 어트리뷰션을 만들고 각각 이벤트 유형과 함수가 지표 두 개를 추가할 수 있습니다. 첫 번째는 SAMPLECOUNT() 함수가 있는 Click이벤트 유형에 대한 것일 수 있습니다. 두 번째는 SUM(Items.LENGTH) 함수가 있는 Watch이벤트 유형에 대한 것일 수 있습니다.

SUM() 함수는 항목 및 항목 상호 작용 데이터 세트의 숫자 열에만 적용할 수 있습니다. SUM() 함수를 항목 데이터세트의 열에 적용하려면 먼저 항목 메타데이터를 가져와야 합니다.

Amazon Personalize 콘솔 AWS Command Line Interface 또는 SDKs를 사용하여 지표 어트리뷰션을 생성할 수 있습니다. AWS

주제

- [지표 어트리뷰션 생성\(콘솔\)](#)
- [지표 어트리뷰션 생성\(AWS CLI\)](#)
- [지표 어트리뷰션 \(AWS SDK\) 생성](#)

지표 어트리뷰션 생성(콘솔)

Personalize 콘솔에서 지표 어트리뷰션을 생성하려면 지표 어트리뷰션 페이지로 이동하여 지표 어트리뷰션 생성을 선택합니다. 지표 어트리뷰션을 생성할 때 선택적 S3 버킷 경로, Personalize IAM 서비스 역할 및 보고할 지표 목록을 지정합니다.

Amazon Personalize 콘솔에서 항목 상호 작용 데이터 세트 가져오기 작업을 생성하는 경우 옵션을 통해 새 탭에서 지표 어트리뷰션을 만들 수 있습니다. 그런 다음 가져오기 작업으로 돌아가 작업을 완료할 수 있습니다. 이미 지표 어트리뷰션 구성 페이지에 있는 경우 4단계로 건너뛰어도 됩니다.

지표 어트리뷰션을 생성하려면

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터세트 그룹을 선택합니다.
3. 탐색 창의 사용자 지정 리소스에서 지표 어트리뷰션을 선택합니다.
4. 지표 어트리뷰션 세부 정보에서 지표 어트리뷰션 생성을 선택합니다.
5. 지표 어트리뷰션 구성 페이지에서 지표 어트리뷰션 이름을 지정합니다.

6. S3에 지표를 게시하려면, S3 데이터 출력 경로에 대상 S3 버킷을 입력합니다. 이렇게 하면 데이터 세트 가져오기 작업을 생성할 때마다 지표를 게시할 수 있는 옵션이 활성화됩니다. 다음 구문을 사용합니다.

s3://<name of your S3 bucket>/<folder> path>

7. AWS KMS 암호화에 사용하는 경우 KMS 키 ARN에 키의 Amazon 리소스 이름 (ARN) 을 입력합니다. AWS KMS 사용자가 Personalize와 Personalize IAM 서비스 역할에 키를 사용할 권한을 부여해야만 합니다. 자세한 내용은 [Amazon Personalize에 AWS KMS 키 사용 권한 부여](#) 단원을 참조하세요.
8. IAM 역할에서 새 서비스 역할 생성 또는 기존 역할 사용 중 하나를 선택합니다. 선택한 역할에는 권한이 있어야 PutMetricData 합니다. CloudWatch S3에 게시하려는 경우 역할에 S3 버킷에 대한 PutObject권한이 있어야 합니다.

에서 [Personalize에 대한 IAM 역할 생성](#) 생성한 역할을 사용하려면 Amazon S3에 대한 CloudWatch 정책을 추가해야 할 수 있습니다.

정책 예제는 [Amazon Personalize에 대한 액세스 권한 부여 CloudWatch](#) 및 [Personalize에 S3 버킷에 대한 액세스 권한 부여](#) 단원을 참조하세요.

9. 다음을 선택합니다.
10. 지표 어트리뷰션 정의 페이지에서 지표 정의 방법을 선택합니다. 빌더 도구를 사용하려면 지표 어트리뷰션 작성을 선택합니다. 지표 어트리뷰션 입력을 선택하여 지표를 JSON 형식으로 입력합니다.
 - 지표 어트리뷰션 작성을 선택하는 경우 각 지표에 대해 이름, 이벤트 유형을 제공하고 함수를 선택합니다. SUM() 함수의 경우 열 이름을 선택합니다. 지표를 더 추가하려면 지표 어트리뷰션 추가를 선택합니다.
 - 지표 어트리뷰션 입력을 선택한 경우 각 지표를 JSON 형식으로 입력합니다. 아래에서는 지표 형식을 지정하는 방법을 보여줍니다.

```
{
  "EventType": "watch",
  "MetricName": "MinutesWatchedTracker",
  "MetricMathExpression": "SUM(Items.LENGTH)"
}
```

11. 다음을 선택합니다.
12. 검토 및 생성 페이지에서 새 지표 어트리뷰션에 대한 세부 정보를 검토합니다. 변경하려면 이전을 선택합니다. 지표 어트리뷰션을 만들려면 생성을 선택합니다. 지표 어트리뷰션이 활성화되면 데

이더 가져오기를 시작하고 그 결과를 볼 수 있습니다. 결과 보기에 대한 자세한 내용은 [결과 게시 및 보기](#) 단원을 참조하세요.

지표 어트리뷰션 생성(AWS CLI)

다음 코드는 AWS Command Line Interface에서 지표 어트리뷰션을 생성하는 방법을 보여줍니다. 지정하는 역할에는 Amazon S3 버킷에 대한 PutMetricData 권한이 있어야 CloudWatch 하며, Amazon S3에 게시하는 경우 Amazon S3 버킷에 대한 PutObject 권한이 있어야 합니다. 에서 [Personalize에 대한 IAM 역할 생성](#) 생성한 역할을 사용하려면 Amazon S3에 대한 CloudWatch 정책을 추가해야 할 수 있습니다. 정책 예제는 [Amazon Personalize에 대한 액세스 권한 부여 CloudWatch](#) 및 [Personalize에 S3 버킷에 대한 액세스 권한 부여](#) 단원을 참조하세요.

각 지표에 대해 이름, 이벤트 유형 및 표현식(함수)을 지정합니다. 사용 가능한 함수에는 SUM(DatasetType.COLUMN_NAME) 및 SAMPLECOUNT()이 포함됩니다. SUM() 함수의 경우 데이터 세트 유형과 열 이름을 지정합니다. 예를 들어 SUM(Items.LENGTH)입니다. 각 파라미터에 대한 자세한 내용은 [CreateMetricAttribution](#) 단원을 참조하세요.

```
aws personalize create-metric-attribution \
--name metric attribution name \
--dataset-group-arn dataset group arn \
--metrics-output-config "{\"roleArn\": \"Amazon Personalize service role ARN\",
  \"s3DataDestination\": {\"kmsKeyArn\": \"kms key ARN\", \"path\": \"s3://bucket-
  name/folder-name/\"}}\" \
--metrics "[{
  \"eventType\": \"event type\",
  \"expression\": \"SUM(DatasetType.COLUMN_NAME)\",
  \"metricName\": \"metric name\"
}]"
```

지표 어트리뷰션 (AWS SDK) 생성

다음 코드는 Python용 SDK(Boto3)에서 지표 어트리뷰션을 생성하는 방법을 보여줍니다. 지정하는 역할에는 Amazon S3 버킷에 대한 PutMetricData 권한이 있어야 CloudWatch 하며, Amazon S3에 게시하는 경우 Amazon S3 버킷에 대한 PutObject 권한이 있어야 합니다. 에서 [Personalize에 대한 IAM 역할 생성](#) 생성한 역할을 사용하려면 Amazon S3에 대한 CloudWatch 정책을 추가해야 할 수 있습니다. 정책 예제는 [Amazon Personalize에 대한 액세스 권한 부여 CloudWatch](#) 및 [Personalize에 S3 버킷에 대한 액세스 권한 부여](#) 단원을 참조하세요.

각 지표에 대해 이름, 이벤트 유형 및 표현식(함수)을 지정합니다. 사용 가능한 함수에는 SUM(DatasetType.COLUMN_NAME) 및 SAMPLECOUNT()이 포함됩니다. SUM() 함수의 경우 데이터

세트 유형과 열 이름을 지정합니다. 예를 들어 `SUM(Items.LENGTH)`입니다. 각 파라미터에 대한 자세한 내용은 [CreateMetricAttribution](#) 단원을 참조하세요.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

metricsList = [{
    "eventType": "event type",
    "expression": "SUM(DatasetType.COLUMN_NAME)",
    "metricName": "metric name"
}]

outputConfig = {
    "roleArn": "Amazon Personalize service role ARN",
    "s3DataDestination": {
        "kmsKeyArn": "key ARN",
        "path": "s3://<name of your S3 bucket>/<folder>"
    }
}

response = personalize.create_metric_attribution(
    name = 'metric attribution name',
    datasetGroupArn = 'dataset group arn',
    metricsOutputConfig = outputConfig,
    metrics = metricsList
)

metric_attribution_arn = response['metricAttributionArn']

print ('Metric attribution ARN: ' + metric_attribution_arn)

description = personalize.describe_metric_attribution(
    metricAttributionArn = metric_attribution_arn)['metricAttribution']

print('Name: ' + description['name'])
print('ARN: ' + description['metricAttributionArn'])
print('Status: ' + description['status'])
```

SDK for Java 2.x

```
public static String createMetricAttribution(PersonalizeClient personalizeClient,
```



```
String eventType,  
String expression,  
String metricName,  
String metricAttributionName,  
String roleArn,  
String s3Path,  
String kmsKeyArn,  
String datasetGroupArn) {  
  
String metricAttributionArn = "";  
  
try {  
  
    MetricAttribute attribute = MetricAttribute.builder()  
        .eventType(eventType)  
        .expression(expression)  
        .metricName(metricName)  
        .build();  
  
    ArrayList<MetricAttribute> metricAttributes = new ArrayList<>();  
    metricAttributes.add(attribute);  
  
    S3DataConfig s3DataDestination = S3DataConfig.builder()  
        .kmsKeyArn(kmsKeyArn)  
        .path(s3Path)  
        .build();  
  
    MetricAttributionOutput outputConfig = MetricAttributionOutput.builder()  
        .roleArn(roleArn)  
        .s3DataDestination(s3DataDestination)  
        .build();  
  
    CreateMetricAttributionRequest createMetricAttributionRequest =  
    CreateMetricAttributionRequest.builder()  
        .name(metricAttributionName)  
        .datasetGroupArn(datasetGroupArn)  
        .metrics(metricAttributes)  
        .metricsOutputConfig(outputConfig)  
        .build();  
  
    CreateMetricAttributionResponse createMetricAttributionResponse =  
    personalizeClient.createMetricAttribution(createMetricAttributionRequest);  
  
    metricAttributionArn =  
    createMetricAttributionResponse.metricAttributionArn();  
    System.out.println("Metric attribution ARN: " + metricAttributionArn);  
}
```

```

        return metricAttributionArn;
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}

```

SDK for JavaScript v3

```

// Get service clients and commands using ES6 syntax.
import { CreateMetricAttributionCommand, PersonalizeClient } from
    "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
    region: "REGION"
});

// set the metric attribution param
export const createMetricAttributionParam = {
    name: "METRIC_ATTRIBUTION_NAME",          /* required */
    datasetGroupArn: "DATASET_GROUP_ARN",    /* required */
    metricsOutputConfig: {
        roleArn: "ROLE_ARN",                 /* required */
        s3DataDestination: {
            kmsKeyArn: "KEY_ARN",            /*
optional */
            path: "s3://<name of your output S3 bucket>/<folderName>/", /* optional */
        },
    },
    metrics: [
        {
            eventType: "EVENT_TYPE",          /* required for each metric */
            expression: "SUM(DatasetType.COLUMN_NAME)", /* required for each metric */
            metricName: "METRIC_NAME",        /* required for each metric */
        }
    ]
};

export const run = async () => {
    try {
        const response = await personalizeClient.send(
            new CreateMetricAttributionCommand(createMetricAttributionParam)

```

```
);  
console.log("Success", response);  
return response; // For unit tests.  
} catch (err) {  
  console.log("Error", err);  
}  
};  
run();
```

지표 어트리뷰션 관리

지표 어트리뷰션을 만든 후 업데이트하거나 삭제할 수 있습니다. 지표 어트리뷰션을 삭제하면 Amazon Personalize는 증분 가져오기와 관련된 보고서 전송 및 증분 PutEvents 가져오기를 중단합니다.

CloudWatch

주제

- [지표 어트리뷰션 업데이트](#)
- [지표 어트리뷰션 삭제](#)

지표 어트리뷰션 업데이트

지표 어트리뷰션을 업데이트할 때 지표를 추가 및 제거하고 출력 구성을 수정할 수 있습니다. Amazon Personalize 콘솔 AWS Command Line Interface 또는 SDKs를 사용하여 지표 어트리뷰션을 업데이트할 수 있습니다. AWS

주제

- [지표 어트리뷰션 업데이트\(콘솔\)](#)
- [지표 어트리뷰션 업데이트\(AWS CLI\)](#)
- [지표 어트리뷰션 \(AWS SDK\) 업데이트](#)

지표 어트리뷰션 업데이트(콘솔)

Personalize 콘솔에서 지표 어트리뷰션을 업데이트하려면 지표 어트리뷰션 페이지에서 변경합니다.

지표 어트리뷰션을 업데이트하려면

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터세트 그룹을 선택합니다.
3. 탐색 창에서 지표 어트리뷰션을 선택합니다.
4. 하단 섹션에서 지표 어트리뷰션 탭 또는 지표 어트리뷰션 구성 탭을 선택하여 변경을 시작합니다.
 - 지표를 추가하거나 제거하려면 지표 어트리뷰션 탭을 선택하고 속성 편집을 선택합니다. 지표 어트리뷰션 편집 페이지에서 변경한 다음 업데이트를 선택하여 변경 사항을 저장합니다.
 - S3 출력 버킷 또는 IAM 서비스 역할을 변경하려면 지표 어트리뷰션 구성 편집 탭을 선택하고 어트리뷰션 구성 편집 페이지에서 변경합니다. 업데이트를 선택하여 변경 사항을 저장합니다.

지표 어트리뷰션 업데이트(AWS CLI)

메트릭 어트리뷰션을 생성한 후 AWS Command Line Interface (AWS CLI) 를 사용하여 지표를 추가 및 제거하고 출력 구성을 수정할 수 있습니다. 다음 코드는 `update-metric-attribution` 명령으로 지표를 제거하는 방법을 보여줍니다.

```
aws personalize update-metric-attribution \
--metric-attribution-arn metric attribution arn \
--remove-metrics metricName1 metricName2
```

다음 코드는 지표를 추가하고 새 출력 구성을 지정하는 방법을 보여줍니다.

```
aws personalize update-metric-attribution \
--metric-attribution-arn metric attribution arn \
--metrics-output-config "{\"roleArn\": \"new role ARN\", \"s3DataDestination\":  
{\"kmsKeyArn\": \"kms key ARN\", \"path\": \"s3://new-bucket-name/new-folder-name/\"}}\" \
--add-metrics "[{  
  \"eventType\": \"event type\",  
  \"expression\": \"SUM(DatasetType.COLUMN_NAME)\",  
  \"metricName\": \"metric name\"  
}]"
```

Personalize는 사용자가 업데이트한 지표 어트리뷰션을 기준으로 ARN을 반환합니다. 파라미터의 전체 목록은 [UpdateMetricAttribution](#) 단원을 참조하세요.

지표 어트리뷰션 (AWS SDK) 업데이트

지표 어트리뷰션을 생성한 후 지표를 추가 또는 제거하고 출력 구성을 수정할 수 있습니다. 다음 코드는 지표 어트리뷰션에서 지표를 제거하는 방법을 보여줍니다.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

metricsToRemove = ["metricName1", "metricName2"]

response = personalize.update_metric_attribution(
    metricAttributionArn = "metric attribution ARN",
    removeMetrics = metricsToRemove
)
```

SDK for Java 2.x

```
public static void removeMetrics(PersonalizeClient client,
                                String metricAttributionArn,
                                String metric1Name,
                                String metric2Name) {

    ArrayList<String> metricsToRemove = new ArrayList<>(Arrays.asList(metric1Name,
metric2Name));

    try {

        UpdateMetricAttributionRequest request =
UpdateMetricAttributionRequest.builder()
            .metricAttributionArn(metricAttributionArn)
            .removeMetrics(metricsToRemove)
            .build();

        UpdateMetricAttributionResponse response =
client.updateMetricAttribution(request);
        System.out.println(response);

    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
}
```

```
}
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import {UpdateMetricAttributionCommand, PersonalizeClient } from
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
  region: "REGION"
});

// set the update request param
export const updateMetricAttributionParam = {
  metricAttributionArn: "METRIC_ATTRIBUTION_ARN",    /* required */
  removeMetrics: ["METRIC_NAME_1", "METRIC_NAME_2"] /* specify list of names of
  metrics to delete */
};
export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new UpdateMetricAttributionCommand(updateMetricAttributionParam)
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

다음 코드는 지표를 추가하고 새 출력 구성을 지정하는 방법을 보여줍니다.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

newMetrics = [{
    "eventType": "event type",
```

```

        "expression": "SUM(DatasetType.COLUMN_NAME)",
        "metricName": "metric name"
    }]

    newOutputConfig = {
        "roleArn": "Amazon Personalize service role ARN",
        "s3DataDestination": {
            "kmsKeyArn": "key ARN",
            "path": "s3://<name of your S3 bucket>/<folder>"
        }
    }

    response = personalize.update_metric_attribution(
        metricAttributionArn = "metric attribution arn",
        metricsOutputConfig = newOutputConfig,
        addMetrics = newMetrics
    )

```

SDK for Java 2.x

```

public static void addMetricsAndUpdateOutputConfig(PersonalizeClient
    personalizeClient,

                                                    String metricAttributionArn,
                                                    String newMetric1EventType,
                                                    String newMetric1Expression,
                                                    String newMetric1Name,
                                                    String newMetric2EventType,
                                                    String newMetric2Expression,
                                                    String newMetric2Name,
                                                    String roleArn,
                                                    String s3Path,
                                                    String kmsKeyArn) {

    try {

        MetricAttribute newAttribute = MetricAttribute.builder()
            .eventType(newMetric1EventType)
            .expression(newMetric1Expression)
            .metricName(newMetric1Name)
            .build();

        MetricAttribute newAttribute2 = MetricAttribute.builder()
            .eventType(newMetric2EventType)
            .expression(newMetric2Expression)

```

```

        .metricName(newMetric2Name)
        .build();

    ArrayList<MetricAttribute> newAttributes = new
    ArrayList<>(Arrays.asList(newAttribute, newAttribute2));

    S3DataConfig newDataDestination = S3DataConfig.builder()
        .kmsKeyArn(kmsKeyArn)
        .path(s3Path)
        .build();

    MetricAttributionOutput newOutputConfig = MetricAttributionOutput.builder()
        .roleArn(roleArn)
        .s3DataDestination(newDataDestination)
        .build();

    UpdateMetricAttributionRequest request =
    UpdateMetricAttributionRequest.builder()
        .metricAttributionArn(metricAttributionArn)
        .metricsOutputConfig(newOutputConfig)
        .addMetrics(newAttributes)
        .build();

    UpdateMetricAttributionResponse response =
    personalizeClient.updateMetricAttribution(request);
    System.out.println("New metrics added!");
    System.out.println(response);

    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
}

```

SDK for JavaScript v3

```

// Get service clients and commands using ES6 syntax.
import {UpdateMetricAttributionCommand, PersonalizeClient } from
"@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
    region: "REGION"
});

```



```

export const updateMetricAttributionParam = {
  metricAttributionArn: "METRIC_ATTRIBUTION_ARN",
  addMetrics: [
    {
      eventType: "EVENT_TYPE", /* required for each metric */
      expression: "SUM(DatasetType.COLUMN_NAME)", /* required for each metric */
      metricName: "METRIC_NAME", /* required for each metric */
    }
  ],
  metricsOutputConfig: {
    roleArn: "ROLE_ARN", /* required */
    s3DataDestination: {
      kmsKeyArn: "KEY_ARN", /*
optional */
      path: "s3://<name of your output S3 bucket>/<folderName>/", /* optional */
    },
  }
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new UpdateMetricAttributionCommand(updateMetricAttributionParam)
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};

run();

```

Personalize는 사용자가 업데이트한 지표 어트리뷰션을 기준으로 ARN을 반환합니다. 파라미터의 전체 목록은 [UpdateMetricAttribution](#) 단원을 참조하세요.

지표 어트리뷰션 삭제

더 이상 보고서를 생성하지 않으려는 경우 지표 어트리뷰션을 삭제하면 됩니다. 지표 어트리뷰션을 삭제하면 해당 지표와 출력 구성이 모두 삭제됩니다.

지표 속성을 삭제하면 Amazon Personalize는 대량 데이터와 관련된 PutEvents 보고서 및 증분 대량 데이터에 대한 보고서 자동 전송을 중단합니다. CloudWatch Amazon S3로 이미 CloudWatch 전송되었거나 Amazon S3에 게시된 데이터는 영향을 받지 않습니다. Amazon Personalize 콘솔 AWS Command Line Interface 또는 SDKs를 사용하여 지표 어트리뷰션을 삭제할 수 있습니다. AWS

주제

- [지표 어트리뷰션 삭제\(콘솔\)](#)
- [지표 어트리뷰션 삭제\(AWS CLI\)](#)
- [지표 어트리뷰션 \(AWS SDK\) 삭제](#)

지표 어트리뷰션 삭제(콘솔)

지표 어트리뷰션의 개요 페이지에서 지표 어트리뷰션을 삭제합니다.

지표 어트리뷰션을 삭제하려면

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터세트 그룹을 선택합니다.
3. 탐색 창에서 지표 어트리뷰션을 선택합니다.
4. 삭제를 선택하고 삭제를 확인합니다.

지표 어트리뷰션 삭제(AWS CLI)

를 사용하여 지표 속성을 삭제하려면 다음과 같이 AWS CLI 명령을 사용합니다. `delete-metric-attribution`

```
aws personalize delete-metric-attribution --metric-attribution-arn metric attribution ARN
```

지표 어트리뷰션 (AWS SDK) 삭제

다음 코드는 Python용 SDK(Boto3)에서 지표 어트리뷰션을 삭제하는 방법을 보여줍니다.

SDK for Python (Boto3)

```
import boto3
```

```
personalize = boto3.client('personalize')

response = personalize.delete_metric_attribution(
    metricAttributionArn = 'metric attribution ARN'
)
```

SDK for Java 2.x

```
public static void deleteMetricAttribution(PersonalizeClient client, String
metricAttributionArn) {

    try {

        DeleteMetricAttributionRequest request =
DeleteMetricAttributionRequest.builder()
            .metricAttributionArn(metricAttributionArn)
            .build();

        DeleteMetricAttributionResponse response =
client.deleteMetricAttribution(request);
        if (response.sdkHttpResponse().statusCode() == 200) {
            System.out.println("Metric attribution deleted!");
        }

    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
}
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { DeleteMetricAttributionCommand, PersonalizeClient } from
"@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
    region: "REGION"
});

export const deleteMetricAttributionParam = {
    metricAttributionArn: "METRIC_ATTRIBUTION_ARN",
};
```

```

export const run = async () => {
  try {
    const response = await personalizeClient.send(
      new DeleteMetricAttributionCommand(deleteMetricAttributionParam)
    );
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();

```

결과 게시 및 보기

Amazon Personalize는 각 지표에 대한 보고서를 아마존 S3에 CloudWatch 전송합니다.

- PutEvents 데이터 및 증분 대량 데이터의 경우 Amazon Personalize는 자동으로 메트릭을 에 전송합니다. CloudWatch 에서 CloudWatch 보고서를 보고 식별하는 방법에 대한 자세한 내용은 을 참조하십시오. [에서 측정항목 보기 CloudWatch](#)
- 모든 대량 데이터의 경우, 지표 어트리뷰션을 생성할 때 S3 버킷을 제공하면 상호작용 데이터에 대한 데이터세트 가져오기 작업을 생성할 때마다 S3 버킷에 지표 보고서를 게시하도록 선택할 수 있습니다.

지표 보고서를 S3에 게시하는 방법에 대한 자세한 내용은 [S3에 지표 게시](#) 단원을 참조하세요.

주제

- [에서 측정항목 보기 CloudWatch](#)
- [S3에 지표 게시](#)

에서 측정항목 보기 CloudWatch

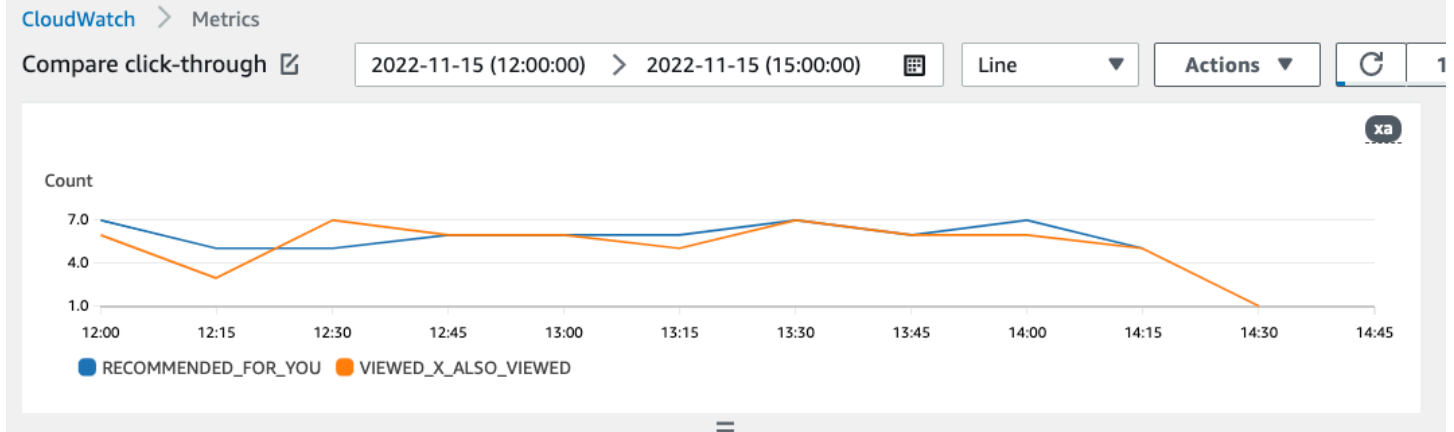
Important

지표 어트리뷰션을 생성하고 이벤트를 기록하거나 증분 대량 데이터를 가져온 후에는 지표 당 월별 CloudWatch 비용이 발생합니다. CloudWatch 요금에 대한 자세한 내용은 [Amazon](#)

[CloudWatch 가격 책정](#) 페이지를 참조하십시오. 메트릭 전송을 CloudWatch 중단하려면 지표 어트리뷰션을 삭제하십시오.

에서 CloudWatch 지표를 보려면 지표 [그래프](#) 작성에 있는 절차를 완료하세요. 그래프로 표시할 수 있는 최소 기간은 15분입니다. 검색어에는 지표 어트리뷰션을 만들 때 지표에 지정한 이름을 지정합니다.

다음은 지표가 표시되는 CloudWatch 방식의 예입니다. 이 지표는 서로 다른 두 추천자에 대한 15분 마다의 클릭률을 보여줍니다.



S3에 지표 게시

S3에 지표를 게시하려면 지표 어트리뷰션에 S3 버킷의 경로를 제공합니다. 그런 다음 데이터세트 가져오기 작업을 생성할 때 S3에 보고서를 게시합니다.

작업이 완료되면 S3 버킷에서 지표를 찾을 수 있습니다. 지표를 게시할 때마다 Personalize는 S3 버킷에 새 파일을 생성합니다. 파일 이름에는 다음과 같은 가져오기 방법과 날짜가 포함됩니다.

AggregatedAttributionMetrics - **ImportMethod** - **Timestamp.csv**

다음은 지표 보고서 CSV 파일의 처음 몇 행이 나타나는 방식의 예입니다. 이 예제의 지표는 15분 간격으로 서로 다른 두 추천자의 총 클릭 수를 보고합니다. 각 추천자는 EVENT_ATTRIBUTION_SOURCE 열의 리소스 이름(ARN)을 기준으로 식별할 수 있습니다.

```
METRIC_NAME,EVENT_TYPE,VALUE,MATH_FUNCTION,EVENT_ATTRIBUTION_SOURCE,TIMESTAMP
COUNTWATCHES,WATCH,12.0,samplecount,arn:aws:personalize:us-west-2:acctNum:recommender/recommender1Name,1666925124
COUNTWATCHES,WATCH,112.0,samplecount,arn:aws:personalize:us-west-2:acctNum:recommender/recommender2Name,1666924224
COUNTWATCHES,WATCH,10.0,samplecount,arn:aws:personalize:us-west-2:acctNum:recommender/recommender1Name,1666924224
```

```
COUNTWATCHES, WATCH, 254.0, samplecount, arn:aws:personalize:us-west-2:acctNum:recommender/
recommender2Name, 1666922424
COUNTWATCHES, WATCH, 112.0, samplecount, arn:aws:personalize:us-west-2:acctNum:recommender/
recommender1Name, 1666922424
COUNTWATCHES, WATCH, 100.0, samplecount, arn:aws:personalize:us-west-2:acctNum:recommender/
recommender2Name, 1666922424
.....
.....
```

S3에 대량 데이터에 대한 지표 게시(콘솔)

Personalize 콘솔을 사용하여 S3 버킷에 지표를 게시하려면 데이터세트 가져오기 작업을 생성하고 S3에 이벤트 지표 게시에서 이 가져오기 작업에 대한 지표 게시를 선택합니다.

자세한 지침은 을 참조하십시오. [step-by-step 대량 레코드 가져오기\(콘솔\)](#)

S3에 대량 데이터에 대한 지표 게시(AWS CLI)

AWS Command Line Interface (AWS CLI) 를 사용하여 Amazon S3 버킷에 지표를 게시하려면 다음 코드를 사용하여 데이터세트 가져오기 작업을 만들고 `publishAttributionMetricsToS3` 플래그를 제공하십시오. 특정 작업에 대한 지표를 게시하지 않으려는 경우 플래그를 생략합니다. 각 파라미터에 대한 자세한 내용은 [CreateDatasetImportJob](#) 단원을 참조하세요.

```
aws personalize create-dataset-import-job \
--job-name dataset import job name \
--dataset-arn dataset arn \
--data-source dataLocation=s3://bucketname/filename \
--role-arn roleArn \
--import-mode INCREMENTAL \
--publish-attribution-metrics-to-s3
```

Amazon S3 (AWS SDK) 에 대량 데이터에 대한 지표 게시

AWS SDK를 사용하여 Amazon S3 버킷에 지표를 게시하려면 데이터세트 가져오기 작업을 생성하고 `publishAttributionMetricsToS3 true`로 설정합니다. 각 파라미터에 대한 자세한 내용은 [CreateDatasetImportJob](#) 단원을 참조하세요.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')
```

```

response = personalize.create_dataset_import_job(
    jobName = 'YourImportJob',
    datasetArn = 'dataset_arn',
    dataSource = {'dataLocation':'s3://bucket/file.csv'},
    roleArn = 'role_arn',
    importMode = 'INCREMENTAL',
    publishAttributionMetricsToS3 = True
)

dsij_arn = response['datasetImportJobArn']

print ('Dataset Import Job arn: ' + dsij_arn)

description = personalize.describe_dataset_import_job(
    datasetImportJobArn = dsij_arn)['datasetImportJob']

print('Name: ' + description['jobName'])
print('ARN: ' + description['datasetImportJobArn'])
print('Status: ' + description['status'])

```

SDK for Java 2.x

```

public static String createPersonalizeDatasetImportJob(PersonalizeClient
personalizeClient,

                                                    String jobName,
                                                    String datasetArn,
                                                    String s3BucketPath,
                                                    String roleArn,
                                                    ImportMode importMode,
                                                    boolean publishToS3) {

    long waitInMilliseconds = 60 * 1000;
    String status;
    String datasetImportJobArn;

    try {
        DataSource importDataSource = DataSource.builder()
            .dataLocation(s3BucketPath)
            .build();

        CreateDatasetImportJobRequest createDatasetImportJobRequest =
        CreateDatasetImportJobRequest.builder()

```

```
        .datasetArn(datasetArn)
        .dataSource(importDataSource)
        .jobName(jobName)
        .roleArn(roleArn)
        .importMode(importMode)
        .publishAttributionMetricsToS3(publishToS3)
        .build();

    datasetImportJobArn =
personalizeClient.createDatasetImportJob(createDatasetImportJobRequest)
        .datasetImportJobArn();

    DescribeDatasetImportJobRequest describeDatasetImportJobRequest =
DescribeDatasetImportJobRequest.builder()
        .datasetImportJobArn(datasetImportJobArn)
        .build();

    long maxTime = Instant.now().getEpochSecond() + 3 * 60 * 60;

    while (Instant.now().getEpochSecond() < maxTime) {

        DatasetImportJob datasetImportJob = personalizeClient
            .describeDatasetImportJob(describeDatasetImportJobRequest)
            .datasetImportJob();

        status = datasetImportJob.status();
        System.out.println("Dataset import job status: " + status);

        if (status.equals("ACTIVE") || status.equals("CREATE FAILED")) {
            break;
        }
        try {
            Thread.sleep(waitInMilliseconds);
        } catch (InterruptedException e) {
            System.out.println(e.getMessage());
        }
    }
    return datasetImportJobArn;

} catch (PersonalizeException e) {
    System.out.println(e.awsErrorDetails().errorMessage());
}
return "";
```



```
}
```

SDK for JavaScript v3

```
// Get service clients and commands using ES6 syntax.
import { CreateDatasetImportJobCommand, PersonalizeClient } from
  "@aws-sdk/client-personalize";

// create personalizeClient
const personalizeClient = new PersonalizeClient({
  region: "REGION"
});

// Set the dataset import job parameters.
export const datasetImportJobParam = {
  datasetArn: 'DATASET_ARN', /* required */
  dataSource: {
    dataLocation: 's3://<name of your S3 bucket>/<folderName>/<CSVfilename>.csv' /*
required */
  },
  jobName: 'NAME', /* required */
  roleArn: 'ROLE_ARN', /* required */
  importMode: "FULL", /* optional, default is FULL */
  publishAttributionMetricsToS3: true /* set to true to publish metrics to
Amazon S3 bucket */
};

export const run = async () => {
  try {
    const response = await personalizeClient.send(new
CreateDatasetImportJobCommand(datasetImportJobParam));
    console.log("Success", response);
    return response; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

A/B 테스트를 통한 권장 사항 영향 측정

A/B 테스트 수행은 여러 변형이 포함된 실험을 실행하고 결과를 비교하는 것으로 구성됩니다. Amazon Personalize 권장 사항을 사용하여 A/B 테스트를 수행하려면 여러 사용자 그룹에 서로 다른 유형의 권장 사항을 표시한 다음 결과를 비교해야 합니다. A/B 테스트를 사용하면 다양한 권장 사항 전략을 비교 및 평가하고 권장 사항의 영향을 측정할 수 있습니다.

예를 들어, A/B 테스트를 사용하여 Amazon Personalize 권장 사항이 클릭률을 높이는지 확인할 수 있습니다. 이 시나리오를 테스트하기 위해 추천 제품과 같이 개인화되지 않은 사용자 추천을 한 그룹에 표시할 수 있습니다. 또한 Amazon Personalize에서 생성한 다른 그룹의 맞춤형 추천을 표시할 수도 있습니다. 고객이 품목과 상호작용하면서 결과를 기록하고 어떤 전략이 가장 높은 클릭률을 기록하는지 확인할 수 있습니다.

Amazon Personalize 권장 사항을 사용하여 A/B 테스트를 수행하는 워크플로는 다음과 같습니다.

1. 실험 계획 – 정량화 가능한 가설을 정의하고, 비즈니스 목표를 확인하고, 실험 변형을 정의하고, 실험 일정을 결정합니다.
2. 사용자 나누기 – 사용자를 대조군과 하나 이상의 실험군으로 구성된 둘 이상의 그룹으로 나눕니다.
3. 실험 실행 실험군의 사용자에게 수정된 권장 사항을 보여줍니다. 대조군의 사용자에게 권장사항을 변경 내용이 없는 상태로 보여주세요. 권장 사항과의 상호작용을 기록하여 결과를 추적하세요.
4. 결과 평가 – 실험 결과를 분석하여 수정으로 인해 실험군에 통계적으로 유의한 차이가 발생했는지 확인합니다.

Amazon CloudWatch Evidently 를 사용하면 Amazon Personalize 권장 사항에 따라 A/B 테스트를 수행할 수 있습니다. CloudWatch Evidently를 사용하면 실험을 정의하고, 핵심 성과 지표(KPI)를 추적하고, 권장 사항 요청 트래픽을 관련 Amazon Personalize 리소스로 라우팅하고, 실험 결과를 평가할 수 있습니다. 자세한 내용은 [CloudWatch를 사용한 A/B 테스트](#) 단원을 참조하세요.

주제

- [A/B 테스트 모범 사례](#)
- [CloudWatch를 사용한 A/B 테스트](#)

A/B 테스트 모범 사례

다음 모범 사례를 사용하면 Amazon Personalize 권장 사항에 대한 A/B 테스트를 설계하고 유지 관리할 수 있습니다.

- 정량화할 수 있는 비즈니스 목표를 확인하세요. 비교하려는 서로 다른 권장 사항이 모두 이 비즈니스 목표와 일치하는지, 그리고 다르거나 정량화할 수 없는 목표와 관련이 없는지 확인하세요.
- 비즈니스 목표에 부합하는 정량화 가능한 가설을 정의하세요. 예를 들어, 맞춤 제작 콘텐츠를 홍보하면 해당 항목에서 20% 이상 클릭이 늘어날 것이라고 예측할 수 있습니다. 가설에 따라 실험군에 적용할 수정이 결정됩니다.
- 가설과 관련된 해당 핵심 성과 지표(KPI) 를 정의하세요. KPI를 사용하여 실험 결과를 측정합니다. 지표는 다음과 같을 수 있습니다.
 - 클릭률
 - 시청 시간
 - 총 가격
- 실험의 총 사용자 수가 가설에 따라 통계적으로 유의한 결과에 도달하기에 충분히 큰지 확인하세요.
- 실험을 시작하기 전에 트래픽 분할 전략을 정의하세요. 실험 진행 중에는 트래픽 분할을 변경하지 마세요.
- 실험과 관련된 수정 사항(예: 모델)을 제외하고, 실험군과 대조군 모두에서 애플리케이션 또는 웹 사이트의 사용자 경험을 동일하게 유지하세요. UI나 지연 시간과 같은 사용자 경험의 변화가 잘못된 결과를 초래할 수 있습니다.
- 휴일, 진행 중인 마케팅 캠페인, 브라우저 제한과 같은 외부 요인을 관리하세요. 이러한 외부 요인으로 인해 잘못된 결과가 나올 수 있습니다.
- 가설 또는 비즈니스 요구 사항과 직접 관련이 없는 한 Amazon Personalize 권장 사항을 변경하지 마세요. 필터 적용 또는 수동으로 주문 변경 등을 통한 변경은 잘못된 결과를 초래할 수 있습니다.
- 결과를 평가할 때는 결론을 내리기 전에 결과가 통계적으로 유의한지 확인하세요. 업계 표준은 5% 유의 수준입니다. 통계적 유의성에 대한 자세한 내용은 [통계적 유의성에 대한 재교육](#) 단원을 참조하세요.

CloudWatch를 사용한 A/B 테스트

추천자를 생성하거나 캠페인과 함께 사용자 지정 솔루션 버전을 배포한 후에는 Amazon Personalize 권장 사항 및 Amazon CloudWatch Evidently 를 사용하여 A/B 테스트를 수행할 수 있습니다. 다음 동영상은 CloudWatch Evidently를 사용하여 Amazon Personalize 권장 사항에 따라 A/B 테스트를 수행하는 프로세스에 대해 설명합니다. 단계별 지침은 [CloudWatch를 사용하여 A/B 테스트 수행](#) 단원을 참조하세요.

[Amazon Personalize와 CloudWatch Evidently를 사용하여 AB 테스트를 수행합니다](#)

주제

- [CloudWatch를 사용하여 A/B 테스트 수행](#)
- [샘플 구현](#)

CloudWatch를 사용하여 A/B 테스트 수행

Amazon Personalize와 Amazon CloudWatch Evidently를 사용하여 A/B 테스트를 수행하려면 CloudWatch Evidently 프로젝트를 생성하고, 특성과 그 변형을 정의하고, 실험을 지원하도록 애플리케이션을 업데이트하고, 실험을 생성 및 실행하세요. 실험이 실행되는 동안 CloudWatch Evidently에서 결과를 확인할 수 있습니다.

Amazon Personalize와 CloudWatch Evidently를 사용하여 AB 테스트 수행하려면,

1. CloudWatch Evidently 프로젝트를 생성합니다. 프로젝트는 CloudWatch 리소스의 논리적 그룹입니다. 프로젝트 내에서 테스트하거나 출시하고 싶은 변형이 있는 특성을 생성합니다. 단계별 지침은 Amazon CloudWatch 사용 설명서의 [새 프로젝트 생성](#) 단원을 참조하세요.
2. 프로젝트에 특성을 추가하고 해당 변형을 정의하세요. 이 실험의 경우, 특성은 테스트하려는 추천 시나리오(예: 클릭률)를 나타내야 합니다.

특성을 추가할 때는 식별자를 지정하여 시나리오의 다양한 변형을 Amazon Personalize 추천자 또는 사용자 지정 캠페인에 매핑하세요. 각 변형의 경우, 변형 유형(예: 문자열)을 지정하고 변형에 이름을 지정한 다음 값을 지정합니다.

실험이 실행되면 애플리케이션은 변형 값을 사용하여 권장 사항에 사용할 Amazon Personalize 리소스를 결정합니다. 예를 들어, VIDEO_ON_DEMAND 추천인 두 개를 테스트하는 경우(하나는 사용 사례에 가장 적합한 추천 제품을 위해 생성되고 다른 하나는 Trending Now 사용 사례용으로 생성됨), 다음 JSON을 각 변형의 값으로 설정할 수 있습니다.

```
{"type":"top-picks-recommendations","arn":"arn:aws:personalize:us-west-2:<acct-id>:recommender/top-picks-recommender"}
```

```
{"type":"trending-recommendations","arn":"arn:aws:personalize:us-west-2:<acct-id>:recommender/trending-now-recommender"}
```

애플리케이션에서 관련 리소스를 확인하기 위해 사용할 수 있다면 어떤 식별자도 지정할 수 있습니다. 예를 들어, 추천자 또는 캠페인의 이름만 지정하고 애플리케이션에서 리소스의 Amazon Resource Name(ARN)을 구성할 수 있습니다.

특성을 추가하기 위한 단계별 지침은 Amazon CloudWatch 사용 설명서의 [프로젝트에 특성 추가](#)를 참조하십시오.

3. 애플리케이션을 업데이트하여 실험을 지원하세요.

- 특성 평가 — CloudWatch Evidently EvaluateFeature API 작업을 사용하여 각 사용자 세션에 변형을 할당합니다. EvaluateFeature 반응에는 이전 단계에서 지정한 변동 값이 포함됩니다. 이 경우에는 추천자 유형의 JSON 객체이고 추천자의 ARN입니다. 추천 요청 코드를 업데이트하여 이 리소스에서 추천을 받으세요.

특성 평가에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 [EvaluateFeature 사용](#) 단원을 참조하세요.

- 결과 기록 — 애플리케이션에 코드를 추가하여 권장 사항에 대한 사용자의 상호작용 결과를 추적합니다.

CloudWatch Evidently에서 실험에 대한 지표를 추적하려면 CloudWatch Evidently PutProjectEvents API 작업을 사용하여 각 사용자의 결과를 기록하세요. 예를 들어, 실험에서 사용자가 추천 항목을 클릭하면 이 이벤트의 세부 정보를 CloudWatch Evidently에 분명히 보낼 수 있습니다.

CloudWatch Evidently로 이벤트를 전송하는 방법에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 [PutProjectEvents 사용](#) 단원을 참조하세요.

Amazon Personalize 권장 사항의 관련성을 개선하기 위해 Amazon Personalize PutEvents API 작업을 사용하여 결과 이벤트를 기록할 수 있습니다. 도메인 사용 사례 또는 사용자 지정 레시피가 권장 사항에 대한 실시간 업데이트를 지원하는 경우, Amazon Personalize는 사용자가 애플리케이션을 사용할 때 사용자의 최신 활동을 학습하고 권장 사항을 업데이트할 수 있습니다. 업데이트를 지원하지 않는 경우, Amazon Personalize는 다음에 모델을 완전히 재학습할 때 이 데이터를 사용하여 권장 사항에 영향을 줍니다.

Amazon Personalize로 이벤트를 스트리밍하는 것에 대한 자세한 내용은 [이벤트 기록](#)을 참조하세요.

4. 실험을 생성하고 시작합니다. 규칙을 생성할 때 다음을 지정할 수 있습니다.

- 특성 — 실험에서 테스트할 특성을 선택합니다.
- 대상 — 참여할 사용자 수를 구성하고 특성 변형 간에 트래픽을 분할하는 방법을 구성합니다.
- 지표 — 실험의 성공 여부를 결정하는 지표를 지정합니다. 클릭 수를 예로 들 수 있습니다.

실험 생성을 완료한 후 실험 기간을 지정하고 실험을 시작합니다. CloudWatch Evidently에서 실험을 생성하고 시작하는 방법에 대한 단계별 지침은 Amazon CloudWatch 사용 설명서의 [실험 생성](#)을 참조하세요.

5. 실험이 실행되는 동안 CloudWatch Evidently에서 결과를 확인할 수 있습니다. 실험 결과 보기에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 [대시보드에서 실험 결과 보기](#)를 참조하세요.

샘플 구현

다음 샘플 구현은 CloudWatch Evidently를 사용하여 A/B 테스트를 구현하는 방법을 보여줍니다.

- A/B 테스트 구현을 위한 소스 코드가 포함된 실시간 API의 전체 예는 샘플 GitHub 리포지토리의 [실시간 개인화 API](#)를 AWS참조하세요.
- 개인화 및 A/B 테스트에 대한 워크숍이 포함된 샘플 리테일 웹 애플리케이션을 보려면 샘플 GitHub 리포지토리의 [리테일 데모 스토어](#)를 AWS참조하세요. CloudWatch Evidently와 리테일 데모 스토어를 사용하여 A/B 실험을 생성하는 방법을 설명하는 노트북은 [리테일 데모 스토어 실험 워크숍 - CloudWatch Evidently](#)를 참조하세요.
- CloudWatch Evidently에서 A/B 테스트를 사용하는 방법과 샘플 반응 애플리케이션을 사용하는 방법을 설명하는 자습서는 Amazon CloudWatch 사용 설명서의 [자습서: Evidently의 샘플 애플리케이션을 사용한 A/B 테스트](#)를 참조하세요.

검색 결과 개인화 OpenSearch

Amazon Personalize를 사용하여 오픈 소스 또는 OpenSearch Amazon OpenSearch 서비스의 결과를 사용자에게 개인화할 수 있습니다.

[OpenSearch](#) Apache 2.0 라이선스를 기반으로 하는 자체 관리형 오픈 소스 검색 서비스입니다.

[Amazon OpenSearch Service](#)는 AWS 클라우드에서 OpenSearch 리소스를 배포, 운영 및 확장하는 데 도움이 되는 관리형 서비스입니다. Amazon OpenSearch 서비스를 사용하면 결과를 OpenSearch 검색하고 순위를 매깁니다.

쿼리 결과의 순위를 매길 때는 [BM-25](#) 라는 확률적 순위 프레임워크를 OpenSearch 사용하여 관련성 점수를 계산합니다. 특정 키워드가 문서에 더 자주 나타나는 경우 BM-25 키워드는 해당 문서에 더 높은 관련성 점수를 부여합니다. OpenSearch 순위는 클릭스루 데이터와 같은 사용자 행동을 고려하지 않습니다.

Amazon Personalize와 함께 OpenSearch 사용하는 경우 Amazon Personalize는 사용자의 과거 행동, 항목에 대한 메타데이터, 사용자에게 대한 모든 메타데이터를 기반으로 결과의 OpenSearch 순위를 다시 매깁니다. OpenSearch 그런 다음 애플리케이션에 검색 응답을 반환하기 전에 순위 조정 작업을 통합합니다. 결과에 적용할 때 Amazon Personalize에서 순위에 OpenSearch 부여하는 가중치를 제어할 수 OpenSearch 있습니다.

이러한 순위 재조정을 통해 결과가 더욱 매력적이고 사용자의 관심사와 관련성이 높아질 수 있습니다. 이로 인해 애플리케이션의 클릭률과 전환율이 증가할 수 있습니다. 검색으로 전자 상거래 애플리케이션의 결과를 개선할 수 있는 방법을 설명하는 사용 사례의 예는 [사용 사례 예제](#) 단원을 참조하세요.

OpenSearch 결과를 개인화하기 전에 에 나열된 요구 사항을 검토하십시오. [지침 및 요구 사항](#)

주제

- [사용 사례 예제](#)
- [검색 워크플로우](#)
- [Personalize 검색 순위 플러그인 작동 방식](#)
- [추가 정보](#)
- [지침 및 요구 사항](#)
- [플러그인 설정 OpenSearch 및 설치](#)
- [플러그인 구성](#)
- [OpenSearch 쿼리에 플러그인 적용](#)
- [OpenSearch 결과를 플러그인 결과와 비교](#)

• [플러그인 모니터링](#)

사용 사례 예제

Amazon Personalize를 사용하여 OpenSearch 결과의 순위를 다시 매기면 사용자에게 더 관련성이 높은 검색 결과를 제공할 수 있습니다. 예를 들어 자동차를 판매하는 전자 상거래 애플리케이션이 있을 수 있습니다. 사용자가 Toyota 자동차에 대한 쿼리를 입력하고 결과를 개인화하지 않으면 데이터의 키워드를 기반으로 Toyota에서 만든 자동차 목록이 반환됩니다. OpenSearch 이 목록은 모든 사용자에게 대해 동일한 순서로 순위가 매겨집니다.

그러나 Amazon Personalize를 사용하여 결과를 개인화하는 경우 특정 사용자의 행동 (예: 클릭) 을 기반으로 관련성 순으로 이러한 차량의 OpenSearch 순위를 다시 매깁니다. 사용자가 클릭할 가능성이 가장 높은 차량이 1위에 랭크됩니다.

OpenSearch 결과를 개인화할 때 Amazon Personalize에서 순위에 OpenSearch 부여하는 가중치 (강조) 를 제어할 수 있습니다. 이 예를 계속 살펴보자면, 사용자가 특정 연도의 특정 유형의 자동차 (예: 2008 Toyota Prius) 를 검색하는 경우 원래 순위에 더 중점을 두는 것이 좋습니다. OpenSearch

그러나 모든 Toyota 차량 검색과 같이 광범위한 결과를 제공하는 보다 일반적인 쿼리의 경우 개인 맞춤에 중점을 둘 수 있습니다. 이렇게 하면 목록 맨 위에 있는 자동차가 특정 사용자와 더 관련성이 높아집니다.

검색 워크플로우

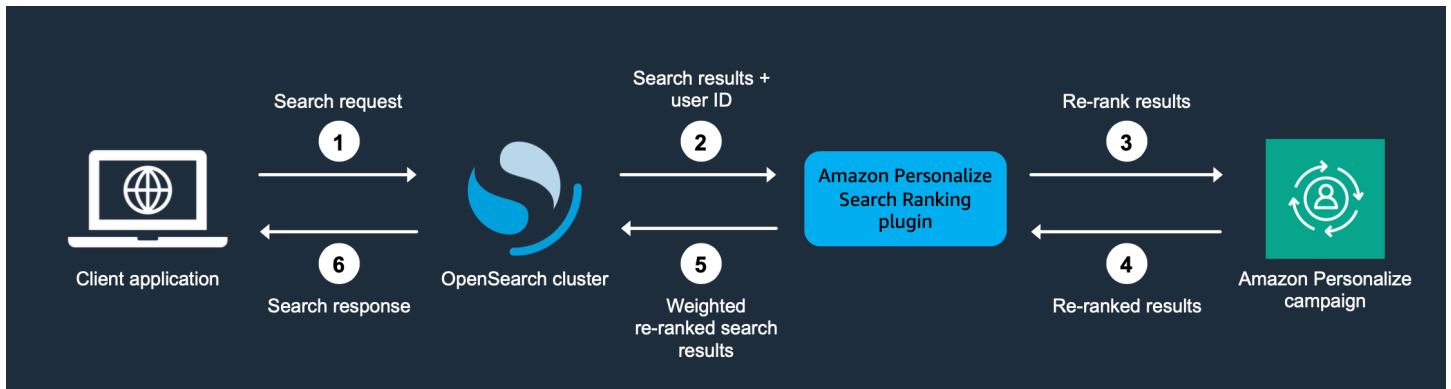
OpenSearch 결과를 개인화하려면 다음과 같이 하십시오.

1. Personalize 설정 — 아직 설정하지 않았다면 [Personalize 설정](#)의 단계를 완료하여 보안 인증을 설정하고 Personalize에 대한 권한을 설정합니다. 결과를 개인화하기 OpenSearch 위해 AWS SDK를 설정할 필요가 없습니다.
2. Personalize 워크플로우 완료 — Personalize 워크플로우를 완료하여 데이터를 가져오고, 순위 레시피로 솔루션을 생성하고, 사용자 지정 솔루션 버전을 학습시키고, 캠페인에 배포합니다. -순위 레시피만 사용할 수 있습니다. 항목 상호 작용 데이터 세트를 만들어야 합니다. 사용자 데이터세트와 항목 데이터세트는 선택사항입니다. 자세한 정보는 [Personalize 워크플로우](#)을 참조하세요.
3. Amazon Personalize 검색 순위 플러그인 설정 OpenSearch 및 설치 — 아직 설정하지 않았다면 OpenSearch 서비스 도메인 또는 오픈 소스 OpenSearch 클러스터를 설정하십시오. 그런 다음 Personalize 검색 순위 플러그인을 설치합니다. 이 플러그인은 Personalize와 순위 재조정 결과와의 통신을 처리합니다. 자세한 내용은 [플러그인 설정 OpenSearch 및 설치](#) 단원을 참조하세요.

4. Personalize 검색 순위 플러그인 구성 – 플러그인을 구성하려면 검색 파이프라인을 생성합니다. 검색 파이프라인은 요청 및 응답 프로세서 세트입니다. 플러그인용 파이프라인을 생성할 때 `personalized_search_ranking` 응답 프로세서에서 Personalize 리소스를 지정합니다. 또한 결과의 순위를 다시 매길 때 플러그인이 Personalize의 결과에 부여하는 가중치를 구성합니다. 자세한 정보는 [플러그인 구성](#)을 참조하세요.
5. OpenSearch 쿼리에 Amazon Personalize 검색 순위 플러그인 적용 — [Amazon Personalize 검색 순위 플러그인을 인덱스에 대한 모든 쿼리와 응답에 적용할 수 있습니다](#). OpenSearch 플러그인을 개별 OpenSearch 쿼리에 적용할 수도 있습니다. 자세한 정보는 [OpenSearch 쿼리에 플러그인 적용](#)을 참조하세요.
6. 결과 비교 — Amazon Personalize 검색 순위 플러그인은 쿼리 응답에서 검색 결과의 순위를 다시 매깁니다. OpenSearch Amazon Personalize의 순위와 순위를 모두 고려합니다. OpenSearch 개인 맞춤을 사용하는 쿼리와 그렇지 않은 쿼리의 결과를 비교해 보면 결과 순위가 어떻게 재조정되는지 이해할 수 있습니다. 자세한 내용은 [OpenSearch 결과를 플러그인 결과와 비교](#) 단원을 참조하세요.
7. Personalize 검색 순위 플러그인 모니터링 — Personalize 검색 순위 플러그인을 검색 쿼리에 적용하면 검색 파이프라인에 대한 지표를 가져와서 플러그인을 모니터링할 수 있습니다. 자세한 내용은 [플러그인 모니터링](#) 단원을 참조하세요.

Personalize 검색 순위 플러그인 작동 방식

다음 다이어그램은 Personalize 검색 순위 플러그인의 작동 방식을 보여줍니다.



1. 고객의 쿼리를 OpenSearch 서비스 도메인 또는 오픈소스 OpenSearch 클러스터에 제출합니다.
2. OpenSearch 쿼리 응답 (쿼리와 관련된 항목 목록) 과 사용자 ID를 Amazon Personalize 검색 순위 플러그인으로 보냅니다.
3. 플러그인은 Personalize 캠페인에 대한 응답으로 항목과 사용자를 전송하여 순위를 매깁니다. 검색 파이프라인의 레시피 및 캠페인 Amazon 리소스 이름(ARN) 값을 사용하여 사용자에게 개인 맞춤형

순위를 부여합니다. 추천을 위해 GetPersonalizedRanking API 작업을 사용합니다. 요청에서 쿼리를 수행하는 사용자의 UserID와 쿼리에서 OpenSearch 반환된 항목을 `inputList`에 전달합니다.

4. Personalize가 순위가 조정된 결과를 플러그인에 반환합니다.
5. 플러그인은 검색 결과를 재정렬하여 OpenSearch 서비스 도메인 또는 오픈소스 클러스터에 반환합니다. OpenSearch Personalize 캠페인의 응답과 설정 중에 지정한 개인 맞춤에 대한 강조에 따라 결과의 순위를 다시 매깁니다.
6. 오픈소스 OpenSearch 클러스터 또는 OpenSearch 서비스 도메인은 최종 결과를 애플리케이션에 반환합니다.

추가 정보

다음 리소스는 사용에 대한 추가 정보를 제공합니다 OpenSearch.

- 오픈소스 시작에 대한 자세한 내용은 OpenSearch [Quickstart](#)를 참조하십시오.
- OpenSearch 서비스를 시작하는 방법에 대한 자세한 내용은 Amazon 서비스 개발자 안내서의 [Amazon OpenSearch OpenSearch Service 시작하기](#)를 참조하십시오.
- Personalize의 개인 맞춤형-순위 레시피에 대한 자세한 내용은 [개인 맞춤형-순위 레시피](#) 단원을 참조하세요.

지침 및 요구 사항

이 단원에는 Personalize 검색 순위 플러그인의 사용 요구 사항이 포함되어 있습니다. 또한 Amazon OpenSearch Service 또는 오픈 소스에 대한 권한을 설정하는 방법도 설명합니다 OpenSearch.

주제

- [플러그인 요구 사항](#)
- [아마존 OpenSearch 서비스 권한 설정](#)
- [오픈 소스 권한 설정 OpenSearch](#)

플러그인 요구 사항

에서 OpenSearch 결과를 개인화하기 전에 Amazon Personalize 검색 순위 플러그인에 대한 다음 지침 및 요구 사항을 참고하십시오.

- OpenSearch 버전 2.9.0 이상을 사용해야 합니다. Amazon OpenSearch Service를 사용하는 경우 도메인은 버전 2.9 이상을 사용해야 합니다.
- 아직 하지 않았다면 [권한 설정](#)의 지침을 완료하여 사용자에게 Personalize에 액세스할 수 있는 권한을 부여하고 Personalize에 있는 리소스에 액세스할 수 있는 권한을 Personalize에 부여합니다.
- OpenSearch 서비스 도메인 또는 오픈 소스 OpenSearch 클러스터에서 Amazon Personalize 리소스에 액세스할 수 있어야 합니다.
 - OpenSearch 서비스 도메인에 대한 액세스 권한 부여에 대한 자세한 내용은 [아마존 OpenSearch 서비스 권한 설정](#)을 참조하십시오.
 - OpenSearch 클러스터에 대한 액세스 권한 부여에 대한 자세한 내용은 [오픈 소스 권한 설정 OpenSearch](#)을 참조하십시오.
- 사용자 지정 Personalize 리소스만 사용할 수 있습니다. 도메인 데이터세트 그룹을 생성한 경우에도 사용자 지정 리소스를 추가할 수 있습니다.
- 사용자 지정 레시피 개인 맞춤형-순위만 사용할 수 있습니다. 이 레시피에 대한 자세한 내용은 [개인 맞춤형-순위 레시피](#) 단원을 참조하세요.
- Amazon Personalize에서 항목 상호 작용 데이터 세트를 생성해야 합니다. 항목 및 사용자 데이터 세트는 선택사항입니다.
- Personalize 검색 순위 플러그인을 사용할 때는 Personalize 필터를 적용할 수 없습니다.
- 기본적으로 플러그인은 의 색인된 문서가 Amazon Personalize 데이터의 OpenSearch itemID와 일치한다고 가정합니다. `_id` OpenSearch 데이터가 Amazon Personalize ItemID에 해당하는 다른 필드를 사용하는 경우 플러그인을 구성할 때 필드 이름을 지정해야 합니다.
- 사용자가 쿼리할 때 사용하는 사용자 ID가 Personalize로 가져오는 데이터의 사용자 ID와 일치해야 합니다.
- 플러그인은 상위 500개 검색 결과의 순위만 다시 매깁니다. OpenSearch 나머지 항목은 순위가 다시 매겨지지 않으며 목록 맨 아래에 있습니다.

아마존 OpenSearch 서비스 권한 설정

아마존 OpenSearch 서비스를 사용하는 경우 서비스 도메인에서 Amazon Personalize 리소스에 액세스할 수 있어야 합니다 OpenSearch .

권한을 설정하려면

1. 리소스가 동일한 계정인지 다른 계정인지에 따라 리소스에 액세스할 권한이 있는 IAM 서비스 역할을 하나 이상 생성하십시오.

- OpenSearch 서비스와 Amazon Personalize 리소스가 동일한 계정에 있는 경우 서비스에 대한 OpenSearch IAM 서비스 역할을 생성하고 Amazon Personalize 캠페인에서 개인화된 순위를 얻을 수 있는 권한을 부여합니다. 자세한 정보는 [리소스가 동일한 계정에 있는 경우 권한 구성](#)을 참조하세요.
 - OpenSearch 서비스와 Amazon Personalize 리소스가 별도의 계정에 있는 경우 두 개의 IAM 서비스 역할을 생성합니다. 서비스 리소스로 계정에 하나를 생성하고 OpenSearch 서비스 리소스에 대한 액세스 권한을 부여합니다 OpenSearch . 그리고 Amazon Personalize 리소스로 계정에 계정을 만들고 Amazon Personalize 캠페인에서 개인화된 순위를 얻을 수 있는 권한을 부여합니다. 자세한 정보는 [리소스가 서로 다른 계정에 있는 경우 권한을 구성합니다.](#)을 참조하세요.
2. OpenSearch 서비스 도메인에 액세스하는 사용자 또는 역할에 서비스용으로 생성한 IAM 서비스 역할에 대한 PassRole 권한을 부여하십시오. OpenSearch 자세한 정보는 [Amazon OpenSearch 서비스 도메인 보안 구성](#)을 참조하세요.

주제

- [리소스가 동일한 계정에 있는 경우 권한 구성](#)
- [리소스가 서로 다른 계정에 있는 경우 권한을 구성합니다.](#)
- [Amazon OpenSearch 서비스 도메인 보안 구성](#)

리소스가 동일한 계정에 있는 경우 권한 구성

OpenSearch 서비스와 Amazon Personalize 리소스가 동일한 계정에 있는 경우 서비스에 대한 IAM 서비스 역할을 생성해야 합니다. OpenSearch 이 역할에 Personalize 캠페인에서 개인 맞춤형 순위를 얻을 수 있는 권한이 있어야 합니다. Amazon Personalize 캠페인에서 개인화된 순위를 얻을 수 있는 권한을 OpenSearch 서비스 서비스 역할에 부여하려면 다음이 필요합니다.

- 역할의 신뢰 정책은 OpenSearch 서비스에 대한 AssumeRole 권한을 부여해야 합니다. 신뢰 정책 예제는 [신뢰 정책 예제](#) 단원을 참조하세요.
- 역할에 Personalize 캠페인에서 개인 맞춤형 순위를 얻을 수 있는 권한이 있어야 합니다. 정책 예제는 [권한 정책 예제](#) 단원을 참조하세요.

IAM 역할 생성에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 역할 생성](#) 단원을 참조하세요. 역할에 정책을 연결하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 자격 증명 권한 추가 및 제거](#) 단원을 참조하세요.

서비스에 대한 IAM 서비스 역할을 생성한 후에는 OpenSearch 서비스 도메인에 액세스하는 사용자 또는 역할에 OpenSearch 서비스 역할 PassRole 권한을 부여해야 합니다. OpenSearch 자세한 정보는 [Amazon OpenSearch 서비스 도메인 보안 구성](#)을 참조하세요.

주제

- [신뢰 정책 예제](#)
- [권한 정책 예제](#)

신뢰 정책 예제

다음 신뢰 정책 예제는 OpenSearch 서비스에 대한 AssumeRole 권한을 부여합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "",
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Principal": {
      "Service": [
        "es.amazonaws.com"
      ]
    }
  }]
}
```

권한 정책 예제

다음 정책 예제에서는 Personalize 캠페인에서 개인 맞춤형 순위를 얻을 수 있는 최소 권한을 역할에 부여합니다. Campaign ARN에 Personalize 캠페인의 Amazon 리소스 이름(ARN)을 지정합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "personalize:GetPersonalizedRanking"
      ],
      "Resource": "Campaign ARN"
    }
  ]
}
```

```

    }
  ]
}

```

리소스가 서로 다른 계정에 있는 경우 권한을 구성합니다.

OpenSearch Service와 Amazon Personalize 리소스가 별도의 계정에 있는 경우 각 계정에서 IAM 역할을 생성하고 해당 역할에 계정의 리소스에 대한 액세스 권한을 부여합니다.

여러 계정에 대한 권한을 설정하려면

1. Amazon Personalize 캠페인이 있는 계정에서 Amazon Personalize 캠페인에서 개인화된 순위를 얻을 수 있는 권한이 있는 IAM 역할을 생성하십시오. 플러그인을 구성할 때 `personalized_search_ranking` 응답 프로세서의 `external_account_iam_role_arn` 파라미터에 이 역할의 ARN을 지정합니다. 자세한 정보는 [플러그인 구성](#)을 참조하세요.

정책 예제는 [권한 정책 예제](#)을 참조하십시오.

2. OpenSearch 서비스 도메인이 있는 계정에서 OpenSearch 서비스 AssumeRole 권한을 부여하는 신뢰 정책이 포함된 역할을 생성합니다. 플러그인을 구성할 때 `personalized_search_ranking` 응답 프로세서의 `iam_role_arn` 파라미터에 이 역할의 ARN을 지정합니다. 자세한 정보는 [플러그인 구성](#)을 참조하세요.

신뢰 정책 예제는 [신뢰 정책 예제](#) 단원을 참조하세요.

3. 다른 역할에 AssumeRole 권한을 부여하도록 각 역할을 수정하십시오. 예를 들어 Amazon Personalize 리소스에 액세스할 수 있는 역할의 경우 IAM 정책은 OpenSearch 서비스 도메인이 있는 계정의 역할에 다음과 같은 역할 수입 권한을 부여합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "",
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::<Account number for role with access to
OpenSearch Service domain>:role/roleName"
  }]
}

```

- 서비스 도메인이 있는 계정에서 OpenSearch 서비스 도메인에 액세스하는 사용자 또는 역할에 방금 생성한 OpenSearch OpenSearch 서비스 역할에 대한 PassRole 권한을 부여하십시오. 자세한 정보는 [Amazon OpenSearch 서비스 도메인 보안 구성](#)을 참조하세요.

Amazon OpenSearch 서비스 도메인 보안 구성

OpenSearch 서비스와 함께 플러그인을 사용하려면 도메인에 액세스하는 사용자 또는 역할에 방금 생성한 [서비스의 IAM 서비스 역할에 대한 PassRole OpenSearch](#) 권한이 있어야 합니다. 또한 사용자 또는 역할에 es:ESHttpGet 및 es:ESHttpPut 작업을 수행할 권한이 있어야 합니다.

OpenSearch 서비스에 대한 액세스를 구성하는 방법에 대한 자세한 내용은 [Amazon OpenSearch Service 개발자 안내서의 Amazon OpenSearch Service 보안](#)을 참조하십시오. 정책 예제는 [OpenSearch 서비스 사용자 또는 역할에 대한 정책 예제](#) 단원을 참조하세요.

OpenSearch 서비스 사용자 또는 역할에 대한 정책 예제

다음 IAM 정책 예제는 서비스를 위해 생성한 IAM 서비스 역할에 대한 PassRole 권한을 사용자 또는 역할에 부여합니다. OpenSearch [리소스가 동일한 계정에 있는 경우 권한 구성](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "OpenSearch Service role ARN"
    }
  ]
}
```

다음 IAM 정책은 Service를 사용하여 파이프라인을 생성하고 쿼리를 검색할 수 있는 최소 권한을 부여합니다. OpenSearch

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "es:ESHttpGet",
        "es:ESHttpPut"
      ]
    }
  ]
}
```

```

    ],
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:ResourceTag/environment": [
          "production"
        ]
      }
    }
  }
}
]
}
}

```

오픈 소스 권한 설정 OpenSearch

오픈 소스를 OpenSearch 사용하는 경우 오픈 검색 클러스터에서 Amazon Personalize 리소스에 액세스할 수 있어야 합니다. 액세스 권한을 부여하려면 다음과 같이 합니다.

- OpenSearch 처음부터 설정하는 경우 [빠른 시작 bash 스크립트](#)를 사용하여 Docker 컨테이너에서 OpenSearch 클러스터를 실행할 수 있습니다. 스크립트는 프로필의 기본 자격 증명을 사용합니다. AWS 스크립트를 실행할 때 대체 프로파일을 지정할 수 있습니다.

이러한 자격 증명은 Amazon Personalize 캠페인에 대한 GetPersonalizedRanking 작업을 수행할 권한이 있는 사용자 또는 역할과 연결되어야 합니다. IAM 정책 예제는 [IAM 정책 예시](#) 단원을 참조하세요. 또는 보안 인증에 이러한 권한이 있는 역할을 수입할 수 있는 권한이 있어야 합니다. Personalize 검색 순위 플러그인에 대한 파이프라인을 생성할 때 이 역할에 대한 Amazon 리소스 이름(ARN)을 지정해야 합니다.

- [퀵 스타트 bash 스크립트](#)를 사용하지 않는 경우 키스토어에 자격 증명을 수동으로 추가할 수 있습니다. OpenSearch . 이러한 자격 증명은 Amazon Personalize 캠페인에 대한 GetPersonalizedRanking 작업을 수행할 권한이 있는 사용자 또는 역할과 일치해야 합니다.

AWS 자격 증명을 OpenSearch 키스토어에 수동으로 추가하려면 OpenSearch 클러스터가 실행 중인 위치 (예: Docker 컨테이너) 에서 다음 명령을 실행합니다. 그런 다음 각 보안 인증 정보를 입력합니다. 세션 토큰을 사용하지 않는 경우 명령의 마지막 줄을 생략해도 됩니다.

```

opensearch-keystore add \
personalized_search_ranking.aws.access_key \
personalized_search_ranking.aws.secret_key \
personalized_search_ranking.aws.session_token

```


- Amazon EC2 인스턴스에서 OpenSearch 클러스터를 실행하는 경우 IAM 인스턴스 프로파일로 권한을 부여할 수 있습니다. 역할에 첨부된 정책은 Amazon Personalize 캠페인에 대한 GetPersonalizedRanking 작업을 수행할 권한을 해당 역할에 부여해야 합니다. 또한 이 역할은 역할을 위임할 수 있는 권한을 Amazon EC2에 부여해야 합니다.

Amazon EC2 인스턴스 프로파일에 대한 자세한 내용은 [인스턴스 프로파일 사용](#) 단원을 참조하세요. 정책 예제는 [IAM 정책 예시](#) 단원을 참조하세요.

IAM 정책 예시

다음 정책 예제에서는 Personalize 캠페인에서 개인 맞춤형 순위를 얻을 수 있는 최소 권한을 사용자나 역할에 부여합니다. Campaign ARN에 Personalize 캠페인의 Amazon 리소스 이름(ARN)을 지정합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "personalize:GetPersonalizedRanking"
      ],
      "Resource": "Campaign ARN"
    }
  ]
}
```

또한 Amazon EC2 인스턴스에서 OpenSearch 클러스터를 실행하고 IAM 인스턴스 프로파일로 권한을 부여하는 경우 역할에 대한 신뢰 정책이 다음과 같은 Amazon AssumeRole EC2 권한을 부여해야 합니다. Amazon EC2 인스턴스 프로파일에 대한 자세한 내용은 [인스턴스 프로파일 사용](#) 단원을 참조하세요.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
    }
  ]
}
```

```

        "Action": "sts:AssumeRole"
    }
}
}

```

플러그인 설정 OpenSearch 및 설치

Amazon Personalize 검색 순위 플러그인은 서비스 도메인 또는 오픈 소스 클러스터에서 OpenSearch Amazon Personalize와의 통신을 처리합니다. OpenSearch 또한 순위 재조정 결과도 처리합니다. 액세스 OpenSearch 방법에 따라 다음과 같이 플러그인을 OpenSearch 설정하고 설치합니다.

- Amazon OpenSearch Service를 사용하는 경우 서비스에 도메인을 생성하고, 데이터를 수집하고, 플러그인을 OpenSearch 설치하여 설정합니다. OpenSearch
- 오픈 소스를 OpenSearch 사용하는 경우 OpenSearch 클러스터를 생성하고, 데이터를 수집하고, 플러그인을 설치합니다.

주제

- [아마존 OpenSearch 서비스 설정](#)
- [오픈 소스 설정 OpenSearch](#)

아마존 OpenSearch 서비스 설정

Amazon Personalize 워크플로를 완료하고 에 [지침 및 요구 사항](#) 나열된 요구 사항을 충족하면 아마존 OpenSearch 서비스를 설정하고 Amazon Personalize 검색 순위 플러그인을 설치할 준비가 된 것입니다.

Amazon OpenSearch Service를 설정하려면 도메인을 생성하고 데이터를 수집한 다음 플러그인을 설치합니다. 이미 도메인을 생성하고 데이터를 수집한 경우 3단계로 건너뛰면 됩니다.

서비스를 OpenSearch 설정하려면

1. 아직 하지 않았다면 OpenSearch 서비스 도메인에서 Amazon Personalize 리소스에 액세스할 수 [아마존 OpenSearch 서비스 권한 설정](#) 있도록 다음 단계를 완료하십시오.
2. 아직 생성하지 않았다면 OpenSearch 서비스 도메인을 생성하십시오. OpenSearch 서비스 도메인은 오픈소스 OpenSearch 클러스터와 동의어입니다. 도메인은 지정된 설정, 인스턴스 유형, 인스턴스 수, 스토리지 리소스를 갖고 있는 클러스터입니다.

- 테스트 도메인 구성에 대한 간략한 자습서는 [Amazon OpenSearch Service 개발자 안내서의 “시작하기”](#) 섹션에서 [1단계: Amazon OpenSearch 서비스 도메인 생성](#)을 참조하십시오.
 - 자세한 단계는 [Amazon OpenSearch Service 도메인 생성 및 관리를](#) 참조하십시오.
3. 아직 아이템을 OpenSearch 서비스에 인제스트하지 않았다면 아이템을 인제스트하세요.
- 소량의 테스트 데이터를 서비스에 [업로드하는 방법에 대한 간략한 자습서](#)는 [Amazon OpenSearch Service 개발자 안내서의 “시작하기”](#) 섹션에서 [색인 생성을 위해 2단계: Amazon OpenSearch 서비스에 데이터 업로드](#)를 참조하십시오. OpenSearch
 - 데이터 수집에 대한 자세한 내용은 Amazon [OpenSearch OpenSearch Service 개발자 안내서의 Amazon Service에서의 데이터 인덱싱](#)을 참조하십시오.
4. Amazon_Personalize_Search_Ranking_Plugin 플러그인을 도메인과 연결합니다. 플러그인은 사전 설치되어 있으므로 S3에서 가져올 필요가 없습니다. OpenSearch 서비스 패키지를 연결하는 것과 같은 방식으로 플러그인을 연결합니다.

OpenSearch 서비스 패키지 연결에 대한 자세한 내용은 [Amazon OpenSearch Service용 사용자 지정 패키지를](#) 참조하십시오.

도메인을 생성하고, 데이터를 수집하고, Personalize 검색 순위 플러그인을 설치했으면 플러그인을 구성할 준비가 된 것입니다. 검색 파이프라인을 생성하고 personalized_search_ranking 응답 프로세서를 지정하여 구성합니다. 자세한 정보는 [플러그인 구성](#)을 참조하세요.

오픈 소스 설정 OpenSearch

Personalize 워크플로우를 완료하고 [지침 및 요구 사항](#)에 나열된 요구 사항을 충족시켰으면 오픈 소스를 설정하고 Personalize 검색 순위 플러그인을 설치할 준비가 된 것입니다.

이미 OpenSearch 클러스터를 실행 중인 경우 플러그인을 수동으로 설치할 수 있습니다. 클러스터가 실행되지 않는 경우 bash 스크립트를 사용하여 플러그인을 처음부터 설치할 OpenSearch 수 있습니다.

주제

- [기존 OpenSearch 클러스터에 플러그인 수동 설치](#)
- [퀵스타트 스크립트를 사용한 클러스터 설정 및 플러그인 설치](#)

기존 OpenSearch 클러스터에 플러그인 수동 설치

이미 OpenSearch 클러스터가 있는 경우 OpenSearch GitHub 리포지토리에서 직접 클러스터에 플러그인을 수동으로 설치할 수 있습니다.

플러그인을 수동으로 설치하려면

1. 다음 명령을 사용하여 OpenSearch 클러스터를 시작합니다.

```
bin/opensearch
```

2. 아직 업로드하지 않았다면 카탈로그 데이터를 OpenSearch 클러스터에 업로드하세요. 데이터를 업로드할 때 OpenSearch 색인을 만들고 필드 매핑을 정의합니다. 그런 다음 해당 색인에 데이터를 업로드합니다. 예제는 [샘플 데이터를 사용하여 색인 및 필드 매핑 생성](#) 단원을 참조하세요.
3. 다음 명령을 사용하여 플러그인을 설치합니다.

```
bin/opensearch-plugin install https://github.com/opensearch-project/search-processor/releases/download/2.9.0/opensearch-search-processor-2.9.0.0.zip
```

플러그인 설치에 대한 자세한 내용은 [플러그인 설치](#) 단원을 참조하세요.

Personalize 검색 순위 플러그인을 설치했으면 구성할 준비가 된 것입니다. 검색 파이프라인을 생성하고 `personalized_search_ranking` 응답 프로세서를 지정하여 플러그인을 구성합니다. 자세한 내용은 [플러그인 구성](#) 단원을 참조하세요.

퀵스타트 스크립트를 사용한 클러스터 설정 및 플러그인 설치

클러스터를 생성하지 않은 경우 퀵스타트 bash 스크립트를 사용하여 OpenSearch 클러스터를 생성할 수 있습니다. 이 스크립트는 Docker 컨테이너에서 OpenSearch 클러스터를 설정하고, 기본 AWS 프로필을 사용하여 자격 증명을 설정하고, Amazon Personalize Search Ranking 플러그인을 설치합니다.

OpenSearch 클러스터를 수동으로 생성하는 방법에 대한 자세한 내용은 설명서의 [Quickstart](#) 지침을 참조하십시오. OpenSearch

quickstart bash 스크립트를 사용하여 플러그인을 설치하려면

1. 스크립트를 실행하기 전에 사용 중인 운영 체제용 [Docker Desktop](#)을 다운로드하여 설치합니다.
2. 에서 [퀵스타트 bash 스크립트](#)를 다운로드하십시오. GitHub
3. 작업 디렉토리에서 다음 명령으로 스크립트를 실행합니다.

```
sh personalized_search_ranking_quickstart.sh
```

이 명령을 사용하면 스크립트는 기본 AWS 프로파일의 자격 증명을 사용합니다. 대체 프로파일을 제공하려면 `--profile` 인수를 사용합니다.

```
sh personalized_search_ranking_quickstart.sh --profile profile-name
```

스크립트를 실행한 후 스크립트로 생성된 고유한 디렉토리에 있는 README 파일에서 스크립트에 대한 자세한 정보를 찾아볼 수 있습니다. 이 디렉토리에는 스크립트에서 사용하는 Dockerfile 및 docker-compose.yml 파일이 저장되어 있습니다. 예를 들면 `../opensearch-personalize-intelligent-ranking-docker.1234/README`입니다.

4. 카탈로그 데이터를 OpenSearch 클러스터에 업로드합니다. 데이터를 업로드할 때 OpenSearch 색인을 만들고 필드 매핑을 정의합니다. 그런 다음 해당 색인에 데이터를 업로드합니다. 예제는 [샘플 데이터를 사용하여 색인 및 필드 매핑 생성](#) 단원을 참조하세요.

Amazon Personalize 검색 순위 플러그인을 OpenSearch 설정하고 설치했으면 구성할 준비가 된 것입니다. 검색 파이프라인을 생성하고 `personalized_search_ranking` 응답 프로세서를 지정하여 플러그인을 구성합니다. 자세한 내용은 [플러그인 구성](#) 단원을 참조하세요.

플러그인 구성

Amazon Personalize 검색 순위 플러그인을 설치했으면 검색 파이프라인을 생성하여 플러그인을 구성할 수 OpenSearch 있습니다.

검색 파이프라인은 생성한 순서대로 순차적으로 실행되는 요청 및 응답 프로세서 세트입니다. 플러그인용 검색 파이프라인을 생성할 때 `personalized_search_ranking` 응답 프로세서를 지정합니다. 검색 파이프라인에 대한 자세한 내용은 [검색 파이프라인](#) 단원을 참조하세요.

주제

- [personalized_search_ranking 응답 프로세서에 대한 필드](#)
- [Amazon OpenSearch 서비스로 파이프라인 생성](#)
- [오픈소스로 파이프라인 만들기 OpenSearch](#)

personalized_search_ranking 응답 프로세서에 대한 필드

personalized_search_ranking 응답 프로세서의 경우 다음 필드를 지정합니다.

- campaign_arn(필수) — 결과를 개인 맞춤하는 데 사용할 Personalize 캠페인의 Amazon 리소스 이름(ARN)을 지정합니다.
- item_id_field (선택 사항) — 인덱싱된 문서의 _id 필드가 Amazon Personalize ItemID와 일치하지 OpenSearch 않는 경우 해당하는 필드의 이름을 지정하십시오. 기본적으로 플러그인은 _id데이터가 Personalize 데이터의 항목 ID와 일치한다고 가정합니다.
- recipe(필수) — 사용할 Personalize 레시피의 이름을 지정합니다. aws-personalized-ranking만 지정할 수 있습니다.
- weight(필수) — 응답 프로세서가 결과의 순위를 다시 매길 때 개인 맞춤에 중점을 두는 부분을 지정합니다. 0.0—1.0 범위의 값을 지정합니다. 값이 1.0에 가까울수록 Personalize의 결과가 더 높은 순위를 차지할 가능성이 높아집니다. 지정하는 경우 개인화가 수행되지 않고 우선합니다. 0.0 OpenSearch
- tag(선택 사항) — 프로세서 식별자를 지정합니다.
- iam_role_arn (OpenSearch 서비스의 경우 필수, 오픈 소스의 경우 선택 사항 OpenSearch) — 서비스의 경우 OpenSearch , 서비스가 Amazon Personalize 리소스에 액세스할 수 있는 권한을 [설정할 때](#) 생성한 역할에 대한 Amazon 리소스 이름 (ARN) 을 제공하십시오. OpenSearch OpenSearch 서비스와 Amazon Personalize 리소스가 서로 다른 계정에 있는 경우 서비스에 대한 AssumeRole OpenSearch 권한을 부여하는 역할을 지정하십시오. 자세한 정보는 [리소스가 서로 다른 계정에 있는 경우 권한을 구성합니다.](#)을 참조하세요.

오픈 소스의 경우 OpenSearch, 여러 역할을 사용하여 조직의 여러 사용자 그룹에 대한 권한을 제한하는 경우 Amazon Personalize에 액세스할 권한이 있는 역할의 ARN을 지정하십시오. OpenSearch 키스토어에 있는 AWS 자격 증명만 사용하는 경우 이 필드를 생략할 수 있습니다.

- aws_region (필수) — 아마존 Personalize 캠페인을 생성한 AWS 지역입니다.
- ignore_failure(선택 사항) — 플러그인이 프로세서 장애를 무시할지 여부를 지정합니다. 값에 true또는 false를 지정합니다. 프로덕션 환경에서는 쿼리 응답이 중단되지 않도록 true를 지정하는 것이 좋습니다. 테스트 환경의 경우 플러그인으로 인해 발생하는 모든 오류를 볼 수 있도록 false를 지정할 수 있습니다.
- external_account_iam_role_arn — OpenSearch 서비스를 사용하고 있고 Amazon Personalize 및 서비스 리소스가 서로 다른 계정에 있는 경우 Amazon OpenSearch Personalize 리소스에 액세스할 권한이 있는 역할의 ARN을 지정하십시오. 이 역할은 Amazon Personalize 리소스와 동일한 계정에 있어야 합니다. 자세한 정보는 [리소스가 서로 다른 계정에 있는 경우 권한을 구성합니다.](#)을 참조하세요.

Amazon OpenSearch 서비스로 파이프라인 생성

다음 Python 코드를 사용하여 OpenSearch 서비스 도메인에 `personalized_search_ranking` 응답 프로세서가 있는 검색 파이프라인을 만들 수 있습니다. `domain endpoint`을 도메인 엔드포인트 URL로 바꿉니다. 예를 들면 `https://<domain name>.<AWS region>.es-staging.amazonaws.com`입니다.

```
import requests
from requests_auth_aws_sigv4 import AWSSigV4

domain_endpoint = 'domain endpoint'
pipeline_name = 'pipeline name'
url = f'{domain_endpoint}/_search/pipeline/{pipeline_name}'
auth = AWSSigV4('es')

headers = {'Content-Type': 'application/json'}

body = {
    "description": "A pipeline to apply custom re-ranking from Amazon Personalize",
    "response_processors": [
        {
            "personalized_search_ranking" : {
                "campaign_arn" : "Amazon Personalize Campaign ARN",
                "item_id_field" : "productId",
                "recipe" : "aws-personalized-ranking",
                "weight" : "0.3",
                "tag" : "personalize-processor",
                "iam_role_arn": "Role ARN",
                "aws_region": "AWS region",
                "ignore_failure": true
            }
        }
    ]
}

try:
    response = requests.put(url, auth=auth, json=body, headers=headers, verify=False)
    print(response.text)
except Exception as e:
    print(f"Error: {e}")
```

`personalized_search_ranking` 응답 프로세서로 검색 파이프라인을 만들었으면 플러그인을 OpenSearch 쿼리에 적용할 준비가 된 것입니다. OpenSearch 색인 또는 개별 OpenSearch 쿼리에 적용할 수 있습니다. 자세한 정보는 [OpenSearch 쿼리에 플러그인 적용](#)을 참조하세요.

오픈소스로 파이프라인 만들기 OpenSearch

다음 curl 명령어를 사용하여 오픈 소스 OpenSearch 클러스터에 `personalized_search_ranking` 응답 프로세서가 있는 검색 파이프라인을 만들 수 있습니다.

```
curl -X PUT "http://localhost:9200/_search/pipeline/pipeline-name" -ku 'admin:admin' --insecure -H 'Content-Type: application/json' -d '{
  "description": "A pipeline to apply custom re-ranking from Amazon Personalize",
  "response_processors" : [
    {
      "personalized_search_ranking" : {
        "campaign_arn" : "Amazon Personalize Campaign ARN",
        "item_id_field" : "productId",
        "recipe" : "aws-personalized-ranking",
        "weight" : "0.3",
        "tag" : "personalize-processor",
        "iam_role_arn": "Role ARN",
        "aws_region": "AWS region",
        "ignore_failure": true
      }
    }
  ]
}'
```

`personalized_search_ranking` 응답 프로세서로 검색 파이프라인을 만들었으면 플러그인을 OpenSearch 쿼리에 적용할 준비가 된 것입니다. OpenSearch 색인 또는 개별 OpenSearch 쿼리에 적용할 수 있습니다. 자세한 정보는 [OpenSearch 쿼리에 플러그인 적용](#)을 참조하세요.

OpenSearch 쿼리에 플러그인 적용

`personalized_search_ranking` 응답 프로세서로 검색 파이프라인을 구성했으면 Amazon Personalize Search Ranking 플러그인을 OpenSearch 쿼리에 적용하고 순위가 다시 매겨진 결과를 볼 수 있습니다.

OpenSearch 쿼리에 플러그인을 적용할 때 검색 파이프라인의 지표를 얻어 플러그인을 모니터링할 수 있습니다. 자세한 정보는 [플러그인 모니터링](#)을 참조하세요.

주제

- [Amazon OpenSearch 서비스 쿼리에 플러그인 적용](#)

- [오픈소스의 쿼리에 플러그인 적용 OpenSearch](#)

Amazon OpenSearch 서비스 쿼리에 플러그인 적용

Personalize 검색 순위 플러그인을 색인에 대한 모든 쿼리와 응답에 적용할 수 있습니다. 플러그인을 개별 쿼리 및 응답에 적용할 수도 있습니다.

- 다음 Python 코드를 사용하면 검색 파이프라인을 색인에 적용할 수 있습니다. 이 접근 방식을 사용하면 이 색인을 사용하는 모든 검색에서 플러그인을 사용하여 검색 결과에 개인 맞춤을 적용합니다.

```
import requests
from requests_auth_aws_sigv4 import AWSSigV4

domain_endpoint = 'domain endpoint'
index = 'index name'
url = f'{domain_endpoint}/{index}/_settings/'
auth = AWSSigV4('es')
headers = {'Content-Type': 'application/json'}
body = {
    "index.search.default_pipeline": "pipeline name"
}
try:
    response = requests.put(url, auth=auth, json=body, headers=headers)
    print(response.text)
except Exception as e:
    print(f"Error: {e}")
```

- 다음 Python 코드를 사용하면 Toyota 브랜드 자동차에 대한 개별 쿼리에 검색 파이프라인을 적용할 수 있습니다.

코드를 업데이트하여 도메인 엔드포인트, OpenSearch 서비스 인덱스, 파이프라인 이름, 쿼리를 지정합니다. `user_id`에, 검색 결과를 얻으려는 사용자의 ID를 지정합니다. 이 사용자는 Personalize 솔루션 버전을 생성하는 데 사용한 데이터에 포함되어 있어야 합니다. 사용자가 없는 경우 Personalize는 인기도에 따라 항목의 순위를 매깁니다.

컨텍스트 메타데이터를 사용하는 경우 `context`에 장치 유형과 같은 사용자의 컨텍스트 메타데이터를 제공합니다. `context` 필드는 선택 사항입니다. 자세한 정보는 [컨텍스트 메타데이터로 추천 관련성 높이기](#)를 참조하세요.

```
import requests
from requests_auth_aws_sigv4 import AWSSigV4
```

```

domain_endpoint = 'domain_endpoint'
index = 'index_name'
url = f'{domain_endpoint}/{index}/_search/'

auth = AWSSigV4('es')
headers = {'Content-Type': 'application/json'}
params = {"search_pipeline": "pipeline-name"}
body = {
    "query": {
        "multi_match": {
            "query": "Toyota",
            "fields": ["BRAND"]
        }
    },
    "ext": {
        "personalize_request_parameters": {
            "user_id": "USER ID",
            "context": { "DEVICE" : "mobile phone" }
        }
    }
}
try:
    response = requests.post(url, auth=auth, params=params, json=body,
headers=headers, verify=False)
    print(response)
except Exception as e:
    print(f"Error: {e}")

```

오픈소스의 쿼리에 플러그인 적용 OpenSearch

Amazon Personalize 검색 순위 플러그인을 색인에 대한 모든 쿼리와 응답에 적용할 수 있습니다. OpenSearch 플러그인을 개별 OpenSearch 쿼리 및 응답에 적용할 수도 있습니다.

- 다음 curl 명령은 로컬에서 실행되는 오픈 소스 OpenSearch 클러스터의 OpenSearch 인덱스에 검색 파이프라인을 적용합니다. 이 접근 방식을 사용하면 이 색인의 모든 검색이 플러그인을 사용하여 검색 결과에 개인 맞춤을 적용합니다.

```

curl -XGET "https://localhost:9200/index/_settings" -ku 'admin:admin' --insecure -H
'Content-Type: application/json' -d'
{

```

```
"index.search.default_pipeline": "pipeline-name"
}
```

- 다음 curl 명령은 로컬에서 실행되는 오픈 소스 OpenSearch 클러스터의 인덱스에 있는 Toyota 브랜드 자동차에 대한 개별 쿼리에 검색 파이프라인을 적용합니다.

`user_id`에, 검색 결과를 얻으려는 사용자의 ID를 지정합니다. 이 사용자는 Personalize 솔루션 버전을 생성하는 데 사용한 데이터에 포함되어 있어야 합니다. 사용자가 없는 경우 Personalize는 인기도에 따라 항목의 순위를 매깁니다. 컨텍스트 메타데이터를 사용하는 경우 `context`에 장치 유형과 같은 사용자의 컨텍스트 메타데이터를 제공합니다. `context` 필드는 선택 사항입니다. 자세한 정보는 [컨텍스트 메타데이터로 추천 관련성 높이기](#)을 참조하세요.

```
curl -XGET "http://localhost:9200/index/_search?search_pipeline=pipeline-name" -ku
'admin:admin' --insecure -H 'Content-Type: application/json' -d'
{
  "query": {
    "multi_match": {
      "query": "Toyota",
      "fields": ["BRAND"]
    }
  },
  "ext": {
    "personalize_request_parameters": {
      "user_id": "USER ID",
      "context": { "DEVICE": "mobile phone" }
    }
  }
}
```

결과 순위가 다시 매겨지는 방식을 이해하기 위해 OpenSearch 대시보드를 사용하여 플러그인을 사용하여 순위가 다시 매겨진 OpenSearch 결과와 결과를 비교할 수 있습니다. 자세한 정보는 [OpenSearch 결과를 플러그인 결과와 비교](#)을 참조하세요.

플러그인을 OpenSearch 쿼리에 적용하면 파이프라인의 메트릭을 가져와서 플러그인을 모니터링할 수 있습니다. OpenSearch 자세한 정보는 [플러그인 모니터링](#)을 참조하세요.

OpenSearch 결과를 플러그인 결과와 비교

Amazon Personalize 검색 순위 플러그인은 Amazon Personalize의 순위와 순위를 기반으로 검색 결과를 재정렬합니다. OpenSearch 플러그인이 결과의 순위를 다시 매기는 방법은 파이프라인에서 `personalized_search_ranking` 응답 프로세서를 구성한 방식에 따라 달라집니다.

개인 맞춤을 사용하거나 사용하지 않고 쿼리를 실행하고 결과를 비교해 보면 결과의 순위가 다시 매겨지는 방식을 이해할 수 있습니다.

주제

- [아마존 OpenSearch 서비스와의 결과 비교](#)
- [결과를 오픈소스와의 비교 OpenSearch](#)

아마존 OpenSearch 서비스와의 결과 비교

개인 맞춤을 사용하거나 사용하지 않고 쿼리를 실행하고 결과를 비교해 보면 결과의 순위가 매겨지는 방식을 이해할 수 있습니다. 다음 Python 코드를 사용하면 서로 다른 두 쿼리를 실행하고 결과를 두 JSON 파일로 출력할 수 있습니다. 첫 번째 메서드는 플러그인을 사용하여 결과의 순위를 다시 매기는 쿼리를 실행합니다. 두 번째 메서드는 개인 맞춤 없이 결과를 생성하는 메서드를 실행합니다.

```
import json
import requests
from requests_auth_aws_sigv4 import AWSSigV4

# Returns re-ranked OpenSearch results using the Amazon Personalize Search Ranking
# plugin.
def get_personalized_results(pipeline_name):
    url = f'{domain}/{index}/_search/'
    auth = AWSSigV4('es')
    headers = {'Content-Type': 'application/json'}
    params = {"search_pipeline": pipeline_name}
    body = {
        "query": {
            "multi_match": {
                "query": "Toyota",
                "fields": ["BRAND"]
            }
        },
        "ext": {
```

```
        "personalize_request_parameters": {
            "user_id": "1"
        }
    }
}
try:
    response = requests.post(url, auth=auth, params=params, json=body,
headers=headers, verify=False)
except Exception as e:
    return f"Error: {e}"
return response.text

# Returns OpenSearch results without personalization.
def get_opensearch_results():
    url = f'{domain}/{index}/_search/'
    auth = AWSSigV4('es')
    headers = {'Content-Type': 'application/json'}
    body = {
        "query": {
            "multi_match": {
                "query": "Toyota",
                "fields": ["BRAND"]
            }
        }
    }
}
try:
    response = requests.post(url, auth=auth, json=body, headers=headers,
verify=False)
except Exception as e:
    return f"Error: {e}"
return response.text

def print_results(file_name, results):
    results_file = open(file_name, 'w')
    results_file.write(json.dumps(results, indent=4))
    results_file.close()

# specify domain endpoint
domain = "DOMAIN_ENDPOINT"
```

```
# specify the region where you created your Amazon Personalize resources and Amazon
  OpenSearch domain
aws_region = "REGION"

# specify the name of the pipeline that uses the Amazon Personalize plugin
pipeline_name = "PIPELINE_NAME"

# specify your Amazon OpenSearch index
index = "INDEX"

# specify names for json files for comparison
personalized_results_file = "personalized_results.json"
opensearch_results_file = "opensearch_results.json"

# get personalized results
personalized_results = json.loads(get_personalized_results(pipeline_name))

# get OpenSearch results without personalization
opensearch_results = json.loads(get_opensearch_results())

# print results to files
print_results(personalized_results_file, personalized_results)
print_results(opensearch_results_file, opensearch_results)
```

결과를 오픈소스와의 비교 OpenSearch

[Dev Tools 콘솔](#)을 사용하여 별도의 두 브라우저 창에서 쿼리를 실행하면 결과의 순위가 어떻게 재조정되는 지를 이해할 수 있습니다. 그런 다음 개인 맞춤을 적용한 쿼리와 사용하지 않은 쿼리의 결과를 비교할 수 있습니다.

Dev Tools 콘솔로 결과를 비교하려면

1. 아직 [플러그인 설정 OpenSearch 및 설치](#) 및 [플러그인 구성](#)의 지침을 따르지 않았다면 지금 따릅니다.
2. OpenSearch 대시보드가 설치되어 있는지 확인하세요. 퀵스타트 bash 스크립트는 대시보드를 설치합니다. OpenSearch 스크립트를 사용하지 않거나 이미 클러스터가 실행 중인 경우 대시보드를 설치해야 합니다. OpenSearch 자세한 내용은 [OpenSearch 대시보드 설치](#)를 참조하십시오.
3. OpenSearch 대시보드 실행. <http://localhost:5601>브라우저에서 열고 OpenSearch 대시보드에 로그인합니다. 기본 보안 인증은 사용자 이름 'admin'과 비밀번호 'admin'입니다.
4. OpenSearch 대시보드 홈 페이지의 관리 메뉴에서 개발 도구를 선택합니다.

5. 별도의 브라우저 창을 열고 Dev Tools 콘솔을 다시 엽니다. 이전 창의 URL을 사용할 수 있습니다.
6. 한 창에서, 개인 맞춤을 위해 순위를 다시 매기지 않은 쿼리를 입력합니다. 다른 창에서, `personalized_search_ranking` 응답 프로세서와 함께 파이프라인을 사용하는 `curl` 명령을 입력합니다. `curl` 명령을 콘솔에 직접 붙여넣으면 콘솔에서 사용하는 형식으로 명령이 자동으로 변환됩니다. 명령의 예는 [OpenSearch 쿼리에 플러그인 적용](#) 단원을 참조하세요.
7. 두 쿼리를 모두 실행하여 결과를 비교합니다.

플러그인 모니터링

OpenSearch 서비스를 사용하는 경우 Amazon의 지표를 통해 플러그인을 모니터링할 수 CloudWatch 있습니다. 자세한 내용은 [Amazon OpenSearch 서비스 도메인 모니터링](#)을 참조하십시오.

Amazon Personalize Search Ranking 플러그인을 OpenSearch 쿼리에 적용할 때 검색 파이프라인에 대한 지표를 가져와서 플러그인을 모니터링할 수 있습니다. 파이프라인 지표에는 `personalized_search_ranking` 응답 프로세서에 대한 실패한 요청 수와 같은 통계가 포함됩니다.

주제

- [Amazon OpenSearch 서비스를 통한 플러그인 모니터링](#)
- [오픈 소스로 플러그인 모니터링 OpenSearch](#)
- [파이프라인 지표 예제](#)

Amazon OpenSearch 서비스를 통한 플러그인 모니터링

다음 Python 코드를 사용하면 모든 파이프라인에 대한 지표를 가져올 수 있습니다. 파이프라인 지표의 예는 [파이프라인 지표 예제](#) 단원을 참조하세요.

```
import requests
from requests_auth_aws_sigv4 import AWSSigV4

domain_endpoint = 'domain endpoint'
url = f'{domain_endpoint}/_nodes/stats/search_pipeline'

auth = AWSSigV4('es')
headers = {'Content-Type': 'application/json'}
try:
    response = requests.get(url, auth=auth, headers=headers, verify=False)
    print(response.text)
except Exception as e:
```

```
print(f"Error: {e}")
```

오픈 소스로 플러그인 모니터링 OpenSearch

다음 코드를 사용하면 모든 파이프라인에 대한 지표를 가져올 수 있습니다. 응답에 모든 검색 파이프라인에 대한 통계가 포함됩니다. 파이프라인 지표의 예는 [파이프라인 지표 예제](#) 단원을 참조하세요.

```
curl -XGET "https://localhost:9200/_nodes/stats/search_pipeline?pretty" -ku
'admin:admin'
```

파이프라인 지표 예제

다음 코드는 에서 반환된 파이프라인 지표를 발췌한 것입니다. OpenSearch 여기에는 서로 다른 두 파이프라인에 대한 통계가 포함된 pipelines 객체만 표시됩니다. personalized_search_ranking 응답 프로세서 목록에서 각 파이프라인에 대해 Personalize 검색 순위 플러그인 지표를 찾아볼 수 있습니다. 모든 지표의 전체 예는 [검색 파이프라인 지표](#) 단원을 참조하세요.

```
{
  ....
  ....
  "pipelines": {
    "pipelineA": {
      "request": {
        "count": 0,
        "time_in_millis": 0,
        "current": 0,
        "failed": 0
      },
      "response": {
        "count": 6,
        "time_in_millis": 2246,
        "current": 0,
        "failed": 0
      },
      "request_processors": [],
      "response_processors": [
        {
          "personalized_search_ranking": {
            "type": "personalized_search_ranking",
            "stats": {
              "count": <number of requests>,

```



```
        "time_in_millis": <time>,
        "current": 0,
        "failed": <number of failed requests>
    }
}
],
},
"pipelineB": {
  "request": {
    "count": 0,
    "time_in_millis": 0,
    "current": 0,
    "failed": 0
  },
  "response": {
    "count": 8,
    "time_in_millis": 2248,
    "current": 0,
    "failed": 0
  },
  "request_processors": [],
  "response_processors": [
    {
      "personalized_search_ranking": {
        "type": "personalized_search_ranking",
        "stats": {
          "count": <number of requests>,
          "time_in_millis": <time>,
          "current": 0,
          "failed": <number of failed requests>
        }
      }
    }
  ]
}
}
....
....
}
```

Personalize 리소스에 태그 지정

태그는 선택적으로 정의하여 특정 유형의 Amazon Personalize AWS 리소스를 비롯한 리소스에 연결할 수 있는 레이블입니다. 리소스는 최대 50개의 태그를 가질 수 있습니다.

태그를 사용하면 용도, 환경 또는 기타 기준과 같은 다양한 방법으로 리소스를 분류하고 관리하는 데 도움이 됩니다. 예를 들어 태그를 사용하여 여러 기능 간에 수익을 분할하거나 다양한 리소스의 개발 환경을 식별할 수 있습니다.

Amazon Personalizes 리소스를 태그별로 검색하려면 Resource Groups 태깅 API의 `GetResources` 작동 시 필터를 사용할 수 있습니다. 자세한 내용은 Resource Groups 태깅 API API 참조 안내서를 참조하십시오 [GetResources](#).

태그를 추가할 수 있는 Personalize 리소스 유형은 다음과 같습니다.

- 배치 추론 작업
- 배치 세그먼트 작업
- 캠페인
- 데이터세트
- 데이터세트 그룹
- 데이터세트 가져오기 및 내보내기 작업
- 이벤트 트래커
- 필터
- 추천자
- 솔루션
- 솔루션 버전

주제

- [지침 및 요구 사항](#)
- [Personalize 리소스에 태그 추가](#)
- [Amazon Personalize 리소스에서 태그 제거](#)
- [IAM 정책에서 태그 사용](#)

지침 및 요구 사항

각 태그는 사용자가 정의하는 필수 태그 키와 선택적 태그 값으로 구성됩니다. 태그 키는 더 구체적인 태그 값에 대해 카테고리화 같은 역할을 하는 일반적인 레이블입니다. 태그 값은 태그 키에 대한 설명자 역할을 합니다.

예를 들어 내부 테스트용과 외부 사용을 위한 두 버전의 Personalize 데이터세트 그룹이 있는 경우 Environment 태그 키를 두 프로젝트 모두에 할당할 수 있습니다. Environment 태그의 태그 값은 데이터세트 그룹의 한 버전의 경우 Test이고 다른 버전의 경우 Production일 수 있습니다.

태그에 적용되는 제한은 다음과 같습니다.

- 리소스당 최대 태그 수 - 50개
- 최대 키 길이 - UTF-8 형식의 유니코드 문자 128자
- 최대 값 길이 - UTF-8 형식의 유니코드 문자 256자
- 태그 키와 값에는 A-Z, a-z, 0-9, 공백 및 `_ . : / = + @ -` (하이픈)이 포함될 수 있습니다. 이것이 태그를 지원하는 AWS 서비스에서 사용할 수 있는 표준 문자 집합입니다. 일부 서비스는 추가 기호를 지원합니다.
- 태그 키와 태그 값은 대/소문자를 구분합니다.
- 연결된 각 리소스에 대해 각 태그 키는 고유해야 하고 하나의 태그 값만 가질 수 있습니다.
- 태그 키와 태그 값은 `aws:` 로 시작할 수 없습니다. AWS 서비스는 `aws:` 로 시작하는 태그를 `aws:` 적용하며 이러한 태그는 수정할 수 없습니다. 이들 태그는 태그 제한에 포함되지 않습니다.
- 태그만을 기반으로 하는 리소스는 업데이트하거나 삭제할 수 없습니다. 사용하는 작업에 따라 Amazon 리소스 이름(ARN) 또는 리소스 ID도 지정해야 합니다.

추가 정보

태그 지정에 대한 자세한 내용은 다음 리소스를 참조하세요.

- AWS AWS 일반 참조의 [태깅 원칙](#)
- [AWS 태깅 전략](#) (다운로드 가능한 PDF)
- AWS AWS IAM 사용 설명서의 [액세스 제어](#)
- [AWS 사용 설명서의 태깅 정책](#) [AWS Organizations](#)

Personalize 리소스에 태그 추가

Amazon Personalize 콘솔, AWS CLI() 또는 SDK를 사용하여 Amazon Personalize 리소스에서 태그 키와 값을 추가, 표시 AWS Command Line Interface, 업데이트 및 제거할 수 있습니다. AWS 다음 예제는 Personalize 데이터세트 그룹에 태그를 추가하는 방법을 보여줍니다. 동일한 방식으로 다른 Amazon Personalize 리소스에 태그를 추가할 수 있습니다.

주제

- [태그 추가\(콘솔\)](#)
- [태그 추가\(AWS CLI\)](#)
- [태그 추가 \(AWS SDK\)](#)

태그 추가(콘솔)

Personalize에서 리소스를 생성할 때 Personalize 콘솔을 사용하여 선택적 태그를 추가할 수 있습니다. 다음 예제에서는 데이터세트 그룹에 태그를 추가합니다.

새 데이터세트 그룹에 태그를 추가하려면

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터세트 그룹 생성을 선택합니다.
3. 이름에 이름을 입력합니다.
4. 도메인에 대해 도메인을 선택합니다.
5. 태그 섹션을 펼치고 새 태그 추가를 선택합니다.
6. 키 및 값에 적절한 값을 입력합니다.

예를 들어, **Environment**과 **Test**을 각각 입력합니다.

7. 태그를 더 추가하려면 새 태그 추가를 선택합니다.

리소스에 최대 50개의 태그를 추가할 수 있습니다.

8. 다음을 선택하여 리소스 생성을 계속합니다.

기존 리소스에 태그를 추가하는 것도 비슷합니다. 리소스를 선택하고 태그 필드를 사용하여 태그를 추가합니다.

태그 추가(AWS CLI)

리소스를 생성하거나 기존 리소스에 태그를 추가할 때 AWS Command Line Interface (AWS CLI) 를 사용하여 태그를 추가할 수 있습니다.

주제

- [리소스를 생성할 때 태그 추가](#)
- [기존 리소스에 태그 추가](#)

리소스를 생성할 때 태그 추가

를 사용하여 새 리소스를 만들고 여기에 태그를 추가하려면 리소스에 적합한 create 명령을 사용하고 tags 매개 변수와 값을 포함시키십시오. AWS CLI예를 들어 다음 명령은 ECOMMERCE 도메인에 대해 myDatasetGroup이라는 이름이 지정된 새 도메인 데이터세트 그룹을 생성하고 다음 태그(Test 태그 값이 있는 Environment태그 키, Owner태그 키 및 xyzCorp값)를 추가합니다.

```
aws personalize create-dataset-group \  
--name myDatasetGroup \  
--domain ECOMMERCE \  
--tags tagKey=Environment,tagValue=Test tagKey=Owner,tagValue=xyzCorp
```

[Amazon Personalize 리소스를 생성하는 데 사용할 수 있는 명령에 대한 자세한 내용은 Amazon Personalize 명령 참조를 참조하십시오. AWS CLI](#)

기존 리소스에 태그 추가

기존 리소스에 태그를 추가하려면 명령을 사용하십시오. tag-resource 리소스의 ARN을 지정하고 파라미터에 태그 키와 값을 입력합니다. tags

```
aws personalize tag-resource \  
--resource-arn resource ARN \  
--tags tagKey=key,tagValue=value
```

태그 추가 (AWS SDK)

AWS SDK를 사용하여 리소스를 생성할 때 태그를 추가하거나 기존 리소스에 태그를 추가할 수 있습니다.

주제

- [리소스를 생성할 때 태그 추가](#)
- [기존 리소스에 태그 추가](#)

리소스를 생성할 때 태그 추가

새 리소스를 만들고 AWS SDK로 태그를 추가하려면 적절한 `create` 방법을 사용하세요. `tags` 파라미터를 사용하여 각 태그의 키-값 쌍을 지정합니다. 예를 들어 다음 코드는 ECOMMERCE 도메인에 대해 `myDatasetGroup`이라는 이름이 지정된 새 도메인 데이터세트 그룹을 생성하고 다음 태그(`Test` 태그 값이 있는 `Environment` 태그 키, `Owner` 태그 키 및 `xyzCorp` 값)를 추가합니다.

SDK for Python (Boto3)

```
import boto3

personalize = boto3.client('personalize')

response = personalize.create_dataset_group(
    name = 'myDatasetGroup',
    domain = 'ECOMMERCE',
    tags = [
        {
            'tagKey': 'Environment',
            'tagValue': 'Test'
        },
        {
            'tagKey': 'Owner',
            'tagValue': 'xyzCorp'
        }
    ]
)
dsg_arn = response['datasetGroupArn']

description = personalize.describe_dataset_group(datasetGroupArn = dsg_arn)
['datasetGroup']

print('Name: ' + description['name'])
print('ARN: ' + description['datasetGroupArn'])
print('Status: ' + description['status'])
```

SDK for Java 2.x

```
public static String createDomainDatasetGroup(PersonalizeClient personalizeClient,
                                             String datasetGroupName,
                                             String domain) {

    try {

        ArrayList <Tag> tags = new ArrayList<>();

        Tag tag1 = Tag.builder()
            .tagKey("Environment")
            .tagValue("Test")
            .build();
        tags.add(tag1);
        Tag tag2 = Tag.builder()
            .tagKey("Owner")
            .tagValue("xyzCorp")
            .build();
        tags.add(tag2);

        CreateDatasetGroupRequest createDatasetGroupRequest =
        CreateDatasetGroupRequest.builder()
            .name(datasetGroupName)
            .domain(domain)
            .tags(tags)
            .build();

        return
        personalizeClient.createDatasetGroup(createDatasetGroupRequest).datasetGroupArn();
    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}
```

기존 리소스에 태그 추가

다음 코드는 기존 Amazon Personalize 리소스에 태그를 추가하는 방법을 보여줍니다. 태그를 추가할 리소스의 Amazon 리소스 이름(ARN)을 지정하고 각 태그에 키-값 쌍을 지정합니다.

SDK for Python (Boto3)

```
import boto3
personalize = boto3.client('personalize')

add_tags_response = personalize.tag_resource(
    resourceArn = "resourceArn",
    tags = [
        {
            'tagKey': 'Environment',
            'tagValue': 'Test'
        },
        {
            'tagKey': 'Owner',
            'tagValue': 'xyzCorp'
        }
    ]
)
```

SDK for Java 2.x

```
public static void tagResource(PersonalizeClient personalizeClient,
                               String resourceArn,
                               String domain) {

    try {

        ArrayList <Tag> tagList = new ArrayList<>();

        Tag tag1 = Tag.builder()
            .tagKey("Environment")
            .tagValue("Test")
            .build();
        tags.add(tag1);
        Tag tag2 = Tag.builder()
            .tagKey("Owner")
            .tagValue("xyzCorp")
            .build();
        tags.add(tag2);

        TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
            .resourceArn(resourceArn)
            .tags(tagList)
```



```
        .build();

        personalizeClient.tagResource(tagResourceRequest);
        System.out.println("Tags have been added to "+ resourceArn);

    } catch (PersonalizeException e) {
        System.out.println(e.awsErrorDetails().errorMessage());
    }
    return "";
}
```

Amazon Personalize 리소스에서 태그 제거

Amazon Personalize 콘솔을 사용하거나 AWS CLI() 또는 [UntagResource](#) SDK를 사용한 API 작업을 사용하여 Amazon Personalize 리소스에서 태그를 제거할 수 있습니다. AWS Command Line Interface AWS 다음 예는 Amazon Personalize 데이터세트 그룹에서 태그를 제거하는 방법을 보여줍니다. 동일한 방법으로 다른 Amazon Personalize 리소스에서 태그를 제거할 수 있습니다.

주제

- [태그 제거 \(콘솔\)](#)
- [태그 제거 \(AWS CLI\)](#)
- [태그 제거 \(AWS SDK\)](#)

태그 제거 (콘솔)

Amazon Personalize에서 리소스에 태그를 추가한 후 Amazon Personalize 콘솔을 사용하여 태그를 제거할 수 있습니다. 다음 예시는 데이터세트 그룹에서 태그를 제거합니다.

데이터세트 그룹에서 태그를 제거하려면

1. <https://console.aws.amazon.com/acm-pca/home>에서 Personalize 콘솔을 열고 계정에 로그인합니다.
2. 데이터세트 그룹을 선택합니다.
3. 페이지 하단에서 태그 탭을 선택하고 태그 관리를 선택합니다.
4. 제거하려는 각 태그에 대해 제거를 선택합니다.
5. [Save] 를 선택하여 태그를 제거합니다.

태그 제거 (AWS CLI)

를 사용하여 기존 리소스에서 태그를 제거하려면 다음 `untag-resource` 명령을 사용합니다. AWS CLI의 `resource-arn` 경우 리소스의 Amazon 리소스 이름 (ARN) 을 지정합니다. 의 `tag-keys` 경우 제거할 태그의 키를 지정합니다.

```
aws personalize untag-resource \
--resource-arn resource ARN \
--tag-keys key1 key2
```

태그 제거 (AWS SDK)

AWS SDK를 사용하여 기존 Amazon Personalize 리소스에서 태그를 제거하려면 API 작업을 사용하십시오. [UntagResource](#) 다음 코드는 Python용 SDK (Boto3) 를 사용하여 데이터세트 그룹에서 여러 태그를 제거하는 방법을 보여줍니다. 의 `resourceArn` 경우 리소스의 Amazon 리소스 이름 (ARN) 을 지정합니다. 의 `tagKeys` 경우 제거할 태그의 키를 지정합니다.

```
import boto3

personalize = boto3.client('personalize')

response = personalize.untag_resource(
    resourceArn="Resource ARN",
    tagKeys=["tag1Key", "tag2Key"]
)
```

IAM 정책에서 태그 사용

태그 구현을 시작한 후 AWS Identity and Access Management (IAM) 정책 및 API 작업에 태그 기반의 리소스 수준 권한을 적용할 수 있습니다. 여기에는 리소스가 생성될 때 리소스에 태그 추가를 지원하는 작업이 포함됩니다. 이러한 방식으로 태그를 사용하면 AWS 계정 내에서 리소스를 만들고 태그를 지정할 권한이 있는 그룹 및 사용자, 태그를 생성, 업데이트 및 제거할 권한이 있는 그룹 및 사용자를 보다 일반적으로 세부적으로 제어할 수 있습니다.

예를 들어, 사용자가 이름이 리소스의 `Owner` 태그 값에 해당하는 모든 Personalize 리소스에 대한 전체 액세스 권한을 갖도록 허용하는 정책을 생성할 수 있습니다.

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "ModifyResourceIfOwner",
    "Effect": "Allow",
    "Action": "personalize:*",
    "Resource": "*",
    "Condition": {
      "StringEqualsIgnoreCase": {
        "aws:ResourceTag/Owner": "${aws:username}"
      }
    }
  }
]
}

```

다음 예제는 데이터세트 생성 및 삭제를 허용하는 정책을 생성하는 방법을 보여줍니다. 이러한 작업은 사용자 이름이 johndoe인 경우에만 허용됩니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "personalize:CreateDataset",
        "personalize>DeleteDataset"
      ],
      "Resource": "arn:aws:personalize:*:*:dataset/*",
      "Condition": {
        "StringEquals": {"aws:username" : "johndoe"}
      }
    },
    {
      "Effect": "Allow",
      "Action": "personalize:DescribeDataset",
      "Resource": "*"
    }
  ]
}

```

태그 기반의 리소스 수준 권한을 정의하면 권한이 즉시 적용됩니다. 즉 리소스를 생성하자마자 더 안전하게 보호할 수 있으며 새 리소스에 태그 사용 적용을 빠르게 시작할 수 있습니다. 리소스 수준 권한을

사용하여 새 리소스 및 기존 리소스와 연결할 수 있는 태그 키와 값을 제어할 수도 있습니다. 자세한 내용은 AWS IAM 사용 설명서의 [태그를 사용한 액세스 제어](#)를 참조하세요.

문제 해결

다음 항목에서는 Amazon Personalize에서 발생할 수 있는 오류 메시지에 대한 일반적인 질문에 대한 답변과 문제 해결 조언을 제공합니다. Amazon Personalize가 사용 사례에 적합한지 판단하는 데 도움이 되는 참조 요약 정보는 [Amazon Personalize 샘플 리포지토리의 Amazon Personalize 치트 시트](#)를 참조하세요.

주제

- [자주 묻는 질문\(FAQ\)](#)
- [오류 메시지](#)

자주 묻는 질문(FAQ)

다음은 Personalize의 데이터 가져오기, 학습, 모델 배포, 추천 및 필터와 관련된 자주 묻는 질문에 대한 답변입니다.

더 많은 질문과 답변은 [Personalize 샘플 리포지토리의 Personalize 치트 시트](#)를 참조하세요.

주제

- [데이터 가져오기 및 관리](#)
- [사용자 지정 솔루션 및 솔루션 버전 생성](#)
- [모델 배포\(사용자 지정 캠페인\)](#)
- [추천](#)
- [추천 필터링](#)

데이터 가져오기 및 관리

대량 데이터는 어떤 형식으로 작성해야 하나요?

대량 데이터는 CSV(쉼표로 구분된 값) 형식이어야 합니다. CSV 파일의 첫 행에는 열 헤더가 포함되어 있어야 합니다. CSV 파일의 열 헤더는 데이터세트에서 생성하는 스키마로 매핑되어야 합니다. 데이터에 ASCII로 인코딩되지 않은 문자가 포함된 경우, CSV 파일은 UTF-8 형식으로 인코딩되어야만 합니다. 헤더를 큰따옴표(")로 묶지 마세요. TIMESTAMP 및 CREATION_TIMESTAMP 데이터는 UNIX epoch 시간 형식이어야 합니다. 데이터 준비에 대한 자세한 내용은 [타임스탬프 데이터](#) 단원을 참조하세요. 스키마에 대한 자세한 내용은 [스키마](#) 단원을 참조하세요.

전체 데이터 형식 지침은 [데이터 형식 지침](#) 단원을 참조하세요. 데이터 형식을 지정하는 방법을 잘 모르겠으면 Amazon 데이터 랭글러 (SageMaker Data Wrangler) 를 사용하여 데이터를 준비할 수 있습니다. 자세한 정보는 [Amazon SageMaker 데이터 랭글러를 사용하여 데이터 준비 및 가져오기](#) 을 참조하세요.

얼마나 많은 학습 데이터가 필요한가요?

모든 사용 사례(도메인 데이터세트 그룹) 및 사용자 지정 레시피에 대해 상호작용 데이터에 다음이 포함되어야 합니다.

- 카탈로그의 항목과 상호 작용하는 사용자의 1,000개 이상의 항목 상호 작용 기록. 이러한 상호작용은 대량 가져오기, 스트리밍된 이벤트 또는 둘 다에서 발생할 수 있습니다.
- 각각 2번 이상의 항목 상호 작용을 갖는 25개 이상의 고유 사용자 ID.

품질 추천의 경우, 각 2번 이상의 항목 상호 작용을 갖는 1,000명 이상의 사용자로부터 50,000건 이상의 항목 상호 작용을 갖는 것이 좋습니다.

빈 항목 상호 작용 데이터 세트로 시작해서 데이터를 충분히 기록했으면 새로 기록된 이벤트만 사용하여 추천자(도메인 데이터 세트 그룹) 또는 사용자 지정 솔루션 버전을 만들 수 있습니다. 일부 레시피와 사용 사례에는 추가 데이터 요구 사항이 있을 수 있습니다. 네트워크 요구 사항에 관한 자세한 내용은 [사용 사례 선택](#) 단원을 참조하세요. 다른 요구 사항에 대한 자세한 내용은 [레시피 선택](#) 단원을 참조하세요.

항목 또는 사용자 속성을 업데이트하려면 어떻게 해야 하나요?

Personalize 콘솔 또는 [PutItems](#) 또는 [PutUsers](#) 작업을 사용하여 아이템 ID는 같지만 속성이 수정된 항목 또는 사용자를 가져올 수 있습니다.

항목 또는 사용자를 삭제하려면 어떻게 해야 하나요?

Personalize는 특정 항목 또는 사용자 삭제를 지원하지 않습니다. 항목이나 사용자가 추천에 나타나지 않도록 하려면 필터를 사용하여 항목을 제외시킵니다. 자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 단원을 참조하세요.

스키마를 삭제하려면 어떻게 해야 하나요?

스키마는 [DeleteSchema](#) 작업을 통해서만 삭제할 수 있습니다. Personalize 콘솔을 사용해서는 스키마를 삭제할 수 없습니다.

사용자 지정 솔루션 및 솔루션 버전 생성

어떤 레시피를 사용해야 하나요?

사용하는 Personalize 레시피는 사용 사례에 따라 다릅니다. 사용 사례를 레시피와 일치시키는 방법에 대한 자세한 내용은 [레시피 선택](#) 단원을 참조하세요. [Personalize 치트 시트](#)에도 사용 사례 및 레시피 정보가 포함되어 있습니다.

얼마나 자주 훈련해야 하나요?

최소 매주 훈련 빈도가 있는 자동 교육을 사용하는 것이 좋습니다. 자동 교육을 통해 권장 사항의 관련성을 더 쉽게 유지할 수 있습니다. 교육 빈도는 비즈니스 요구 사항, 사용하는 레시피, 데이터 가져오기 빈도에 따라 달라집니다. 자세한 정보는 [자동 교육 구성](#)을 참조하세요. 관련성 유지에 대한 자세한 내용은 [추천 관련성 유지](#)를 참조하십시오.

AutoML을 사용해야 하나요?

아니요. 대신 사용 사례를 다른 Personalize 레시피에 맞추고 레시피를 선택하는 것이 좋습니다. 사용 사례를 레시피와 일치시키는 방법에 대한 자세한 내용은 [레시피 선택](#) 단원을 참조하세요.

모델 배포(사용자 지정 캠페인)

캠페인의 최소 프로비저닝 DPS에는 무엇을 설정해야 하나요?

minProvisionedTPS가 높으면 비용이 증가합니다. minProvisionedTPS(기본값)의 경우 1부터 시작하는 것이 좋습니다. Amazon CloudWatch 지표를 사용하여 사용량을 추적하고 minProvisionedTPS 필요에 따라 사용량을 늘리십시오.

캠페인 비용을 모니터링하려면 어떻게 해야 하나요?

Amazon Personalize Monitor 프로젝트는 Amazon Personalize 캠페인을 위한 CloudWatch 대시보드, 사용자 지정 지표, 사용률 경고 및 비용 최적화 기능을 제공합니다. [아마존 퍼스널라이즈 샘플 리포트 토리의 아마존 퍼스널라이즈 모니터](#)를 참조하십시오.

캠페인의 최대 트랜잭션 처리량을 설정하려면 어떻게 해야 하나요?

캠페인의 최소 처리량만 설정할 수 있습니다. Personalize 캠페인을 생성할 때 애플리케이션 사용자를 위한 실시간 추천을 생성하기 위한 전용 트랜잭션 용량을 지정합니다. TPS가 minProvisionedTPS 이상으로 증가할 경우, Personalize는 프로비저닝 용량을 자동으로 확장 및 축소하지만 minProvisionedTPS 이하로는 절대 그렇게 하지 않습니다. 자세한 정보는 [초당 최소 프로비저닝 트랜잭션 수 및 자동 스케일링](#)을 참조하세요.

추천

Personalize 모델이 품질 추천을 생성하고 있는지 어떻게 알 수 있나요?

오프라인 및 온라인 지표([지표를 사용하여 솔루션 버전 평가](#) 참조)와 온라인 테스트(예: A/B 테스트)를 사용하여 솔루션 버전의 성능을 평가합니다. A/B 테스트에 대한 자세한 내용은 [A/B 테스트를 통한 권장 사항 영향 측정](#).

배치 추론 작업을 삭제하려면 어떻게 해야 하며, 상태가 "활성"인 이유는 무엇인가요?

배치 추론 작업은 삭제할 수 없습니다. 배치 추론 작업의 상태가 활성이면 작업이 완료된 것입니다. 출력 S3 버킷 또는 폴더의 추천에 액세스할 수 있습니다. 작업이 완료되면 배치 추론 작업으로 인해 추가 비용이 발생하지 않습니다. 하지만 S3와 같은 다른 서비스에서는 입력 및 출력 데이터 저장에 대해 추가 요금이 부과될 수 있습니다.

SIMS 지원 캠페인에서 메타데이터를 기반으로 유사하지 않은 항목을 추천하는 이유는 무엇인가요?

SIMS는 색상이나 가격 같은 항목 메타데이터가 아닌 항목 상호 작용 데이터 세트를 사용하여 유사성을 판단합니다. SIMS는 상호작용 데이터 세트의 사용자 이력에서 해당 항목이 동시에 발생한 경우를 식별하여 유사한 항목을 추천합니다. 자세한 정보는 [SIMS 레시피](#)를 참조하세요.

단일 GetRecommendations API 작업으로 500개 이상의 항목을 가져올 수 있나요?

500개는 단일 [GetRecommendations](#)에서 검색할 수 있는 최대 항목 수입니다. 이 값은 늘릴 수 없습니다.

추천 필터링

내 추천이 예상대로 필터링되지 않는 이유는 무엇인가요?

이 오류는 여러 가지 이유로 발생할 수 있습니다.

- 필터 표현식의 형식이나 구문에 문제가 있을 수 있습니다. 올바른 형식의 필터 표현식의 예는 [필터 표현식 예제](#) 단원을 참조하세요.
- Personalize는 이벤트 유형별로 사용자당 가장 최근의 상호작용을 최대 100건까지 고려합니다. 이 할당량은 조정 가능한 할당량입니다. [Service Quotas 콘솔](#)을 사용하여 IAM 할당량 증가를 요청할 수 있습니다.

자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 단원을 참조하세요.

이미 구매한 항목을 추천에서 제거하려면 어떻게 해야 하나요?

ECOMMERCE 도메인 데이터세트 그룹의 경우, [추천 제품](#) 또는 [X를 본 고객도 보는 제품](#) 사용 사례를 사용하여 추천자를 생성하면 Personalize는 지정한 사용자 ID 및 Purchase 이벤트를 기반으로 사용자가 구매한 항목을 자동으로 필터링합니다.

다른 도메인 데이터세트 그룹 사용 사례 또는 사용자 지정 리소스의 경우 필터를 사용하여 구매한 항목을 제거합니다. 데이터에 Purchased 이벤트 유형 속성을 추가하고, PutItems 작업을 사용하여 구매 이벤트를 기록하고, 추천에서 구매한 항목을 제거하는 필터를 생성합니다. 예:

```
EXCLUDE ItemID WHERE Interactions.EVENT_TYPE IN ("purchased")
```

자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 단원을 참조하세요.

오류 메시지

다음 단원에서는 Amazon Personalize를 사용할 때 발생할 수 있는 몇 가지 메시지를 나열하고 설명합니다.

주제

- [데이터 가져오기 및 관리](#)
- [솔루션 및 솔루션 버전 생성\(사용자 지정 리소스\)](#)
- [모델 배포\(사용자 지정 캠페인\)](#)
- [추천자\(도메인 데이터 세트 그룹\)](#)
- [추천](#)
- [추천 필터링](#)

데이터 가져오기 및 관리

오류 메시지: 데이터 위치가 잘못되었습니다.

Amazon S3 버킷 위치에 맞는 구문을 사용했는지 확인합니다. 데이터 세트 가져오기 작업의 경우 Amazon S3의 데이터 위치에 대해 다음 구문을 사용합니다.

s3://<name of your S3 bucket>/<folder path>/<CSVfilename>

CSV 파일이 폴더에 있고 하나의 데이터 세트 가져오기 작업으로 여러 파일을 업로드하려면 CSV 파일 이름 없이 이 구문을 사용합니다.

오류 메시지: CreateDatasetImportJob 작업을 호출하는 동안 오류가 발생했습니다 (LimitExceedExceededException). PENDING 또는 IN_PROGRESS 상태인 리소스가 5개가 넘습니다.

지역별로 총 5개의 보류 중이거나 진행 중인 데이터 세트 가져오기 작업을 가질 수 있습니다. 이 할당량은 조정할 수 없습니다. Amazon Personalize의 할당량 목록은 [Personalize 엔드포인트 및 할당량 단원](#)을 참조하세요.

오류 메시지: <데이터 세트 유형> 데이터 세트에 대한 데이터 가져오기 작업을 생성하지 못했습니다... Amazon S3의 데이터에 액세스할 수 있는 권한이 부족합니다.

Amazon S3 버킷과 Amazon Personalize 서비스 역할에 액세스 정책을 연결하여 Amazon S3 리소스에 대한 액세스 권한을 Amazon Personalize에 부여합니다. [Personalize에 S3 리소스에 대한 액세스 권한 부여](#) 단원을 참조하세요.

암호화에 AWS Key Management Service(AWS KMS)를 사용하는 경우 Amazon Personalize와 Amazon Personalize IAM 서비스 역할에 키를 사용할 수 있는 권한을 부여해야 합니다. 자세한 내용은 [Amazon Personalize에 AWS KMS 키 사용 권한 부여](#) 단원을 참조하세요.

오류 메시지: 데이터 가져오기 작업 <데이터 세트 유형> 데이터 세트를 생성하지 못했습니다... 입력 CSV에 다음 열이 누락되었습니다. [COLUMN_NAME, COLUMN_NAME].

속성 이름 및 데이터 유형을 포함하여 Amazon Personalize로 가져오는 데이터는 대상 데이터 세트의 스키마와 일치해야 합니다. 자세한 내용은 [스키마](#) 섹션을 참조하세요.

오류 메시지: <COLUMN_NAME>에 대해 길이가 <문자 한도> 자를 초과할 수 없습니다. 문자 한도를 초과하는 값이 없는 경우 <https://docs.aws.amazon.com/personalize/latest/dg/data-prep-formatting.html>에 나열된 형식 지침을 데이터가 준수하는지 확인합니다.

이 열의 모든 값이 문자 한도를 초과하지 않는지 확인합니다. 문자 한도를 초과하는 값이 없는 경우 앞에 있는 모든 텍스트 필드에서 다음 사항을 확인합니다.

- 모든 텍스트 데이터는 큰따옴표로 묶어야 합니다. 이 \ 문자를 사용하여 데이터에 있는 큰따옴표나 \ 문자를 이스케이프할 수 있습니다.
- CSV 파일의 각 레코드가 한 줄에 있는지 확인합니다.

솔루션 및 솔루션 버전 생성(사용자 지정 리소스)

오류 메시지: 생성에 실패했습니다. 데이터 세트의 사용자가 25명 미만이고, 각 사용자가 최소 2회 이상 상호작용하고 있습니다.

모델을 학습시키려면 먼저 더 많은 데이터를 가져와야 합니다. 모델을 학습시키는 데 필요한 최소 데이터 요구 사항은 다음과 같습니다.

- 카탈로그의 항목과 상호 작용하는 사용자의 1,000개 이상의 항목 상호 작용 기록. 이러한 상호 작용은 대량 가져오기, 스트리밍된 이벤트 또는 이 둘 다에서 발생할 수 있습니다.
- 각각 2번 이상의 항목 상호 작용을 갖는 25개 이상의 고유 사용자 ID.

실시간 추천을 받으려면 데이터 세트 가져오기 작업으로 더 많은 데이터를 가져오거나 이벤트 트래커 및 [PutEvents](#) 작업을 사용하여 사용자에게 대한 더 많은 상호 작용 [이벤트](#)를 기록합니다. 실시간 이벤트 기록에 대한 자세한 내용은 [이벤트 기록](#) 단원을 참조하세요.

배치 추천의 경우 데이터가 더 많을 때 데이터 세트 가져오기 작업을 통해 데이터를 가져옵니다. 대량 데이터 가져오기에 대한 자세한 정보는 [단계 2: 데이터 준비 및 가져오기](#) 단원을 참조하세요.

모델 배포(사용자 지정 캠페인)

오류: 캠페인을 생성할 수 없습니다. ACTIVE 상태인 리소스가 5개 넘게 있습니다. 일부를 삭제하고 다시 시도하세요.

데이터 세트 그룹당 총 5개의 활성 Amazon Personalize 캠페인을 가질 수 있습니다. Service Quotas 콘솔을 사용하여 조정 가능한 IAM 할당량에 대한 할당량 증가를 요청할 수 있습니다. Amazon Personalize의 전체 제한 및 할당량 목록은 [Personalize 엔드포인트 및 할당량](#) 단원을 참조하세요.

추천자(도메인 데이터 세트 그룹)

오류: 이벤트 유형: <이벤트 유형>별로 필터링한 후 데이터 세트의 상호 작용이 1000개 미만입니다

사용 사례에 따라 필요한 이벤트 유형도 다릅니다. 데이터에는 사용 사례에 필요한 유형의 이벤트가 1000개 이상 있어야 합니다. 자세한 내용은 [사용 사례 선택](#) 단원을 참조하세요.

추천

배치 추천 작업 오류 메시지: 잘못된 S3 입력 경로 또는 잘못된 S3 출력 경로

Amazon S3 입력 또는 출력 위치에 올바른 구문을 사용해야 합니다. 또한 출력 위치가 입력 데이터와 달라야 합니다. 동일한 Amazon S3 버킷 또는 다른 버킷에 있는 폴더여야 합니다.

Amazon S3의 입력 파일 위치에 다음 구문을 사용합니다. **s3://<name of your S3 bucket>/<folder name>/<input JSON file name>**

Amazon S3의 출력 폴더에는 다음 구문을 사용합니다. **s3://<name of your S3 bucket>/<output folder name>/**

추천 필터링

오류 메시지: 필터를 생성할 수 없습니다. 잘못된 입력 기호: \$parameterName. NOT_IN 연산자에는 자리 표시자를 사용할 수 없습니다.

NOT_IN 연산자를 사용하는 필터 표현식에는 자리 표시자 파라미터를 사용할 수 없습니다. 대신 IN 연산자를 사용하고 반대의 동작을 사용합니다. 즉, 제외 대신 포함을 사용합니다(또는 그 반대).

예를 들어, INCLUDE ItemID WHERE Items.GENRE NOT IN (\$GENRE)를 사용하려는 경우 EXCLUDE ItemID WHERE Items.GENRE IN (\$GENRE)를 사용하여 동일한 결과를 얻을 수 있습니다.

필터에 대한 자세한 내용은 [필터 표현식 요소](#) 단원을 참조하세요.

오류 메시지: 필터를 생성할 수 없습니다. 잘못된 표현식... 부울 유형 필드를 필터링하는 경우

스키마에서 부울 유형의 값을 사용하여 필터링하는 필터 표현식은 만들 수 없습니다. 부울 값을 기준으로 필터링하려면 필드가 String 유형인 스키마를 사용하고 데이터에 값 True 및 False를 사용합니다. 또는 유형 int 또는 long와 값 0 및 1를 사용할 수도 있습니다.

필터에 대한 자세한 내용은 [필터 표현식 요소](#) 단원을 참조하세요.

AWS CloudFormation를 사용하여 리소스 지정하기

Amazon Personalize는 리소스 및 인프라를 생성하고 관리하는 데 소요되는 시간을 줄일 수 있도록 AWS 리소스를 모델링하고 설정하는 데 도움이 되는 서비스인 AWS CloudFormation과 통합됩니다. 지정할 수 있는 모든 AWS 리소스(예: Amazon Personalize 데이터 세트 그룹)를 설명하는 템플릿을 생성할 수 있습니다. 그러면 AWS CloudFormation에서 이러한 리소스를 프로비저닝하고 구성합니다.

AWS CloudFormation을 사용할 때 템플릿을 재사용하여 Amazon Personalize 리소스를 일관되고 반복적으로 설정할 수 있습니다. 리소스를 한 번 설명한 후 여러 AWS 계정 및 리전에서 동일한 리소스를 반복적으로 프로비저닝할 수 있습니다.

주제

- [Amazon Personalize 및 AWS CloudFormation 템플릿](#)
- [Amazon Personalize 리소스의 예제 AWS CloudFormation 템플릿](#)
- [AWS CloudFormation에 대해 자세히 알아보기](#)

Amazon Personalize 및 AWS CloudFormation 템플릿

Amazon Personalize 및 관련 서비스에 대한 리소스를 프로비저닝하고 구성하려면 [AWS CloudFormation 템플릿](#)을 이해해야만 합니다. 템플릿은 JSON 또는 YAML로 서식 지정된 텍스트 파일입니다. 이 템플릿은 AWS CloudFormation 스택에서 프로비저닝할 리소스에 대해 설명합니다. JSON 또는 YAML에 익숙하지 않은 경우 AWS CloudFormation Designer를 사용하면 AWS CloudFormation 템플릿을 시작하는 데 도움이 됩니다. 자세한 내용은 AWS CloudFormation 사용 설명서에서 [AWS CloudFormation Designer이란 무엇입니까?](#)를 참조하세요.

Amazon Personalize는 AWS CloudFormation에서 데이터 세트, 데이터 세트 그룹, 데이터 세트 가져오기 작업, 스키마 및 솔루션 지정을 지원합니다. 자세한 내용은 AWS CloudFormation 사용 설명서의 [Amazon Personalize 리소스 유형 참조](#)를 참조하세요.

Amazon Personalize 리소스의 예제 AWS CloudFormation 템플릿

다음 AWS CloudFormation 템플릿 예제는 다양한 Amazon Personalize 리소스를 지정하는 방법을 보여줍니다.

주제

- [CreateDatasetGroup](#)

- [CreateDataset](#)
- [CreateSchema](#)
- [CreateSolution](#)

CreateDatasetGroup

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "MyDatasetGroup": {
      "Type": "AWS::Personalize::DatasetGroup",
      "Properties": {
        "Name": "my-dataset-group-name"
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  MyDatasetGroup:
    Type: 'AWS::Personalize::DatasetGroup'
    Properties:
      Name: my-dataset-group-name
```

CreateDataset

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "MyDataset": {
      "Type": "AWS::Personalize::Dataset",
      "Properties": {
```

```

    "Name": "my-dataset-name",
    "DatasetType": "Interactions",
    "DatasetGroupArn": "arn:aws:personalize:us-west-2:123456789012:dataset-
group/dataset-group-name",
    "SchemaArn": "arn:aws:personalize:us-west-2:123456789012:schema/schema-
name",
    "DatasetImportJob": {
      "JobName": "my-import-job-name",
      "DataSource": {
        "DataLocation": "s3://bucket-name/file-name.csv"
      },
      "RoleArn": "arn:aws:iam::123456789012:role/personalize-role"
    }
  }
}
}
}
}
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Resources:
  MyDataset:
    Type: 'AWS::Personalize::Dataset'
    Properties:
      Name: my-dataset-name
      DatasetType: Interactions
      DatasetGroupArn: 'arn:aws:personalize:us-west-2:123456789012:dataset-group/
dataset-group-name'
      SchemaArn: 'arn:aws:personalize:us-west-2:123456789012:schema/schema-name'
      DatasetImportJob:
        JobName: my-import-job-name
        DataSource:
          DataLocation: 's3://bucket-name/file-name.csv'
        RoleArn: 'arn:aws:iam::123456789012:role/personalize-role'

```

CreateSchema

JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",

```

```

"Resources": {
  "MySchema": {
    "Type": "AWS::Personalize::Schema",
    "Properties": {
      "Name": "my-schema-name",
      "Schema": "{\"type\": \"record\", \"name\": \"Interactions\",
        \"namespace\": \"com.amazonaws.personalize.schema\", \"fields\": [ { \"name\":
        \"USER_ID\", \"type\": \"string\" }, { \"name\": \"ITEM_ID\", \"type\": \"string
        \\\" }, { \"name\": \"TIMESTAMP\", \"type\": \"long\"}], \"version\": \"1.0\"}"
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Resources:
  MySchema:
    Type: AWS::Personalize::Schema
    Properties:
      Name: "my-schema-name"
      Schema: >-
        {"type": "record", "name": "Interactions", "namespace":
        "com.amazonaws.personalize.schema", "fields": [ { "name": "USER_ID",
        "type": "string" }, { "name": "ITEM_ID", "type": "string" }, { "name":
        "TIMESTAMP", "type": "long"}], "version": "1.0"}

```

CreateSolution

JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "MySolution": {
      "Type": "AWS::Personalize::Solution",
      "Properties": {
        "Name": "my-solution-name",
        "DatasetGroupArn": "arn:aws:personalize:us-
west-2:123456789012:dataset-group/my-dataset-group-name",
        "RecipeArn": "arn:aws:personalize:::recipe/aws-user-personalization",

```



```
        "SolutionConfig": {
          "EventValueThreshold" : ".05"
        }
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  MySolution:
    Type: 'AWS::Personalize::Solution'
    Properties:
      Name: my-solution-name
      DatasetGroupArn: >-
        arn:aws:personalize:us-west-2:123456789012:dataset-group/my-dataset-group-
name
      RecipeArn: 'arn:aws:personalize:::recipe/aws-user-personalization'
      SolutionConfig:
        EventValueThreshold: '.05'
```

AWS CloudFormation에 대해 자세히 알아보기

AWS CloudFormation에 대한 자세한 내용은 다음 리소스를 참조하세요.

- [AWS CloudFormation](#)
- [AWS CloudFormation 사용 설명서](#)
- [AWS CloudFormation API 참조](#)
- [AWS CloudFormation 명령줄 인터페이스 사용 설명서](#)

Personalize의 보안

클라우드 AWS 보안이 최우선 과제입니다. AWS 고객은 가장 보안에 민감한 조직의 요구 사항을 충족 하도록 구축된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 기업과 기업 간의 AWS 공동 책임입니다. [공동 책임 모델](#)은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 - AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호하는 역할을 합니다. AWS 또한 안전하게 사용할 수 있는 서비스를 제공합니다. Personalize는 데이터 암호화를 사용하여 데이터를 보호합니다. 자세한 내용은 [데이터 암호화](#) 단원을 참조하세요. Personalize에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 [AWS 규정 준수 프로그램의 범위에 속하는 서비스](#)를 참조하세요.
- 클라우드에서의 보안 — 귀하의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 귀하는 귀사의 데이터의 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 Personalize를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목적에 맞게 Personalize를 구성하는 방법을 보여줍니다. 또한 Amazon Personalize 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법도 알아봅니다.

주제

- [Personalize의 데이터 보호](#)
- [Personalize의 Identity and Access Management](#)
- [Personalize의 로깅 및 모니터링](#)
- [Personalize에 대한 규정 준수 검증](#)
- [Personalize의 복원성](#)
- [Personalize의 인프라 보안](#)
- [Personalize 및 인터페이스 VPC 엔드포인트\(AWS PrivateLink\)](#)

Personalize의 데이터 보호

AWS [공동 책임 모델](#) [공동 책임 모델](#) Amazon Personalize의 데이터 보호에 적용됩니다. 이 모델에 설명된 대로 AWS 은 (는) 모두를 실행하는 글로벌 인프라를 보호할 책임이 있습니다. AWS 클라우드사

용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스의 보안 구성과 관리 작업에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 [데이터 프라이버시 FAQ](#)를 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 [AWS 공동 책임 모델 및 GDPR](#) 블로그 게시물을 참조하십시오.

데이터 보호를 위해 AWS 계정 자격 증명을 보호하고 AWS IAM Identity Center OR AWS Identity and Access Management (IAM) 을 사용하여 개별 사용자를 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 멀티 팩터 인증 설정(MFA)을 사용하세요.
- SSL/TLS를 사용하여 리소스와 통신하세요. AWS TLS 1.2는 필수이며 TLS 1.3를 권장합니다.
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다. AWS CloudTrail
- 포함된 모든 기본 보안 제어와 함께 AWS 암호화 솔루션을 사용하십시오 AWS 서비스.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하세요.
- 명령줄 인터페이스 또는 API를 AWS 통해 액세스할 때 FIPS 140-2로 검증된 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용하십시오. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 [Federal Information Processing Standard\(FIPS\) 140-2](#)를 참조하십시오.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 양식 필드에 입력하지 않는 것이 좋습니다. 여기에는 Amazon Personalize 또는 기타 콘솔 AWS CLI, API 또는 AWS 서비스 SDK를 사용하여 작업하는 경우가 포함됩니다. AWS 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 보안 인증 정보를 URL에 포함해서는 안 됩니다.

데이터 암호화

다음 정보는 Personalize에서 데이터 암호화를 사용하여 데이터를 보호하는 방법을 설명합니다.

저장 중 암호화

Amazon Personalize에 저장된 모든 데이터는 Amazon Personalize AWS Key Management Service 관리AWS KMS() 키를 사용하여 저장 중에도 항상 암호화됩니다. 리소스 생성 중에 자체 AWS KMS 키를 제공하면, Personalize에서 키를 사용하여 데이터를 암호화하고 저장합니다. 예를 들어, [CreateDatasetGroup](#)작업 시 AWS KMS ARN을 제공하는 경우 Personalize에서 키를 사용하여 사용자가 해당 데이터세트 그룹에서 생성하는 모든 데이터세트로 가져온 데이터를 암호화하고 저장합니다.

사용자가 Personalize와 Personalize IAM 서비스 역할에 키를 사용할 권한을 부여해야만 합니다. 자세한 내용은 [Amazon Personalize에 AWS KMS 키 사용 권한 부여](#) 단원을 참조하세요.

S3의 데이터 암호화에 대한 내용은 Simple Storage Service 사용 설명서에서 [암호화를 사용하여 데이터 보호](#)를 참조하세요. 자체 AWS KMS 키 관리에 대한 자세한 내용은 AWS Key Management Service 개발자 [안내서의 키 관리](#)를 참조하십시오.

전송 중 암호화

Amazon Personalize는 AWS 인증서와 함께 TLS를 사용하여 다른 서비스로 전송되는 모든 데이터를 암호화합니다. AWS 다른 AWS 서비스와의 모든 통신은 HTTPS를 통해 이루어지며, Amazon Personalize 엔드포인트는 HTTPS를 통한 보안 연결만 지원합니다.

Amazon Personalize는 사용자 계정에서 데이터를 복사하여 내부 AWS 시스템에서 처리합니다. 데이터를 처리할 때 Amazon Personalize는 Amazon Personalize 키 또는 사용자가 제공한 AWS KMS 키를 사용하여 데이터를 암호화합니다. AWS KMS

키 관리

AWS 모든 기본 키를 관리합니다. AWS KMS 소유한 모든 AWS KMS 키를 관리하는 것은 사용자의 책임입니다. 사용자가 Personalize와 Personalize IAM 서비스 역할에 키를 사용할 권한을 부여해야만 합니다. 자세한 정보는 [Amazon Personalize에 AWS KMS 키 사용 권한 부여](#)를 참조하세요.

자체 AWS KMS 키 관리에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 [키 관리](#)를 참조하십시오.

Personalize의 Identity and Access Management

AWS Identity and Access Management (IAM)은 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어할 수 AWS 서비스 있도록 도와줍니다. IAM 관리자는 어떤 사용자가 Personalize 리소스를 사용할 수 있도록 인증(로그인)되고 권한이 부여(권한 있음)될 수 있는지 제어합니다. IAM은 추가 AWS 서비스 비용 없이 사용할 수 있습니다.

주제

- [고객](#)
- [ID를 통한 인증](#)
- [정책을 사용한 액세스 관리](#)
- [IAM으로 Personalize를 사용하는 방법](#)

- [교차 서비스 혼동된 대리인 방지](#)
- [Personalize의 자격 증명 기반 정책 예](#)
- [Personalize 자격 증명 및 액세스 문제 해결](#)

고객

Amazon Personalize에서 수행하는 작업에 따라 사용 방법 AWS Identity and Access Management (IAM) 이 다릅니다.

서비스 사용자 - Personalize 서비스를 사용하여 작업을 수행하는 경우 필요한 자격 증명과 권한을 관리자가 제공합니다. 더 많은 Personalize 특성을 사용하여 작업을 수행하게 되면 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다. Personalize 지원 Personalize의 특성에 액세스할 수 없다면 [Personalize 자격 증명 및 액세스 문제 해결](#) 단원을 참조하세요.

서비스 관리자 - 회사에서 Personalize 리소스를 책임지고 있다면 Personalize에 대한 모든 액세스 권한이 있을 것입니다. 서비스 관리자는 서비스 사용자가 액세스해야 하는 Personalize 특성과 리소스를 결정합니다. 그런 다음, IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여 IAM의 기본 개념을 이해하십시오. 기사에서 Personalize와 함께 IAM을 사용하는 방법에 대해 자세히 알아보려면 [IAM으로 Personalize를 사용하는 방법](#) 단원을 참조하세요.

IAM 관리자 - IAM 관리자라면 Personalize에 대한 액세스 권한을 관리하는 정책을 작성하는 방법을 자세히 알고 싶을 것입니다. IAM에서 사용할 수 있는 Personalize 자격 증명 기반 정책 예제를 보려면 [Personalize의 자격 증명 기반 정책 예](#) 단원을 참조하세요.

ID를 통한 인증

인증은 ID 자격 증명을 AWS 사용하여 로그인하는 방법입니다. IAM 사용자로 인증 (로그인 AWS) 하거나 IAM 역할을 맡아 인증 (로그인) 해야 합니다. AWS 계정 루트 사용자

ID 소스를 통해 제공된 자격 증명을 사용하여 페더레이션 ID로 로그인할 수 있습니다. AWS IAM Identity Center (IAM ID 센터) 사용자, 회사의 싱글 사인온 인증, Google 또는 Facebook 자격 증명, 페더레이션 ID의 예입니다. 페더레이션 ID로 로그인할 때 관리자가 이전에 IAM 역할을 사용하여 ID 페더레이션을 설정했습니다. 페더레이션을 사용하여 액세스하는 경우 AWS 간접적으로 역할을 맡게 됩니다.

사용자 유형에 따라 AWS Management Console 또는 AWS 액세스 포털에 로그인할 수 있습니다. 로그인에 대한 자세한 내용은 AWS 로그인 사용 설명서의 [내 로그인 방법](#)을 참조하십시오. AWS 계정

AWS 프로그래밍 방식으로 액세스하는 경우 자격 증명을 사용하여 요청에 암호화 방식으로 서명할 수 있는 소프트웨어 개발 키트 (SDK) 와 명령줄 인터페이스 (CLI) 를 AWS 제공합니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. 권장 방법을 사용하여 직접 요청에 서명하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 [AWS API 요청 서명](#)을 참조하십시오.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어, AWS 계정의 보안을 강화하기 위해 다단계 인증 (MFA) 을 사용할 것을 권장합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 [다중 인증](#) 및 IAM 사용 설명서의 [AWS에서 다중 인증\(MFA\) 사용](#)을 참조하십시오.

AWS 계정 루트 사용자

계정을 AWS 계정만들 때는 먼저 계정의 모든 AWS 서비스 리소스에 대한 완전한 액세스 권한을 가진 하나의 로그인 ID로 시작합니다. 이 ID를 AWS 계정 루트 사용자라고 하며, 계정을 만들 때 사용한 이메일 주소와 비밀번호로 로그인하여 액세스할 수 있습니다. 일상적인 태스크에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 보안 인증 정보를 보호하고 루트 사용자만 수행할 수 있는 태스크를 수행하는 데 사용하세요. 루트 사용자로 로그인해야 하는 전체 작업 목록은 IAM 사용 설명서의 [루트 사용자 보안 인증이 필요한 작업](#)을 참조하십시오.

페더레이션 자격 증명

가장 좋은 방법은 관리자 액세스가 필요한 사용자를 비롯한 수동 AWS 서비스 사용자가 ID 공급자와의 페더레이션을 사용하여 임시 자격 증명을 사용하여 액세스하도록 하는 것입니다.

페더레이션 ID는 기업 사용자 디렉토리, 웹 ID 공급자, Identity Center 디렉터리의 사용자 또는 ID 소스를 통해 제공된 자격 증명을 사용하여 액세스하는 AWS 서비스 모든 사용자를 말합니다. AWS Directory Service 페더레이션 ID에 AWS 계정 액세스하면 이들이 역할을 맡고 역할은 임시 자격 증명을 제공합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center(을)를 사용하는 것이 좋습니다. IAM Identity Center에서 사용자 및 그룹을 생성하거나 자체 ID 소스의 사용자 및 그룹 집합에 연결하고 동기화하여 모든 사용자 및 애플리케이션에서 사용할 수 있습니다. AWS 계정 IAM Identity Center에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 [IAM Identity Center란 무엇입니까?](#)를 참조하십시오.

IAM 사용자 및 그룹

[IAM 사용자는 단일 사용자](#) 또는 애플리케이션에 대한 특정 권한을 AWS 계정 가진 사용자 내 자격 증명입니다. 가능하면 암호 및 액세스 키와 같은 장기 보안 인증이 있는 IAM 사용자를 생성하는 대신 임시 보안 인증을 사용하는 것이 좋습니다. 하지만 IAM 사용자의 장기 보안 인증이 필요한 특정 사용 사

례가 있는 경우, 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 [장기 보안 인증이 필요한 사용 사례의 경우 정기적으로 액세스 키 교체](#)를 참조하십시오.

[IAM 그룹](#)은 IAM 사용자 컬렉션을 지정하는 자격 증명입니다. 사용자는 그룹으로 로그인할 수 없습니다. 그룹을 사용하여 여러 사용자의 권한을 한 번에 지정할 수 있습니다. 그룹을 사용하면 대규모 사용자 집합의 권한을 더 쉽게 관리할 수 있습니다. 예를 들어, IAMAdmins라는 그룹이 있고 이 그룹에 IAM 리소스를 관리할 권한을 부여할 수 있습니다.

사용자는 역할과 다릅니다. 사용자는 한 사람 또는 애플리케이션과 고유하게 연결되지만, 역할은 해당 역할이 필요한 사람이라면 누구나 수입할 수 있습니다. 사용자는 영구적인 장기 보안 인증 정보를 가지고 있지만, 역할은 임시 보안 인증만 제공합니다. 자세한 내용은 IAM 사용 설명서의 [IAM 사용자를 만들어야 하는 경우\(역할이 아님\)](#)를 참조하십시오.

IAM 역할

[IAM 역할](#)은 특정 권한을 가진 사용자 AWS 계정 내의 자격 증명입니다. IAM 사용자와 유사하지만, 특정 개인과 연결되지 않습니다. 역할을 AWS Management Console [전환하여](#) 에서 일시적으로 IAM 역할을 맡을 수 있습니다. AWS CLI 또는 AWS API 작업을 호출하거나 사용자 지정 URL을 사용하여 역할을 수입할 수 있습니다. 역할 사용 방법에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 역할 사용](#)을 참조하십시오.

임시 보안 인증이 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션 ID에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 페더레이션 ID가 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션 역할에 대한 자세한 내용은 IAM 사용 설명서의 [서드 파티 ID 공급자의 역할 생성](#) 단원을 참조하십시오. IAM Identity Center를 사용하는 경우, 권한 집합을 구성합니다. 인증 후 ID가 액세스할 수 있는 항목을 제어하기 위해 IAM Identity Center는 권한 세트를 IAM의 역할과 연관짓습니다. 권한 세트에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 [권한 세트](#)를 참조하십시오.
- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할은 IAM 역할을 수입하여 특정 태스크에 대한 다양한 권한을 임시로 받을 수 있습니다.
- 크로스 계정 액세스 - IAM 역할을 사용하여 다른 계정의 사용자(신뢰할 수 있는 보안 주체)가 내 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 계정 간 액세스를 부여하는 기본적인 방법입니다. 그러나 일부 AWS 서비스 경우에는 역할을 프록시로 사용하는 대신 정책을 리소스에 직접 연결할 수 있습니다. 계정 간 액세스에 대한 역할과 리소스 기반 정책의 차이점을 알아보려면 [IAM 사용 설명서의 IAM의 교차 계정 리소스 액세스](#)를 참조하십시오.
- 서비스 간 액세스 — 일부는 다른 기능을 사용합니다. AWS 서비스 AWS 서비스예를 들어 서비스에서 직접 호출을 수행하면 일반적으로 해당 서비스는 Amazon EC2에서 애플리케이션을 실행하거

나 Amazon S3에 객체를 저장합니다. 서비스는 직접적으로 호출하는 보안 주체의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 태스크를 수행할 수 있습니다.

- 순방향 액세스 세션 (FAS) — IAM 사용자 또는 역할을 사용하여 작업을 수행하는 경우 보안 AWS 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 전화를 거는 주체의 권한을 다운스트림 AWS 서비스서비스에 AWS 서비스 요청하기 위한 요청과 결합하여 사용합니다. FAS 요청은 다른 서비스 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 서비스가 수신한 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 [전달 액세스 세션](#)을 참조하세요.
- 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 태스크를 수행하기 위해 맡는 [IAM 역할](#)입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [AWS 서비스에 대한 권한을 위임할 역할 생성](#)을 참조하십시오.
- 서비스 연결 역할 — 서비스 연결 역할은 에 연결된 서비스 역할의 한 유형입니다. AWS 서비스서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 사용자에게 AWS 계정 표시되며 해당 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.
- Amazon EC2에서 실행되는 애플리케이션 — IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 API 요청을 AWS CLI 하는 애플리케이션의 임시 자격 증명을 관리할 수 있습니다. AWS 이는 EC2 인스턴스 내에 액세스 키를 저장할 때 권장되는 방법입니다. EC2 인스턴스에 AWS 역할을 할당하고 모든 애플리케이션에서 사용할 수 있게 하려면 인스턴스에 연결된 인스턴스 프로필을 생성합니다. 인스턴스 프로파일에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 보안 인증을 얻을 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여](#)를 참조하십시오.

IAM 역할을 사용할지 또는 IAM 사용자를 사용할지를 알아보려면 [IAM 사용 설명서](#)의 IAM 역할(사용자 대신)을 생성하는 경우를 참조하십시오.

정책을 사용한 액세스 관리

정책을 생성하고 이를 AWS ID 또는 리소스에 AWS 연결하여 액세스를 제어할 수 있습니다. 정책은 ID 또는 리소스와 연결될 때 AWS 해당 권한을 정의하는 객체입니다. AWS 주도자 (사용자, 루트 사용자 또는 역할 세션) 가 요청할 때 이러한 정책을 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는 지를 결정합니다. 대부분의 정책은 JSON 문서로 AWS 저장됩니다. JSON 정책 문서의 구조와 콘텐츠에 대한 자세한 내용은 IAM 사용 설명서의 [JSON 정책 개요](#)를 참조하십시오.

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

기본적으로, 사용자와 역할에는 어떠한 권한도 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 수임할 수 있습니다.

IAM 정책은 작업을 수행하기 위해 사용하는 방법과 상관없이 작업에 대한 권한을 정의합니다. 예를 들어, iam:GetRole 작업을 허용하는 정책이 있다고 가정합니다. 해당 정책을 사용하는 사용자는 AWS Management Console, AWS CLI, 또는 AWS API에서 역할 정보를 가져올 수 있습니다.

보안 인증 기반 정책

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 [IAM 정책 생성](#)을 참조하십시오.

보안 인증 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은 내 여러 사용자, 그룹 및 역할에 연결할 수 있는 독립형 정책입니다. AWS 계정관리형 정책에는 AWS 관리형 정책과 고객 관리형 정책이 포함됩니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 [관리형 정책과 인라인 정책의 선택](#)을 참조하십시오.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우, 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 [보안 주체를 지정](#)해야 합니다. 보안 주체에는 계정, 사용자, 역할, 연동 사용자 등이 포함될 수 있습니다. AWS 서비스

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. IAM의 AWS 관리형 정책은 리소스 기반 정책에 사용할 수 없습니다.

액세스 제어 목록(ACL)

액세스 제어 목록(ACL)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACLs는 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

ACL을 지원하는 서비스의 예로는 아마존 S3와 아마존 VPC가 있습니다. AWS WAF ACL에 대해 자세히 알아보려면 Amazon Simple Storage Service 개발자 가이드의 [ACL\(액세스 제어 목록\) 개요](#)를 참조하십시오.

기타 정책 타입

AWS 일반적이지 않은 추가 정책 유형을 지원합니다. 이러한 정책 타입은 더 일반적인 정책 타입에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 - 권한 경계는 자격 증명 기반 정책에 따라 IAM 엔티티(IAM 사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 개체에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻는 권한은 개체의 보안 인증 기반 정책과 그 권한 경계의 교집합입니다. Principal 필드에서 사용자나 역할을 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 엔티티에 대한 권한 경계](#)를 참조하십시오.
- 서비스 제어 정책 (SCP) - SCP는 조직 또는 조직 단위 (OU)에 대한 최대 권한을 지정하는 JSON 정책입니다. AWS Organizations AWS Organizations 사업체가 소유한 여러 AWS 계정 개를 그룹화하고 중앙에서 관리하는 서비스입니다. 조직에서 모든 기능을 활성화할 경우, 서비스 제어 정책 (SCP)을 임의의 또는 모든 계정에 적용할 수 있습니다. SCP는 구성원 계정의 엔티티 (각 엔티티 포함)에 대한 권한을 제한합니다. AWS 계정 루트 사용자조직 및 SCP에 대한 자세한 내용은 AWS Organizations 사용 설명서의 [SCP 작동 방식](#)을 참조하십시오.
- 세션 정책 - 세션 정책은 역할 또는 페더레이션 사용자에게 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할의 보안 인증 기반 정책의 교차와 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 내용은 IAM 사용 설명서의 [세션 정책](#)을 참조하십시오.

여러 정책 타입

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 여러 정책 유형이 관련된 경우 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 [정책 평가 로직](#)을 참조하십시오.

IAM으로 Personalize를 사용하는 방법

IAM을 사용하여 Personalize에 대한 액세스를 관리하기 전에 Personalize에서 사용할 수 있는 IAM 특성에 대해 알아보십시오.

EMR에서 사용할 수 있는 IAM 특성

IAM 특성	Personalize 지원
ID 기반 정책	예
리소스 기반 정책	아니요
정책 작업	예
정책 리소스	예
정책 조건 키(서비스별)	예
ACLs	아니요
ABAC(정책의 태그)	예
임시 보안 인증	예
보안 주체 권한	예
서비스 역할	예
서비스 연결 역할	아니요

Amazon Personalize 및 AWS 기타 서비스가 대부분의 IAM 기능과 어떻게 작동하는지 자세히 알아보려면 IAM 사용 설명서의 [IAM과 함께 작동하는 서비스를AWS 참조하십시오](#).

Personalize의 자격 증명 기반 정책

보안 인증 기반 정책 지원	예
----------------	---

자격 증명 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 자격 증명에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 [IAM 정책 생성](#)을 참조하십시오.

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. 보안 인증 기반 정책에서는 보안 주체가 연결된 사용자 또는 역할에 적용되므로 보안 주체를 지정할 수 없습니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 [IAM JSON 정책 요소 참조](#)를 참조하십시오.

Personalize 의 자격 증명 기반 정책 예

Personalize 자격 증명 기반 정책 예제를 보려면 [Personalize의 자격 증명 기반 정책 예](#) 단원을 참조하십시오.

Personalize 내의 리소스 기반 정책

리소스 기반 정책 지원	아니요
--------------	-----

리소스 기반 정책은 리소스에 연결하는 JSON 정책 문서입니다. 리소스 기반 정책의 예는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우, 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 [보안 주체를 지정](#)해야 합니다. 보안 주체에는 계정, 사용자, 역할, 연동 사용자 등이 포함될 수 있습니다. AWS 서비스

교차 계정 액세스를 활성화하려는 경우, 전체 계정이나 다른 계정의 IAM 개체를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 리소스 기반 정책에 크로스 계정 보안 주체를 추가하는 것은 트러스트 관계 설정의 절반밖에 되지 않는다는 것을 유념하십시오. 보안 주체와 리소스가 다른 AWS 계정경우 신뢰할 수 있는 계정의 IAM 관리자는 보안 주체 개체 (사용자 또는 역할) 에게 리소스에 액세스할 수 있는 권한도 부여해야 합니다. 엔터티에 ID 기반 정책을 연결하여 권한을 부여합니다. 하지만 리소스 기반 정책이 동일 계정의 보안 주체에 액세스를 부여하는 경우, 추가 자격 증명 기반 정책이 필요하지 않습니다. 자세한 내용은 IAM 사용 설명서의 [IAM의 교차 계정 리소스 액세스](#)를 참조하십시오.

Personalize의 정책 작업

정책 작업 지원	예
----------	---

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

JSON 정책의 Action요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 태스크를 설명합니다. 정책 작업은 일반적으로 관련 AWS API 작업과 이름이 같습니다. 일치하는 API 작업이 없는 권한 전용 작업 같은 몇 가지 예외도 있습니다. 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 일컬어 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하십시오.

Personalize 작업 목록을 보려면 서비스 승인 참조의 [Personalize에서 정의한 작업](#)을 참조하세요.

Personalize의 정책 작업은 작업 앞에 다음 접두사를 사용합니다.

```
personalize
```

단일 문에서 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
  "personalize:action1",
  "personalize:action2"
]
```

와일드카드(*)를 사용하여 여러 작업을 지정할 수 있습니다. 예를 들어, Describe라는 단어로 시작하는 모든 태스크를 지정하려면 다음 태스크를 포함합니다.

```
"Action": "personalize:Describe*"
```

Personalize 자격 증명 기반 정책 예제를 보려면 [Personalize의 자격 증명 기반 정책 예](#)단원을 참조하세요.

Personalize의 정책 리소스

정책 리소스 지원

예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

Resource JSON 정책 요소는 작업이 적용되는 하나 이상의 개체를 지정합니다. 문장에는 Resource또는 NotResource요소가 반드시 추가되어야 합니다. 모범 사례에 따라 [Amazon 리소스](#)

[이름\(ARN\)](#)을 사용하여 리소스를 지정합니다. 리소스 수준 권한이라고 하는 특정 리소스 유형을 지원하는 작업에 대해 이 태스크를 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*"

```

Personalize 리소스 유형 및 해당 ARN의 목록을 보려면 서비스 권한 부여 참조에서 [Personalize 에서 정의한 리소스](#)를 참조하세요. 각 리소스의 ARN을 지정할 수 있는 작업을 알아보려면 [Personalize 에서 정의한 작업](#)을 참조하세요.

Personalize 자격 증명 기반 정책 예제를 보려면 [Personalize의 자격 증명 기반 정책 예](#)단원을 참조하세요.

Personalize에 사용되는 정책 조건 키

서비스별 정책 조건 키 지원	예
-----------------	---

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

Condition 요소(또는 Condition 블록)를 사용하면 정책이 발효되는 조건을 지정할 수 있습니다. Condition 요소는 옵션입니다. 같거나 작음과 같은 [조건 연산자](#)를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다.

한 문에서 여러 Condition 요소를 지정하거나 단일 Condition 요소에서 여러 키를 지정하는 경우, AWS 는 논리적 AND 태스크를 사용하여 평가합니다. 단일 조건 키에 여러 값을 지정하는 경우는 논리적 OR 연산을 사용하여 조건을 AWS 평가합니다. 명문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

조건을 지정할 때 자리 표시자 변수를 사용할 수도 있습니다. 예컨대, IAM 사용자에게 IAM 사용자 이름으로 태그가 지정된 경우에만 리소스에 액세스할 수 있는 권한을 부여할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [IAM 정책 요소: 변수 및 태그](#)를 참조하십시오.

AWS 글로벌 조건 키 및 서비스별 조건 키를 지원합니다. 모든 AWS 글로벌 조건 키를 보려면 IAM 사용 [AWS 설명서의 글로벌 조건 컨텍스트 키](#)를 참조하십시오.

Personalize 조건 키 목록을 보려면 서비스 승인 참조의 [Personalize에 사용되는 조건 키](#)를 참조하세요. 조건 키를 사용할 수 있는 작업과 리소스를 알아보려면 [Personalize에서 정의한 작업](#)을 참조하세요.

Personalize 자격 증명 기반 정책 예제를 보려면 [Personalize의 자격 증명 기반 정책 예](#)단원을 참조하세요.

Personalize의 ACL

ACL 지원	아니요
--------	-----

ACL(액세스 통제 목록)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACLs는 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

Personalize와 ABAC

ABAC 지원(정책의 태그)	예
-----------------	---

ABAC(속성 기반 액세스 제어)는 속성을 기반으로 권한을 정의하는 권한 부여 전략입니다. AWS에서는 이러한 속성을 태그라고 합니다. IAM 엔티티(사용자 또는 역할) 및 여러 AWS 리소스에 태그를 첨부할 수 있습니다. ABAC의 첫 번째 단계로 개체 및 리소스에 태그를 지정합니다. 그런 다음 보안 주체의 태그가 액세스하려는 리소스의 태그와 일치할 때 작업을 허용하도록 ABAC 정책을 설계합니다.

ABAC는 빠르게 성장하는 환경에서 유용하며 정책 관리가 번거로운 상황에 도움이 됩니다.

태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 [조건 요소](#)에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

ABAC에 대한 자세한 정보는 IAM 사용 설명서의 [ABAC란 무엇입니까?](#)를 참조하십시오. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 [속성 기반 액세스 제어\(ABAC\) 사용](#)을 참조하십시오.

Personalize 리소스 태그 지정에 대한 자세한 내용은 [Personalize 리소스에 태그 지정](#) 단원을 참조하세요.

리소스의 태그를 기반으로 리소스에 대한 액세스를 제한하는 자격 증명 기반 정책의 예제는 [IAM 정책에서 태그 사용](#) 섹션에서 확인할 수 있습니다.

Personalize 에서 임시 자격 증명 사용

임시 보안 인증 지원

예

임시 자격 증명을 사용하여 로그인하면 작동하지 AWS 서비스 않는 것도 있습니다. 임시 자격 증명을 사용하는 방법을 AWS 서비스 비롯한 추가 정보는 [IAM 사용 설명서의 IAM과AWS 서비스 연동되는](#) 내용을 참조하십시오.

사용자 이름과 암호를 제외한 다른 방법을 AWS Management Console 사용하여 로그인하면 임시 자격 증명을 사용하는 것입니다. 예를 들어 회사의 SSO (Single Sign-On) 링크를 AWS 사용하여 액세스하는 경우 이 프로세스에서 자동으로 임시 자격 증명을 생성합니다. 또한 콘솔에 사용자로 로그인한 다음 역할을 전환할 때 임시 보안 인증을 자동으로 생성합니다. 역할 전환에 대한 자세한 내용은 IAM 사용 설명서의 [역할로 전환\(콘솔\)](#)을 참조하십시오.

또는 API를 사용하여 임시 자격 증명을 수동으로 생성할 수 있습니다 AWS CLI . AWS 그런 다음 해당 임시 자격 증명을 사용하여 액세스할 수 AWS있습니다. AWS 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성할 것을 권장합니다. 자세한 정보는 [IAM의 임시 보안 자격 증명](#) 섹션을 참조하십시오.

Personalize에 대한 교차 서비스 보안 주체 권한

전달 액세스 세션(FAS) 지원

예

IAM 사용자 또는 역할을 사용하여 작업을 수행하는 AWS 경우 보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 전화를 거는 주체의 권한을 다운스트림 서비스에 AWS 서비스 요청하라는 요청과 결합하여 사용합니다. AWS 서비스 FAS 요청은 다른 서비스 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 서비스가 수신한 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 [전달 액세스 세션](#)을 참조하세요.

Personalize의 서비스 역할

서비스 역할 지원

예

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하는 것으로 가정하는 [IAM 역할](#)입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 [AWS 서비스에 대한 권한을 위임할 역할 생성](#)을 참조하세요.

Warning

서비스 역할에 대한 권한을 변경하면 Personalize 기능이 중단될 수 있습니다. Personalize에서 관련 지침을 제공하는 경우에만 서비스 역할을 편집합니다.

Personalize에 대한 서비스 연결 역할

서비스 연결 역할 지원

아니요

서비스 연결 역할은 에 연결된 서비스 역할의 한 유형입니다. AWS 서비스서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수입할 수 있습니다. 서비스 연결 역할은 사용자에게 AWS 계정 표시되며 해당 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

서비스 연결 역할 생성 또는 관리에 대한 자세한 내용은 [IAM으로 작업하는AWS 서비스](#)를 참조하십시오. 서비스 연결 역할 열에서 Yes(이)가 포함된 서비스를 테이블에서 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 Yes(네) 링크를 선택합니다.

교차 서비스 혼동된 대리인 방지

혼동된 대리자 문제는 작업을 수행할 권한이 없는 엔터티가 권한이 더 많은 엔터티에게 작업을 수행하도록 강요할 수 있는 보안 문제입니다. 에서 AWS크로스 서비스 사칭으로 인해 대리인 문제가 발생할 수 있습니다. 교차 서비스 가장은 한 서비스(호출하는 서비스)가 다른 서비스(호출되는 서비스)를 호출할 때 발생할 수 있습니다. 직접적으로 호출하는 서비스는 다른 고객의 리소스에 대해 액세스 권한이 없는 방식으로 작동하게 권한을 사용하도록 조작될 수 있습니다. 이를 방지하기 위해 AWS에서는 계정의 리소스에 대한 액세스 권한이 부여된 서비스 보안 주체를 사용하여 모든 서비스에 대한 데이터를 보호하는 데 도움이 되는 도구를 제공합니다.

Personalize가 리소스에 다른 서비스를 제공하는 권한을 제한하려면 리소스 정책에서 [aws:SourceArn](#) 및 [aws:SourceAccount](#) 글로벌 조건 컨텍스트 키를 사용하는 것이 좋습니다.

Personalize에서 맡은 역할의 대리인 문제가 혼동되지 않도록 역할의 신뢰 정책에서 값을 `aws:SourceArn~arn:aws:personalize:region:accountNumber:*` 사이로 설정하세요. 와일드카드(*)는 모든 Personalize 리소스에 대한 조건을 적용합니다.

다음 신뢰 관계 정책에서는 Personalize에게 사용자의 리소스에 대한 액세스 권한을 부여하고 혼동된 대리인 문제를 방지하기 위해 `aws:SourceArn` 및 `aws:SourceAccount` 전역 조건 컨텍스트 키를 사용합니다. Personalize([Personalize에 대한 IAM 역할 생성](#))에 대한 역할을 생성할 때는 이 정책을 사용하세요.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "personalize.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "accountNumber"
        },
        "StringLike": {
          "aws:SourceArn": "arn:aws:personalize:region:accountNumber:*"
        }
      }
    }
  ]
}
```

Personalize의 자격 증명 기반 정책 예

기본적으로 사용자 및 역할은 Personalize 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console, AWS Command Line Interface (AWS CLI) 또는 API를 사용하여 작업을 수행할 수도 없습니다. AWS 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 맡을 수 있습니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 [IAM 정책 생성](#)을 참조하십시오.

각 리소스 유형에 대한 ARN 형식을 포함하여 Personalize에서 정의한 작업 및 리소스 유형에 대한 자세한 내용은 서비스 인증 참조에서 [Personalize에 대한 작업, 리소스 및 조건 키](#)를 참조하세요.

주제

- [정책 모범 사례](#)
- [AWS 관리형 정책](#)
- [Personalize 콘솔 사용](#)
- [사용자가 자신의 고유한 권한을 볼 수 있도록 허용](#)
- [Personalize 리소스에 대한 전체 액세스 허용](#)
- [다음 권한은 Personalize 리소스에 대한 읽기 전용 액세스 허용](#)

정책 모범 사례

ID 기반 정책에 따라 계정에서 사용자가 Personalize 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따릅니다.

- AWS 관리형 정책으로 시작하고 최소 권한 권한으로 이동 — 사용자와 워크로드에 권한을 부여하려면 여러 일반적인 사용 사례에 권한을 부여하는 AWS 관리형 정책을 사용하세요. 해당 내용은 에서 사용할 수 있습니다. AWS 계정사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 더 줄이는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 [AWS 관리형 정책](#) 또는 [직무에 대한AWS 관리형 정책](#)을 참조하십시오.
- 최소 권한 적용 – IAM 정책을 사용하여 권한을 설정하는 경우, 태스크를 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서의 [IAM의 정책 및 권한](#)을 참조하십시오.
- IAM 정책의 조건을 사용하여 액세스 추가 제한 – 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어 SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 작성할 수 있습니다. 예를 AWS 서비스들어 특정 작업을 통해 서비스 작업을 사용하는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 AWS CloudFormation있습니다. 자세한 내용은 IAM 사용 설명서의 [IAM JSON 정책 요소: 조건](#)을 참조하십시오.
- IAM Access Analyzer를 통해 IAM 정책을 확인하여 안전하고 기능적인 권한 보장 - IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 신규 및 기존 정책을 확

인합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 추천을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 내용은 IAM 사용 설명서의 [IAM Access Analyzer 정책 검증](#)을 참조하십시오.

- 멀티 팩터 인증 (MFA) 필요 - IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 AWS 계정 MFA를 활성화하십시오. API 작업을 직접적으로 호출할 때 MFA가 필요하다면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 [MFA 보호 API 액세스 구성](#)을 참조하십시오.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 [IAM의 보안 모범 사례](#)를 참조하십시오.

AWS 관리형 정책

AWS 관리형 정책은 에서 생성하고 관리하는 AWS정책입니다. 다음은 Amazon Personalize로 작업할 때 사용할 수 있는 AWS 관리형 정책의 예입니다.

AmazonPersonalizeFullAccess 정책

AWS 관리형 AmazonPersonalizeFullAccess 정책을 사용하여 사용자에게 다음과 같은 권한을 부여할 수 있습니다.

- 모든 Personalize 리소스에 액세스
- Amazon에 지표 게시 및 나열 CloudWatch
- 버킷 이름에 Personalize또는 personalize가 포함된 S3 버킷에서 모든 객체 나열, 읽기, 쓰기, 삭제
- Personalize에 역할 전달

AmazonPersonalizeFullAccess에서 필요 이상으로 많은 권한을 제공합니다. 필요한 권한만 부여하는 새 IAM 정책을 생성하는 것이 좋습니다([Personalize에 리소스에 액세스할 수 있는 권한 부여](#) 참조).

CloudWatchFullAccess

사용자에게 Amazon Personalize를 모니터링할 수 있는 CloudWatch 권한을 부여하려면 CloudWatchFullAccess 정책을 역할에 연결하십시오. 자세한 정보는 [Personalize 모니터링](#)을 참조하세요.

CloudWatchFullAccess 정책은 선택형이며, 다음 작업에 대한 권한을 부여합니다.

- Amazon Personalize 메트릭을 게시하고 나열하십시오. CloudWatch
- 지표 및 지표 통계를 볼 수 있습니다.
- 지표 기반 경보를 설정합니다.

Personalize 콘솔 사용

Personalize 콘솔에 액세스하려면 최소한의 권한 집합이 있어야 합니다. 이러한 권한을 통해 자신의 Amazon Personalize 리소스에 대한 세부 정보를 나열하고 볼 수 있어야 합니다. AWS 계정최소 필수 권한보다 더 제한적인 자격 증명 기반 정책을 만들면 콘솔이 해당 정책에 연결된 엔터티(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

AWS CLI 또는 AWS API만 호출하는 사용자에게 최소 콘솔 권한을 허용할 필요는 없습니다. 그 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

사용자가 자신의 고유한 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 AWS CLI 또는 AWS API를 사용하여 프로그래밍 방식으로 이 작업을 완료할 수 있는 권한이 포함됩니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",

```

```

        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
}

```

Personalize 리소스에 대한 전체 액세스 허용

다음 예제는 AWS 계정의 IAM 사용자에게 모든 Amazon Personalize 리소스 및 작업에 대한 전체 액세스 권한을 부여합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "personalize:*"
      ],
      "Resource": "*"
    }
  ]
}

```

다음 권한은 Personalize 리소스에 대한 읽기 전용 액세스 허용

이 예시에서는 AWS 계정의 IAM 사용자에게 Amazon Personalize 데이터 세트, 데이터 세트 그룹, 솔루션 및 캠페인을 비롯한 Amazon Personalize 리소스에 대한 읽기 전용 액세스 권한을 부여합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [

```

```

    "personalize:DescribeAlgorithm",
    "personalize:DescribeBatchInferenceJob",
    "personalize:DescribeBatchSegmentJob",
    "personalize:DescribeCampaign",
    "personalize:DescribeDataset",
    "personalize:DescribeDatasetExportJob",
    "personalize:DescribeDatasetGroup",
    "personalize:DescribeDatasetImportJob",
    "personalize:DescribeEventTracker",
    "personalize:DescribeFeatureTransformation",
    "personalize:DescribeFilter",
    "personalize:DescribeRecipe",
    "personalize:DescribeRecommender",
    "personalize:DescribeSchema",
    "personalize:DescribeSolution",
    "personalize:DescribeSolutionVersion",
    "personalize:GetSolutionMetrics",
    "personalize:ListBatchInferenceJobs",
    "personalize:ListBatchSegmentJobs",
    "personalize:ListCampaigns",
    "personalize:ListDatasetExportJobs",
    "personalize:ListDatasetGroups",
    "personalize:ListDatasetImportJobs",
    "personalize:ListDatasets",
    "personalize:ListEventTrackers",
    "personalize:ListFilters",
    "personalize:ListRecipes",
    "personalize:ListRecommenders",
    "personalize:ListSchemas",
    "personalize:ListSolutions",
    "personalize:ListSolutionVersions"
  ],
  "Resource": "*"
}
]
}

```

Personalize 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 Personalize 및 IAM으로 작업할 때 발생할 수 있는 일반적인 문제를 진단하고 수정할 수 있습니다.

주제

- [Personalize에서 작업을 수행할 권한이 없음](#)
- [저는 IAM을 수행할 권한이 없습니다. PassRole](#)
- [외부 사용자가 내 Amazon Personalize AWS 계정 리소스에 액세스할 수 있도록 허용하고 싶습니다.](#)

Personalize에서 작업을 수행할 권한이 없음

작업을 수행할 권한이 없다는 오류가 수신되면, 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음 예제 오류는 mateojacksonIAM 사용자가 콘솔을 사용하여 가상 *my-example-widget* 리소스에 대한 세부 정보를 보려고 하지만 가상 *personalize:GetWidget* 권한이 없을 때 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
personalize:GetWidget on resource: my-example-widget
```

이 경우 *personalize:GetWidget* 작업을 사용하여 *my-example-widget* 리소스에 액세스할 수 있도록 mateojackson 사용자 정책을 업데이트해야 합니다.

도움이 필요한 경우 관리자에게 문의하십시오. AWS 관리자는 로그인 자격 증명을 제공한 사람입니다.

저는 IAM을 수행할 권한이 없습니다. PassRole

iam:PassRole 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 Personalize에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

새 서비스 역할 또는 서비스 연결 역할을 만드는 대신 기존 역할을 해당 서비스에 전달할 AWS 서비스 수 있는 기능도 있습니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 있어야 합니다.

다음 예제 오류는 marymajor라는 IAM 사용자가 콘솔을 사용하여 Personalize에서 태스크를 수행하려고 하는 경우에 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 수 있는 권한을 가지고 있지 않습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

이 경우, Mary가 *iam:PassRole* 작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요하면 관리자에게 문의하세요. AWS 관리자는 로그인 자격 증명을 제공한 사람입니다.

외부 사용자가 내 Amazon Personalize AWS 계정 리소스에 액세스할 수 있도록 허용하고 싶습니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록(ACL)을 지원하는 서비스의 경우 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세히 알아보려면 다음을 참조하십시오.

- Personalize에서 이러한 특성을 지원하는지 여부를 알아보려면 [IAM으로 Personalize를 사용하는 방법](#) 단원을 참조하세요.
- 소유하고 AWS 계정 있는 모든 리소스에 대한 액세스 권한을 [AWS 계정 부여하는 방법을 알아보려면 IAM 사용 설명서의 다른 IAM 사용자에게 액세스 권한 제공](#)을 참조하십시오.
- [제3자에게 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사 AWS 계정](#) AWS 계정 소유에 대한 액세스 제공을 참조하십시오.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 [외부에서 인증된 사용자에게 액세스 권한 제공\(자격 증명 페더레이션\)](#)을 참조하십시오.
- 교차 계정 액세스에 대한 역할 사용과 리소스 기반 정책의 차이점을 알아보려면 [IAM 사용 설명서의 IAM의 교차 계정 리소스 액세스](#)를 참조하십시오.

Personalize의 로깅 및 모니터링

이 섹션에서는 Amazon Personalize를 통한 Amazon Personalize 모니터링 및 로깅에 대한 정보를 제공합니다. CloudWatch AWS CloudTrail

주제

- [Personalize 모니터링](#)
- [CloudWatch 아마존 퍼스널라이즈용 메트릭](#)
- [를 사용하여 Amazon Personalize API 호출을 로깅합니다. AWS CloudTrail](#)

Personalize 모니터링

Amazon을 사용하면 Amazon CloudWatch Personalize와 관련된 메트릭을 가져올 수 있습니다. 하나 이상의 지표가 정의한 임계값 범위를 벗어나는 경우 이를 알리도록 경보를 설정할 수 있습니다. 지표를

보려면 Amazon CloudWatch AWS Command Line Interface, [Amazon](#) 또는 [CloudWatch API](#)를 사용할 수 있습니다.

주제

- [Amazon Personalize용 CloudWatch 지표 사용](#)
- [Personalize 지표에 액세스](#)
- [경보 생성](#)
- [Personalize 서버리스 모니터링 앱 예제](#)

Amazon Personalize용 CloudWatch 지표 사용

측정치를 사용하려면 다음 정보를 지정해야 합니다.

- 척도 이름.
- 지표 측정기준. 차원은 지표를 고유하게 식별하는 데 도움이 되는 이름-값 페어입니다.

AWS Management Console AWS CLI, 또는 API를 사용하여 Amazon Personalize에 대한 모니터링 데이터를 가져올 수 있습니다. CloudWatch AWS SDK 또는 CloudWatch API 도구 중 하나를 통해 API를 사용할 수도 있습니다. CloudWatch 콘솔에는 CloudWatch API의 원시 데이터를 기반으로 일련의 그래프가 표시됩니다. 필요에 따라 콘솔에 표시되거나 API에서 가져온 그래프를 사용하는 것이 더 나을 수 있습니다.

다음 목록은 몇 가지 일반적인 지표 사용 사례를 보여 줍니다. 모든 사용 사례를 망라한 것은 아니지만 시작하는 데 참고가 될 것입니다.

방법	관련 측정치
기록된 이벤트의 수를 추적하려면 어떻게 해야 하나요?	PutEventsRequests 지표를 모니터링합니다.
DatasetImportJob 오류를 모니터링하려면 어떻게 해야 하나요?	DatasetImportJobError 측정치를 사용합니다.
GetRecommendations 호출의 지연 시간은 어떻게 모니터링할 수 있습니까?	GetRecommendationsLatency 측정치를 사용합니다.

Amazon Personalize를 모니터링하려면 적절한 CloudWatch 권한이 있어야 합니다. CloudWatch 자세한 내용은 [Amazon의 인증 및 액세스 제어를 참조하십시오 CloudWatch](#).

Personalize 지표에 액세스

다음 예는 CloudWatch 콘솔 AWS CLI, 및 API를 사용하여 Amazon Personalize 지표에 액세스하는 방법을 보여줍니다. CloudWatch

지표를 보려면(콘솔)

1. <https://console.aws.amazon.com/cloudwatch/> 에서 AWS Management Console 로그인하고 CloudWatch 콘솔을 엽니다.
2. [측정치]를 선택하고 [모든 측정치] 탭을 선택한 후 AWS/Personalize를 선택합니다.
3. 지표 차원을 선택합니다.
4. 목록에서 원하는 지표를 선택하고 그래프의 기간을 선택합니다.

일정한 기간 동안 수신된 이벤트의 지표를 보려면(CLI)

- 를 AWS CLI 열고 다음 명령을 입력합니다.

```
aws cloudwatch get-metric-statistics \  
  --metric-name PutEventsRequests \  
  --start-time 2019-03-15T00:00:20Z \  
  --period 3600 \  
  --end-time 2019-03-16T00:00:00Z \  
  --namespace AWS/Personalize \  
  --dimensions Name=EventTrackerArn,Value=EventTrackerArn \  
  --statistics Sum
```

이 예제는 일정한 기간 동안 주어진 이벤트 추적기 ARN에 대해 수신된 이벤트를 보여줍니다. 자세한 내용은 [get-metric-statistics](#)를 참조하십시오.

지표 (CloudWatch API) 에 액세스하려면

- [GetMetricStatistics](#)을 호출합니다. 자세한 내용은 [Amazon CloudWatch API 참조를 참조하십시오](#).

경보 생성

CloudWatch 알람 상태가 변경될 때 Amazon Simple Service (Amazon SNS) 메시지를 보내는 경보를 생성할 수 있습니다. 경보는 지정한 기간 동안 단일 지표를 감시합니다. 이 경보는 여러 기간에 대해 주어진 임계값과 지표 값을 비교하여 하나 이상의 작업을 수행합니다. 이 작업은 SNS 주제 또는 AWS Auto Scaling 정책에 전송되는 알림입니다.

경보는 지속적인 상태 변경에 대한 조치만 호출합니다. CloudWatch 경보가 특정 상태에 있다는 이유만으로 경보가 조치를 호출하지는 않습니다. 상태가 변경되어 지정된 기간 수 동안 유지되어야 합니다.

경보를 설정하려면(콘솔)

1. [에 AWS Management Console 로그인하고 https://console.aws.amazon.com/cloudwatch/ 에서 CloudWatch 콘솔을 엽니다.](https://console.aws.amazon.com/cloudwatch/)
2. 탐색 창에서 경보를 선택한 다음 경보 생성을 선택합니다. 그러면 Create Alarm Wizard가 시작됩니다.
3. 지표 선택을 선택하세요.
4. All metrics 탭에서 AWS/Personalize를 선택합니다.
5. EventTrackerArn을 선택한 다음 PutEvents요청 지표를 선택합니다.
6. 그래프로 표시된 지표 탭을 선택합니다.
7. Statistic(통계)에서 Sum(합계)을 선택합니다.
8. 지표 선택을 선택하세요.
9. [Name]과 [Description]을 입력합니다. 다음 경우 항상에서 >를 선택하고 원하는 최대값을 입력합니다.
10. 경보 상태에 도달했을 때 이메일을 CloudWatch 보내려면 이 경보가 발생할 때마다:에서 상태가 ALARM임을 선택합니다. 기존 SNS 주제에 경보를 전송하려면 다음 주소로 알림 전송:에서 기존 SNS 주제를 선택합니다. 새 이메일 구독 목록의 이름과 이메일 주소를 설정하려면 새 목록을 선택합니다. CloudWatch 목록을 저장하고 필드에 표시하므로 이 목록을 사용하여 향후 경보를 설정할 수 있습니다.

Note

새 목록을 사용하여 새 SNS 주제를 만드는 경우, 의도한 수신자가 알림을 받기 전에 이메일 주소를 확인해야만 합니다. SNS는 경보가 경보 상태에 진입할 때만 이메일을 전송합니다. 이러한 경보 상태 변경이 이메일 주소 확인 전에 발생할 경우, 의도된 수신자는 알림을 받지 못합니다.

11. 경보 생성(Create alarm)을 선택하십시오.

경보를 설정하려면(AWS CLI)

- AWS CLI를 열고 다음 명령을 입력합니다. 이전에 만든 SNS 주제를 참조하도록 `alarm-actions` 파라미터의 값을 변경합니다.

```
aws cloudwatch put-metric-alarm \
  --alarm-name PersonalizeCLI \
  --alarm-description "Alarm when more than 10 events occur" \
  --metric-name PutEventsRequests \
  --namespace AWS/Personalize \
  --statistic Sum \
  --period 300 \
  --threshold 10 \
  --comparison-operator GreaterThanThreshold \
  --evaluation-periods 1 \
  --unit Count \
  --dimensions Name=EventTrackerArn,Value=EventTrackerArn \
  --alarm-actions SNSTopicArn
```

이 예제는 주어진 이벤트 추적기 ARN에 대해 10건 이상의 이벤트가 5분 이내에 발생할 때 경보를 생성하는 방법을 보여줍니다. 자세한 내용은 [put-metric-alarm](#)을 참조하십시오.

알람 설정하기 (CloudWatch API)

- [PutMetricAlarm](#)을 호출합니다. 자세한 내용은 [Amazon CloudWatch API 레퍼런스를](#) 참조하십시오.

Personalize 서버리스 모니터링 앱 예제

Personalize에 모니터링, 경고 및 최적화 기능을 추가하는 예제 앱에 대해서는 [Personalize 샘플 리포지토리의 Personalize](#) 모니터를 참조하세요.

CloudWatch 아마존 퍼스널라이즈용 메트릭

이 섹션에는 Amazon Personalize에서 사용할 수 있는 아마존 CloudWatch 지표에 대한 정보가 포함되어 있습니다. 자세한 정보는 [Personalize 모니터링](#)을 참조하세요.

다음 표에는 Personalize 지표가 나열되어 있습니다. 다음 통계를 제외하고 GetRecommendations GetPersonalizedRanking 지원하는 모든 지표는 다음과 같습니다. Average, Minimum, Maximum, Sum GetRecommendations 그리고 GetPersonalizedRanking Sum 지원만 가능합니다.

지표	설명
DatasetImportJobRequests	성공한 CreateDatasetImportJob API 호출 수입입니다. 차원: DatasetGroupArn, DatasetArn, DatasetImportJobArn
DatasetImportJobError	오류가 발생한 CreateDatasetImportJob API 호출 수입입니다. 차원: DatasetGroupArn, DatasetArn, DatasetImportJobArn
DatasetImportJobExecutionTime	CreateDatasetImportJob API 호출 시간부터 작업 완료(또는 실패) 시간 사이의 간격입니다. 차원: DatasetGroupArn, DatasetArn, DatasetImportJobArn 단위: 초
DatasetSize	데이터세트 가져오기 작업에서 가져오는 데이터의 크기입니다. 차원: DatasetGroupArn, DatasetArn, DatasetImportJobArn 단위: 바이트
SolutionTrainingJobRequests	성공한 CreateSolutionVersion API 호출 수입입니다. 차원: SolutionArn, SolutionVersionArn
SolutionTrainingJobError	오류가 발생한 CreateSolutionVersion API 호출 수입입니다.

지표	설명
	차원: SolutionArn, SolutionVersionArn
SolutionTrainingJobExecutionTime	CreateSolutionVersion API 호출 시간부터 작업 완료(또는 실패) 시간 사이의 간격입니다. 차원: SolutionArn, SolutionVersionArn 단위: 초
GetPersonalizedRanking	GetPersonalizedRanking API 직접 호출 성공 여부. sum통계를 사용하여 성공한 GetPersonalizedRanking API 호출의 총 수를 볼 수 있습니다. 이 지표는 다른 통계를 지원하지 않습니다. 차원: CampaignArn
GetPersonalizedRanking4xxErrors	4xx HTTP 응답 코드를 반환한 GetPersonalizedRanking API 호출 수입니다. 차원: CampaignArn
GetPersonalizedRanking5xxErrors	5xx HTTP 응답 코드를 반환한 GetPersonalizedRanking API 호출 수입니다. 차원: CampaignArn
GetPersonalizedRankingLatency	GetPersonalizedRanking API 호출을 받은 시간부터 추천을 보낸 시간 사이의 간격(4xx 및 5xx 오류 제외)입니다. 차원: CampaignArn 단위: 밀리초

지표	설명
GetRecommendations	<p>GetRecommendations API 직접 호출 성공 여부. sum통계를 사용하여 성공한 GetRecommendations API 호출의 총 수를 볼 수 있습니다. 이 지표는 다른 통계를 지원하지 않습니다.</p> <p>차원: CampaignArn</p>
GetRecommendations4xxErrors	<p>4xx HTTP 응답 코드를 반환한 GetRecommendations API 호출 수입니다.</p> <p>차원: CampaignArn</p>
GetRecommendations5xx 오류	<p>5xx HTTP 응답 코드를 반환한 GetRecommendations API 호출 수입니다.</p> <p>치수: CampaignArn</p>
GetRecommendationsLatency	<p>GetRecommendations API 호출을 받은 시간부터 추천을 보낸 시간 사이의 간격(4xx 및 5xx 오류 제외)입니다.</p> <p>차원: CampaignArn</p> <p>단위: 밀리초</p>
PutEventsRequests	<p>성공한 PutEvents API 호출 수입니다.</p> <p>차원: DatasetGroupArn, DatasetArn, EventTrackerArn</p>
PutEvents4xxErrors	<p>4xx HTTP 응답 코드를 반환한 PutEvents API 호출 수입니다.</p> <p>차원: DatasetGroupArn, DatasetArn, EventTrackerArn</p>

지표	설명
PutEvents5xxErrors	5xx HTTP 응답 코드를 반환한 PutEvents API 호출 수입니다. 차원: DatasetGroupArn, DatasetArn, EventTrackerArn
PutEventLatency	PutEvents API 호출을 완료하는 데 걸린 시간(4xx 및 5xx 오류 제외)입니다. 차원: DatasetGroupArn, DatasetArn, EventTrackerArn 단위: 밀리초
PutItemsRequests	성공한 PutItems API 호출 수입니다. 차원: DatasetGroupArn, DatasetArn
PutItems4xxErrors	4xx HTTP 응답 코드를 반환한 PutItems API 호출 수입니다. 차원: DatasetGroupArn, DatasetArn
PutItems5xxErrors	5xx HTTP 응답 코드를 반환한 PutItems API 호출 수입니다. 차원: DatasetGroupArn, DatasetArn
PutItemsLatency	PutItems API 호출을 완료하는 데 걸린 시간(4xx 및 5xx 오류 제외)입니다. 차원: DatasetGroupArn, DatasetArn 단위: 밀리초
PutUsersRequests	성공한 PutUsers API 호출 수입니다. 차원: DatasetGroupArn, DatasetArn

지표	설명
PutUsers4xxErrors	4xx HTTP 응답 코드를 반환한 PutUsersAPI 호출 수입니다. 차원: DatasetGroupArn, DatasetArn
PutUsers5xxErrors	5xx HTTP 응답 코드를 반환한 PutUsersAPI 호출 수입니다. 차원: DatasetGroupArn, DatasetArn
PutUsersLatency	PutUsers API 호출을 완료하는 데 걸린 시간(4xx 및 5xx 오류 제외)입니다. 차원: DatasetGroupArn, DatasetArn 단위: 밀리초

를 사용하여 Amazon Personalize API 호출을 로깅합니다. AWS CloudTrail

Amazon Personalize는 Amazon Personalize에서 사용자, 역할 또는 서비스가 수행한 작업의 기록을 제공하는 AWS 서비스와 통합되어 있습니다. AWS CloudTrail CloudTrail Amazon Personalize에 대한 API 호출의 하위 집합을 이벤트로 캡처합니다. 여기에는 Amazon Personalize 콘솔에서의 호출 및 Amazon Personalize API에 대한 코드 호출이 포함됩니다. 트레일을 생성하면 Amazon Personalize용 CloudTrail 이벤트를 포함하여 Amazon S3 버킷으로 이벤트를 지속적으로 전송할 수 있습니다. 트레일을 구성하지 않아도 CloudTrail 콘솔의 이벤트 기록에서 가장 최근 이벤트를 계속 볼 수 있습니다. 에서 수집한 CloudTrail 정보를 사용하여 Amazon Personalize에 이루어진 요청, 요청이 이루어진 IP 주소, 요청한 사람, 요청 시기 및 추가 세부 정보를 확인할 수 있습니다.

구성 및 활성화 방법을 CloudTrail 포함하여 자세한 내용은 사용 [AWS CloudTrail 설명서를](#) 참조하십시오.

Amazon은 다음과 같은 정보를 개인화합니다. CloudTrail

CloudTrail 계정을 생성하면 AWS 계정에서 활성화됩니다. Amazon Personalize에서 지원되는 이벤트 활동이 발생하면 해당 활동이 CloudTrail 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 이벤트에 기록됩니다. AWS 계정에서 최근 이벤트를 보고, 검색하고, 다운로드할 수 있습니다. 자세한 내용은 이벤트 [기록으로 CloudTrail 이벤트 보기를](#) 참조하십시오.

Amazon Personalize의 이벤트를 포함하여 AWS 계정의 지속적인 이벤트 기록을 보려면 트레일을 생성하십시오. 트레일을 사용하면 CloudTrail Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 추적을 생성하면 기본적으로 모든 리전에 추적이 적용됩니다. 트레일은 AWS 파티션에 있는 모든 지역의 이벤트를 기록하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 이에 따라 조치를 취하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음을 참조하세요.

- [추적 생성 개요](#)
- [CloudTrail 지원되는 서비스 및 통합](#)
- [CloudTrail에 대한 Amazon SNS 알림 구성](#)
- [여러 지역에서 CloudTrail 로그 파일 받기 및 여러 계정에서 CloudTrail 로그 파일 받기](#)

Amazon Personalize는 모든 작업 (API 작업) 을 CloudTrail 로그 파일에 이벤트로 기록할 수 있도록 지원합니다. 자세한 정보는 [작업](#)을 참조하세요.

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에게 대한 정보가 들어 있습니다. 보안 인증 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트로 했는지 아니면 사용자 자격 증명으로 했는지 여부
- 역할 또는 페더레이션 사용자에게 대한 임시 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 요청이 다른 AWS 서비스에 의해 이루어졌는지 여부.

자세한 내용은 [CloudTrail UserIdentity](#) 요소를 참조하십시오.

예: CloudTrail 로그 파일 항목

트레일은 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 전송할 수 있는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함되어 있습니다. 이벤트는 모든 소스의 단일 요청을 나타내며 요청된 작업, 작업 날짜 및 시간, 요청 매개 변수 등에 대한 정보를 포함합니다. CloudTrail 로그 파일은 공개 API 호출의 정렬된 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

다음 예제는 ListDatasetGroups API 작업에 대한 작업이 포함된 CloudTrail 로그 항목을 보여줍니다. 참고로 ListDatasetGroups API 작업은 상태를 변경하지 않는 작업이므로 responseElements 응답은 null입니다. 레코드 본문에 대한 자세한 내용은 CloudTrail [CloudTrail 레코드 콘텐츠를](#) 참조하십시오.

```
{
  "eventVersion": "1.05",
```

```

"userIdentity": {
  "type": "IAMUser",
  "principalId": "principal-id",
  "arn": "arn:aws:iam::user-arn",
  "accountId": "account-id",
  "accessKeyId": "access-key",
  "userName": "user-name"
},
"eventTime": "2018-11-22T02:18:03Z",
"eventSource": "personalize.amazonaws.com",
"eventName": "ListDatasetGroups",
"awsRegion": "us-west-2",
"sourceIPAddress": "source-ip-address",
"userAgent": "aws-cli/1.11.16 Python/2.7.11 Darwin/15.6.0 botocore/1.4.73",
"requestParameters": null,
"responseElements": null,
"requestID": "request-id",
"eventID": "event-id",
"eventType": "AwsApiCall",
"recipientAccountId": "recipient-account-id"
}

```

Personalize에 대한 규정 준수 검증

타사 감사자는 여러 규정 준수 프로그램의 일환으로 Amazon Personalize의 보안 및 규정 준수를 평가합니다. AWS 여기에는 SOC, PCI, HIPAA 등이 포함됩니다.

특정 규정 준수 프로그램 범위 내 AWS 서비스 목록은 규정 준수 프로그램별 범위 [내 서비스 규정 준수 프로그램별](#) 참조하십시오. 일반 정보는 [AWS 규정 준수 프로그램](#) [AWS 보증 프로그램](#) [규정](#) [AWS](#) 참조하십시오.

를 사용하여 AWS Artifact 타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 보고서 <https://docs.aws.amazon.com/artifact/latest/ug/downloading-documents.html> 참조하십시오 AWS Artifact.

Personalize 사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 결정됩니다. AWS 는 규정 준수를 지원할 다음과 같은 리소스를 제공합니다.

- [보안 및 규정 준수 킷스타트 가이드](#) - 이 배포 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수에 중점을 둔 기본 환경을 배포하기 위한 단계를 제공합니다. AWS
- [HIPAA 보안 및 규정 준수를 위한 설계 백서](#) — 미국 건강 보험 이전 및 책임법 (HIPAA) 에 따라 규제 되는 민감한 워크로드를 실행하는 AWS 데 사용할 수 있는 방법을 알아보십시오.

- [AWS 규정 준수 리소스 규정](#) — 이 통합 문서 및 가이드 모음은 해당 산업 및 지역에 적용될 수 있습니다.
- AWS Config 개발자 안내서의 [규칙을 사용하여 리소스 평가](#) — 이 AWS Config 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.
- [AWS Security Hub](#) — 이 AWS 서비스는 보안 업계 표준 및 모범 사례를 준수하는지 확인하는 데 도움이 되는 내부 보안 상태를 종합적으로 보여줍니다.

Personalize의 복원성

AWS 글로벌 인프라는 지역 및 가용 AWS 영역을 중심으로 구축됩니다. AWS 지역은 물리적으로 분리되고 격리된 여러 가용 영역을 제공하며, 이러한 가용 영역은 지연 시간이 짧고 처리량이 높으며 중복성이 높은 네트워크로 연결됩니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

Amazon Personalize는 데이터 복원성을 위해 AWS 글로벌 인프라를 활용합니다. AWS 지역에 Amazon Personalize 리소스를 생성하면 Amazon Personalize는 여러 가용 영역에서 리소스의 복원력과 데이터 중복성을 관리합니다. Amazon Personalize 리소스를 생성할 수 있는 AWS 지역 목록은 Amazon Web Services 일반 참조의 [AWS 지역 및 엔드포인트](#)를 참조하십시오. AWS 지역 및 가용 영역에 대한 자세한 내용은 [AWS 글로벌 인프라](#)를 참조하십시오.

Personalize의 인프라 보안

관리형 서비스인 Amazon Personalize는 AWS 글로벌 네트워크 보안의 보호를 받습니다. AWS 보안 서비스 및 인프라 AWS 보호 방법에 대한 자세한 내용은 [AWS 클라우드 보안을](#) 참조하십시오. 인프라 보안 모범 사례를 사용하여 AWS 환경을 설계하려면 Security Pillar AWS Well-Architected Framework의 [인프라 보호](#)를 참조하십시오.

AWS 게시된 API 호출을 사용하여 네트워크를 통해 Amazon Personalize에 액세스할 수 있습니다. 고객은 다음을 지원해야 합니다.

- 전송 계층 보안(TLS) TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군 Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 요청은 액세스 키 ID 및 IAM 주체와 관련된 비밀 액세스 키를 사용하여 서명해야 합니다. 또는 [AWS Security Token Service\(AWS STS\)](#)를 사용하여 임시 보안 인증을 생성하여 요청에 서명할 수 있습니다.

Personalize 및 인터페이스 VPC 엔드포인트(AWS PrivateLink)

Amazon VPC (Virtual Private Cloud) 를 사용하여 AWS 리소스를 호스팅하는 경우, VPC와 Amazon Personalize 사이에 프라이빗 연결을 설정할 수 있습니다. 이 연결을 사용하면 Personalize가 퍼블릭 인터넷을 통하지 않고 VPC의 리소스와 통신하게 할 수 있습니다.

Amazon VPC는 사용자가 AWS 서비스 정의한 가상 사설 클라우드 (VPC) 또는 가상 네트워크에서 AWS 리소스를 시작하는 데 사용합니다. VPC가 있으면 IP 주소 범위, 서브넷, 라우팅 테이블, 네트워크 게이트웨이 등 네트워크 설정을 제어할 수 있습니다. VPC 엔드포인트를 사용하면 AWS 네트워크가 VPC와 VPC 간의 라우팅을 처리합니다. AWS 서비스

VPC를 Personalize에 연결하려면 Personalize에 대해 인터페이스 VPC 엔드포인트를 정의하세요. 인터페이스 엔드포인트는 지원되는 AWS 서비스로 전달되는 트래픽에 대한 진입점 역할을 하는 프라이빗 IP 주소를 포함하는 탄력적 네트워크 인터페이스입니다. 엔드포인트는 Personalize에 안정적이고 확장 가능한 연결을 제공합니다. 인터넷 게이트웨이, NAT(네트워크 주소 변환) 인스턴스 또는 VPN 연결이 필요 없습니다. 자세한 내용은 VPC 사용 설명서의 [VPC란 무엇입니까?](#) 단원을 참조하세요.

인터페이스 VPC 엔드포인트는 를 통해 활성화됩니다. AWS PrivateLink이 AWS 기술은 사설 IP 주소가 있는 Elastic Network 인터페이스를 사용하여 사설 간의 AWS 서비스 사설 통신을 가능하게 합니다.

Note

모든 Amazon Personalize 연방 정보 처리 표준 (FIPS) 엔드포인트는 에서 지원됩니다. AWS PrivateLink

주제

- [Personalize에 대한 인터페이스 VPC 엔드포인트 생성](#)
- [ECS에 대한 VPC 엔드포인트 정책 생성](#)

Personalize에 대한 인터페이스 VPC 엔드포인트 생성

Amazon VPC 콘솔 또는 () 를 사용하여 Amazon Personalize 서비스를 위한 VPC 엔드포인트를 생성할 수 있습니다. AWS Command Line Interface AWS CLI 자세한 내용은 Amazon VPC 사용 설명서의 [인터페이스 VPC 엔드포인트를 사용한 AWS 서비스 액세스](#)를 참조하십시오.

Personalize의 VPC 엔드포인트를 생성하려면 다음 중 하나를 선택합니다.

- com.amazonaws. ##.개인화
- com.amazonaws. ##.개인화-이벤트
- com.amazonaws. ##.개인화-런타임

엔드포인트에 프라이빗 DNS를 사용하도록 설정하는 경우, 리전에 대한 기본 DNS 이름(예: personalize.us-east-1.amazonaws.com)을 사용하여 Personalize에 API 요청을 할 수 있습니다.

ECS에 대한 VPC 엔드포인트 정책 생성

Personalize에 대한 액세스를 제어하는 VPC 엔드포인트에 엔드포인트 정책을 연결할 수 있습니다. 이 정책은 다음 정보를 지정합니다.

- 작업을 수행할 수 있는 보안 주체.
- 수행할 수 있는 작업.
- 작업을 수행할 수 있는 리소스.

자세한 정보는 VPC 사용 설명서의 [VPC 엔드포인트를 통해 서비스에 대한 액세스 제어](#)를 참조하세요.

예제: Personalize 작업 및 PassRole 작업을 허용하는 VPC 엔드포인트

이 정책은 엔드포인트에 연결될 때 모든 Personalize 작업 및 PassRole 작업에 액세스 권한을 부여합니다.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
```

```
        "personalize:*",
        "iam:PassRole"
    ],
    "Resource": "*"
}
]
```

예: Amazon Personalize 작업을 허용하는 VPC 엔드포인트 정책 ListDatasets

엔드포인트에 연결되면 이 정책은 나열된 Amazon Personalize ListDatasets 작업에 대한 액세스 권한을 부여합니다.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "personalize:ListDatasets"
      ],
      "Resource": "*"
    }
  ]
}
```


Personalize 엔드포인트 및 할당량

다음 단원에는 Personalize 지침, 할당량 및 엔드포인트에 관한 정보가 포함되어 있습니다. 조정 가능한 할당량의 경우 [Service Quotas 콘솔](#)을 사용하여 할당량 증가를 요청할 수 있습니다. 자세한 내용은 [할당량 증가 요청](#) 단원을 참조하세요.

주제

- [Personalize 엔드포인트 및 리전](#)
- [규정 준수](#)
- [Service quotas](#)
- [할당량 증가 요청](#)

Personalize 엔드포인트 및 리전

리전별 Personalize 엔드포인트 목록은 Web Services 일반 참조의 [AWS 리전 및 엔드포인트](#) 단원을 참조하세요.

규정 준수

Personalize 규정 준수 프로그램에 대한 자세한 내용은 [AWS 규정 준수](#), [AWS 규정 준수 프로그램](#) 및 [AWS 규정 준수 프로그램 범위의 서비스](#) 단원을 참조하세요.

Service quotas

귀하의 AWS 계정에는 다음과 같은 Amazon Personalize 할당량이 있습니다.

Resource	할당량
Item interactions	
솔루션 버전 또는 추천자를 생성하는 데 필요한 최소 고유 항목 상호 작용 횟수 사용자 지정 솔루션의 경우 훈련 전에 이벤트 유형이나 이벤트 값을 기준으로 필터링한 후 이 만큼의 레코드가 있어야 합니다.	1000

Resource	할당량
사용자 개인화-v2 및 개인화 순위-v2 레시피의 경우 훈련 중에 모델이 고려하는 항목 상호 작용의 최대 수입입니다.	30억
사용자 개인화-v2 또는 개인화 순위-v2를 제외한 모든 도메인 사용 사례 및 사용자 지정 레시피의 경우 학습 중에 모델이 고려하는 항목 상호 작용의 최대 수입입니다.	5억 (조정 가능)
항목 상호작용 데이터세트에 있는 개별 이벤트 유형의 최대 수와 선택적 메타데이터 열의 총 개수를 합한 값입니다.	10
항목 상호 작용 데이터세트에 있는 최대 메타데이터 열 수 (예약된 필드 제외)	5
범주형 데이터 및 노출 값의 최대 문자 수	1000
전체 가져오기 모드의 데이터세트 가져오기 작업당 대량 항목 상호 작용 데이터의 최대 양	100GB (모델에서 고려한 항목 상호 작용이 증가하면 1TB로 증가)
중분 가져오기 모드를 사용하는 데이터세트 가져오기 작업당 대량 항목 상호 작용 데이터의 최대 양	1GB
전체 또는 중분 가져오기 모드의 데이터세트 가져오기 작업당 항목 상호 작용 레코드 최소 수입입니다.	1000
Users	
도메인 추천자 또는 사용자 지정 솔루션 버전을 생성하려면 항목 상호 작용 데이터의 최소 고유 사용자 수 (항목 상호 작용이 각각 최소 2개 이상)가 필요합니다.	25

Resource	할당량
도메인 추천자 또는 사용자 지정 솔루션 버전을 생성하려면 최소 2개 이상의 항목 상호 작용이 있어야 전체 사용자 중 최소 비율입니다.	1%
사용자 데이터셋의 최대 메타데이터 필드 수입니다.	25
USER_ID 데이터 값의 최대 문자 수입니다.	256
범주형 데이터 값의 최대 문자 수입니다.	1000자
전체 가져오기 모드에서 데이터세트 가져오기 작업당 대량 사용자 데이터의 최대 양	100GB
중분 가져오기 모드의 데이터세트 가져오기 작업당 최대 대량 사용자 데이터 양	1GB
Items	
사용자 개인화-v2 또는 개인화 순위-v2의 경우 훈련 중에 모델이 고려하는 최대 항목 수입니다. 이러한 항목은 항목 및 항목 상호 작용 데이터세트 모두에서 가져온 것입니다.	500만
사용자 맞춤화-v2 및 개인화 순위-v2를 제외한 모든 도메인 사용 사례 및 사용자 지정 레시피의 경우, 학습 및 권장 사항 생성 중에 모델이 고려하는 최대 항목 수입니다.	750,000
항목 데이터세트의 최대 메타데이터 필드 수.	100
ITEM_ID 데이터 값의 최대 문자 수입니다.	256
범주형 데이터 값의 최대 문자 수입니다.	1000자
항목 데이터셋의 최대 텍스트 필드 수입니다.	1

Resource	할당량
중국어 및 일본어 텍스트 데이터 값의 최대 문자 수입입니다.	7,000자
다른 모든 언어의 텍스트 데이터 값의 최대 문자 수입입니다.	20,000자
대량 가져오기 모드의 데이터세트 가져오기 작업당 최대 대량 항목 데이터 양	100GB
중분 가져오기 모드의 데이터세트 가져오기 작업당 최대 대량 항목 데이터 양	1GB
Actions	
권장 사항을 학습시키고 생성하는 동안 모델이 고려하는 최대 조치 수입입니다.	1000
Actions 데이터세트의 최대 메타데이터 필드 수입입니다.	10
ACTION_ID 데이터 값의 최대 문자 수입입니다.	256
범주형 데이터 값의 최대 문자 수입입니다.	1000자
대량 가져오기 모드의 데이터세트 가져오기 작업당 최대 대량 작업 데이터 양	100GB
중분 가져오기 모드의 데이터세트 가져오기 작업당 최대 대량 작업 데이터 양	1GB
Action interactions	
학습 중에 모델이 고려하는 동작 상호 작용의 최대 수입입니다.	5억 개
Action 상호 작용 데이터세트에 있는 최대 메타데이터 열 수 (예약된 필드 제외)	5

Resource	할당량
전체 가져오기 모드의 데이터세트 가져오기 작업당 대량 상호작용 데이터 최대 양	100GB (모델에서 고려한 작업 항목 상호 작용이 증가하면 1TB로 증가)
중분 가져오기 모드를 사용한 데이터세트 가져오기 작업당 대량 상호 작용 데이터의 최대 양	1GB
Individual record import APIs	
데이터세트 그룹당 최대 PutEvents 요청 속도.	1000/초
PutEvents 호출의 최대 이벤트 수	10
이벤트의 최대 크기	10KB
데이터세트 그룹당 최대 PutAction Interactions 요청 속도.	1000/초
통화당 액션 인터랙션 이벤트의 최대 수입입니다. PutActionInteractions	10
액션 인터랙션 이벤트의 최대 크기.	10KB
데이터세트 그룹당 최대 PutItems 요청 비율.	10/초
통화 시 사용할 수 있는 PutItems 최대 항목 수.	10
데이터세트 그룹당 최대 PutUsers 요청 속도.	10/초
통화당 최대 사용자 수. PutUsers	10
데이터세트 그룹당 최대 PutActions 요청 속도.	10/초
통화당 최대 사용자 수. PutActions	10
Legacy recipes	

Resource	할당량
HRNN-메타데이터 및 HRNN-ColdStart 레시피에 대한 사용자 및 항목 데이터셋의 결합된 최대 데이터 양입니다.	5GB
HRNN-ColdStart 레시피가 모델 학습 (솔루션 버전 생성) 을 위해 지원하는 콜드 스타트 항목의 최대 수입니다.	8만 달러
HRNN-ColdStart 레시피에 따라 모델을 학습시키는 데 필요한 최소 콜드 스타트 항목 수 (솔루션 버전 생성).	100
Filters	
데이터세트 그룹당 총 필터 개수.	10
필터에 사용할 수 있는 고유 데이터셋 필드의 최대 개수.	5
데이터셋 그룹의 모든 필터에 걸친 개별 데이터셋 필드의 총 개수.	10
필터에서 고려하는 이벤트 유형별 사용자당 최대 항목 상호 작용 수입니다.	상호작용 100개 (조정 가능)
필터에서 고려한 이벤트 유형별 사용자당 최대 동작 상호 작용 수입니다.	300개의 액션 인터랙션 (조정 가능)
GetRecommendations / GetPersonalizedRanking / GetActionRecommendations requests	
GetActionRecommendations 및 GetPersonalizedRanking 요청에 대한 GetRecommendations 최대 거래 속도.	초당 2500개
캠페인마다 초당 최대 GetRecommendations 요청 수	초당 500개

Resource	할당량
캠페인마다 초당 최대 GetAction Recommendations 요청 수	초당 500개
캠페인마다 초당 최대 GetPersonalizedRanking 요청 수	500/초
GetRecommendations 또는 GetPersonalizedRanking 요청당 최대 메타데이터 열 수	10
메타데이터가 없는 GetRecommendation 요청의 최대 추천 결과 수입니다.	500
메타데이터가 포함된 GetRecommendation 요청의 최대 추천 결과 수입니다.	50
메타데이터를 제외한 GetPersonalizedRanking 요청에서 순위를 매길 수 있는 최대 항목 수입니다.	500
메타데이터가 포함된 GetPersonalizedRanking 요청에서 순위를 매길 수 있는 최대 항목 수입니다.	50
Metric attribution quotas	
지표 어트리뷰션을 위한 최대 지표 수	10
고유 이벤트 어트리뷰션 소스의 최대 수	100
Batch inference jobs	
일괄 추론 작업에 사용할 수 있는 최대 입력 파일 수	1000
배치 추론 작업 입력의 최대 크기	1GB

Resource	할당량
테마가 없는 일괄 추론 작업의 입력 파일당 최대 레코드 수	5천만
테마가 있는 일괄 추론 작업의 입력 파일당 최대 레코드 수입니다.	100
Batch segment jobs	
배치 세그먼트 작업의 최대 입력 파일 수입니다.	1000
배치 세그먼트 작업 입력의 최대 크기.	1GB
항목 선호도 레시피의 입력 파일당 최대 쿼리 수입니다.	500
항목-속성-선호도 레시피의 입력 파일당 최대 쿼리 수입니다.	10
세그먼트당 최대 사용자 수	500만
Data deletion jobs	
PENDING 상태인 데이터세트 그룹의 최대 데이터 삭제 작업 수입니다.	5(조정 가능)
데이터 삭제 입력 파일 또는 파일의 최대 총 크기	50MB

AWS 계정에는 지역별로 다음과 같은 할당량이 있습니다.

Resource	할당량
총 활성 스키마 수.	500
총 활성 데이터세트 그룹 수.	5(조정 가능)
보류 중이거나 진행 중인 데이터세트 가져오기 작업의 총 수.	5
보류 중이거나 진행 중인 배치 추론 작업의 총 수.	5(조정 가능)

Resource	할당량
보류 중이거나 진행 중인 배치 세그먼트 작업의 총 수.	5
보류 중이거나 진행 중인 솔루션 버전의 총 수.	20(조정 가능)

각 데이터세트 그룹에는 다음과 같은 할당량이 있습니다.

Resource	할당량
총 활성 솔루션 수.	10(조정 가능)
총 활성 캠페인 수.	5(조정 가능)
총 추천자 수.	5
총 필터 수.	10(조정 가능)
모든 필터 간 개별 데이터세트 필드의 총 수.	10
PENDING 상태인 데이터세트 그룹의 총 데이터 삭제 작업 수입니다.	5(조정 가능)

할당량 증가 요청

조정 가능한 할당량의 경우 [Service Quotas 콘솔](#)을 사용하여 할당량 증가를 요청할 수 있습니다. 다음 Personalize 할당량은 조정이 가능합니다.

- 교육 중 하나의 모델에서 고려되는 최대 항목 상호 작용 수.
- 데이터세트 그룹별 활성 캠페인
- 활성 데이터세트 그룹
- 데이터세트 그룹당 활성 필터
- 데이터세트 그룹당 활성 솔루션
- 증분 가져오기당 데이터 양
- 필터에서 고려하는 이벤트 유형별 사용자당 최대 항목 상호 작용 수
- 보류 중이거나 진행 중인 배치 추론 작업의 총 수
- PENDING 상태인 데이터세트 그룹의 총 데이터 삭제 작업 수입니다.

- 보류 중이거나 진행 중인 솔루션 버전의 총 수
- 최대 PutEvents 또는 PutActionInteraction 요청 속도

[Service Quotas 콘솔](#)을 사용하여 할당량 증가를 요청하려면 Service Quotas 사용 설명서의 [할당량 증가 요청](#) 단원을 참조하세요.

API 참조

이 단원에서는 Personalize API 작업에 대한 설명서를 제공합니다. Personalize 엔드포인트 목록은 AWS 일반 참조의 [AWS 리전 및 엔드포인트](#)를 참조하세요.

주제

- [작업](#)
- [데이터 타입](#)
- [일반적인 오류](#)
- [공통 파라미터](#)

작업

Personalize에서 지원하는 작업은 다음과 같습니다.

- [CreateBatchInferenceJob](#)
- [CreateBatchSegmentJob](#)
- [CreateCampaign](#)
- [CreateDataDeletionJob](#)
- [CreateDataset](#)
- [CreateDatasetExportJob](#)
- [CreateDatasetGroup](#)
- [CreateDatasetImportJob](#)
- [CreateEventTracker](#)
- [CreateFilter](#)
- [CreateMetricAttribution](#)
- [CreateRecommender](#)
- [CreateSchema](#)
- [CreateSolution](#)
- [CreateSolutionVersion](#)
- [DeleteCampaign](#)
- [DeleteDataset](#)

- [DeleteDatasetGroup](#)
- [DeleteEventTracker](#)
- [DeleteFilter](#)
- [DeleteMetricAttribution](#)
- [DeleteRecommender](#)
- [DeleteSchema](#)
- [DeleteSolution](#)
- [DescribeAlgorithm](#)
- [DescribeBatchInferenceJob](#)
- [DescribeBatchSegmentJob](#)
- [DescribeCampaign](#)
- [DescribeDataDeletionJob](#)
- [DescribeDataset](#)
- [DescribeDatasetExportJob](#)
- [DescribeDatasetGroup](#)
- [DescribeDatasetImportJob](#)
- [DescribeEventTracker](#)
- [DescribeFeatureTransformation](#)
- [DescribeFilter](#)
- [DescribeMetricAttribution](#)
- [DescribeRecipe](#)
- [DescribeRecommender](#)
- [DescribeSchema](#)
- [DescribeSolution](#)
- [DescribeSolutionVersion](#)
- [GetSolutionMetrics](#)
- [ListBatchInferenceJobs](#)
- [ListBatchSegmentJobs](#)
- [ListCampaigns](#)
- [ListDataDeletionJobs](#)

- [ListDatasetExportJobs](#)
- [ListDatasetGroups](#)
- [ListDatasetImportJobs](#)
- [ListDatasets](#)
- [ListEventTrackers](#)
- [ListFilters](#)
- [ListMetricAttributionMetrics](#)
- [ListMetricAttributions](#)
- [ListRecipes](#)
- [ListRecommenders](#)
- [ListSchemas](#)
- [ListSolutions](#)
- [ListSolutionVersions](#)
- [ListTagsForResource](#)
- [StartRecommender](#)
- [StopRecommender](#)
- [StopSolutionVersionCreation](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateCampaign](#)
- [UpdateDataset](#)
- [UpdateMetricAttribution](#)
- [UpdateRecommender](#)

Personalize 이벤트에서 지원하는 작업은 다음과 같습니다.

- [PutActionInteractions](#)
- [PutActions](#)
- [PutEvents](#)
- [PutItems](#)
- [PutUsers](#)

Personalize 런타임에서 지원하는 작업은 다음과 같습니다.

- [GetActionRecommendations](#)
- [GetPersonalizedRanking](#)
- [GetRecommendations](#)

Personalize

Personalize에서 지원하는 작업은 다음과 같습니다.

- [CreateBatchInferenceJob](#)
- [CreateBatchSegmentJob](#)
- [CreateCampaign](#)
- [CreateDataDeletionJob](#)
- [CreateDataset](#)
- [CreateDatasetExportJob](#)
- [CreateDatasetGroup](#)
- [CreateDatasetImportJob](#)
- [CreateEventTracker](#)
- [CreateFilter](#)
- [CreateMetricAttribution](#)
- [CreateRecommender](#)
- [CreateSchema](#)
- [CreateSolution](#)
- [CreateSolutionVersion](#)
- [DeleteCampaign](#)
- [DeleteDataset](#)
- [DeleteDatasetGroup](#)
- [DeleteEventTracker](#)
- [DeleteFilter](#)
- [DeleteMetricAttribution](#)
- [DeleteRecommender](#)

- [DeleteSchema](#)
- [DeleteSolution](#)
- [DescribeAlgorithm](#)
- [DescribeBatchInferenceJob](#)
- [DescribeBatchSegmentJob](#)
- [DescribeCampaign](#)
- [DescribeDataDeletionJob](#)
- [DescribeDataset](#)
- [DescribeDatasetExportJob](#)
- [DescribeDatasetGroup](#)
- [DescribeDatasetImportJob](#)
- [DescribeEventTracker](#)
- [DescribeFeatureTransformation](#)
- [DescribeFilter](#)
- [DescribeMetricAttribution](#)
- [DescribeRecipe](#)
- [DescribeRecommender](#)
- [DescribeSchema](#)
- [DescribeSolution](#)
- [DescribeSolutionVersion](#)
- [GetSolutionMetrics](#)
- [ListBatchInferenceJobs](#)
- [ListBatchSegmentJobs](#)
- [ListCampaigns](#)
- [ListDataDeletionJobs](#)
- [ListDatasetExportJobs](#)
- [ListDatasetGroups](#)
- [ListDatasetImportJobs](#)
- [ListDatasets](#)
- [ListEventTrackers](#)

- [ListFilters](#)
- [ListMetricAttributionMetrics](#)
- [ListMetricAttributions](#)
- [ListRecipes](#)
- [ListRecommenders](#)
- [ListSchemas](#)
- [ListSolutions](#)
- [ListSolutionVersions](#)
- [ListTagsForResource](#)
- [StartRecommender](#)
- [StopRecommender](#)
- [StopSolutionVersionCreation](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateCampaign](#)
- [UpdateDataset](#)
- [UpdateMetricAttribution](#)
- [UpdateRecommender](#)

CreateBatchInferenceJob

서비스: Amazon Personalize

Amazon S3에 저장된 항목 또는 사용자 목록을 기반으로 배치 추천을 생성하고 Amazon S3 버킷으로 추천을 내보냅니다.

배치 추천을 생성하려면 솔루션 버전의 ARN과 입력 및 출력 데이터에 대한 Amazon S3 URI를 지정합니다. 사용자 개인 맞춤, 인기 항목, 개인 맞춤형 순위 솔루션의 경우 배치 추천 작업은 입력 파일의 각 사용자 ID에 대한 추천 항목 목록을 생성합니다. 관련 항목 솔루션의 경우 작업은 입력 파일의 각 항목 ID에 대한 추천 항목 목록을 생성합니다.

자세한 내용은 [배치 추천 작업 생성](#) 단원을 참조하세요.

유사-항목 레시피를 사용하는 경우 Amazon Personalize는 설명 테마를 배치 추천에 추가할 수 있습니다. 테마를 생성하려면 작업 모드를 THEME_GENERATION으로 설정하고 입력 데이터에 항목 이름이 포함된 필드의 이름을 지정하세요.

테마 생성에 대한 자세한 내용은 [콘텐츠 생성기에서 테마가 있는 배치 추천](#)을 참조하세요.

지금-유행 또는 차선책-작업 레시피로는 배치 추천을 받을 수 없습니다.

구문 요청

```
{
  "batchInferenceJobConfig": {
    "itemExplorationConfig": {
      "string" : "string"
    }
  },
  "batchInferenceJobMode": "string",
  "filterArn": "string",
  "jobInput": {
    "s3DataSource": {
      "kmsKeyArn": "string",
      "path": "string"
    }
  },
  "jobName": "string",
  "jobOutput": {
    "s3DataDestination": {
      "kmsKeyArn": "string",
      "path": "string"
    }
  }
}
```

```

    }
  },
  "numResults": number,
  "roleArn": "string",
  "solutionVersionArn": "string",
  "tags": [
    {
      "tagKey": "string",
      "tagValue": "string"
    }
  ],
  "themeGenerationConfig": {
    "fieldsForThemeGeneration": {
      "itemName": "string"
    }
  }
}

```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

batchInferenceJobConfig

배치 추론 작업의 구성 세부 정보입니다.

유형: [BatchInferenceJobConfig](#) 객체

필수 항목 여부: 아니요

batchInferenceJobMode

배치 추론 작업의 모드. 유사한 항목 그룹을 설명하는 테마를 생성하려면 작업 모드를 THEME_GENERATION으로 설정합니다. 테마를 생성하지 않으려면 기본 BATCH_INFERENCE를 사용하세요.

테마가 있는 배치 추천을 받으면 추가 비용이 발생합니다. 자세한 내용은 [Amazon Personalize 요금](#)을 참조하세요.

타입: 문자열

유효 값: BATCH_INFERENCE | THEME_GENERATION

필수 여부: 아니요

filterArn

배치 추론 작업에 적용할 필터의 ARN입니다. 자세한 내용은 [배치 추천 필터링](#) 단원을 참조하세요.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

jobInput

추천의 기반이 되는 입력 파일로 연결되는 S3 경로. 입력 구성 요소는 JSON 형식이어야 합니다.

유형: [BatchInferenceJobInput](#) 객체

필수 여부: 예

jobName

생성할 배치 추론 작업의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

필수 사항 여부: Yes

jobOutput

작업의 출력이 저장되는 S3 버킷의 경로.

유형: [BatchInferenceJobOutput](#) 객체

필수 여부: 예

numResults

검색할 추천 수.

유형: 정수

필수 항목 여부: 아니요

roleArn

자격 증명 및 액세스 관리 역할의 ARN으로, 입력 및 출력 S3 버킷에 대한 읽기 및 쓰기 권한을 각각 가지고 있습니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.\@-_/]+`

필수 사항 여부: Yes

solutionVersionArn

배치 추론 추천을 생성하는 데 사용할 솔루션 버전의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

필수 사항 여부: Yes

tags

배치 추론 작업에 적용할 [태그](#) 목록입니다.

유형: [Tag](#) 객체 어레이

어레이 멤버: 최소 항목 수 0개. 최대 항목 수 200개.

필수 여부: 아니요

themeGenerationConfig

테마 생성 작업의 경우 각 항목의 이름이 포함된 항목 데이터 세트의 열 이름을 지정하세요.

유형: [ThemeGenerationConfig](#) 객체

필수 항목 여부: 아니요

응답 구문

```
{
```

```
"batchInferenceJobArn": "string"  
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

batchInferenceJobArn

배치 추론 작업의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

LimitExceededException

초당 요청 수 한도를 초과했습니다.

HTTP 상태 코드: 400

ResourceAlreadyExistsException

지정된 리소스가 이미 있습니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

TooManyTagsException

이 리소스에 적용할 수 있는 최대 태그 수를 초과했습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

CreateBatchSegmentJob

서비스: Amazon Personalize

배치 세그먼트 작업을 생성합니다. 이 작업은 최대 5천만 개의 레코드를 처리할 수 있으며 입력 파일은 JSON 형식이어야 합니다. 자세한 내용은 [배치 추천 및 사용자 세그먼트 받기](#) 단원을 참조하세요.

구문 요청

```
{
  "filterArn": "string",
  "jobInput": {
    "s3DataSource": {
      "kmsKeyArn": "string",
      "path": "string"
    }
  },
  "jobName": "string",
  "jobOutput": {
    "s3DataDestination": {
      "kmsKeyArn": "string",
      "path": "string"
    }
  },
  "numResults": number,
  "roleArn": "string",
  "solutionVersionArn": "string",
  "tags": [
    {
      "tagKey": "string",
      "tagValue": "string"
    }
  ]
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[filterArn](#)

배치 세그먼트 작업에 적용할 필터의 ARN입니다. 자세한 내용은 [배치 추천 필터링](#) 단원을 참조하세요.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

jobInput

배치 세그먼트 작업을 생성하는 데 사용된 입력 데이터의 S3 경로.

유형: [BatchSegmentJobInput](#) 객체

필수 여부: 예

jobName

생성할 배치 세그먼트 작업의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

필수 사항 여부: Yes

jobOutput

작업의 출력을 저장할 버킷의 S3 경로.

유형: [BatchSegmentJobOutput](#) 객체

필수 여부: 예

numResults

입력 데이터의 각 라인에 대해 배치 세그먼트 작업으로 생성된 예상 사용자 수. 세그먼트당 최대 사용자 수는 5백만 명입니다.

유형: 정수

필수 항목 여부: 아니요

roleArn

자격 증명 및 액세스 관리 역할의 ARN으로, 입력 및 출력 S3 버킷에 대한 읽기 및 쓰기 권한을 각각 가지고 있습니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.\@-_/]+`

필수 사항 여부: Yes

solutionVersionArn

배치 세그먼트 작업을 사용하여 배치 세그먼트를 생성하는 데 사용하려는 솔루션 버전의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

필수 사항 여부: Yes

tags

배치 세그먼트 작업에 적용할 [태그](#) 목록입니다.

유형: [Tag](#) 객체 어레이

어레이 멤버: 최소 항목 수 0개. 최대 항목 수 200개.

필수 여부: 아니요

응답 구문

```
{
  "batchSegmentJobArn": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

batchSegmentJobArn

배치 세그먼트 작업의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*+

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

LimitExceededException

초당 요청 수 한도를 초과했습니다.

HTTP 상태 코드: 400

ResourceAlreadyExistsException

지정된 리소스가 이미 있습니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

TooManyTagsException

이 리소스에 적용할 수 있는 최대 태그 수를 초과했습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

CreateCampaign

서비스: Amazon Personalize

Important

캠페인이 활성화되어 있는 동안에는 캠페인 비용이 발생합니다. 불필요한 비용이 발생하지 않도록 하려면 완료 후 캠페인을 삭제하세요. 캠페인 비용에 대한 자세한 내용은 [Amazon Personalize](#) 가격을 참조하십시오.

솔루션 버전을 배포하는 캠페인을 생성합니다. 클라이언트가 [GetRecommendations](#) 및 [GetPersonalizedRanking](#) API를 호출하면 요청에 캠페인이 지정됩니다.

프로비저닝된 최소 TPS 및 자동 크기 조정

Important

minProvisionedTPS가 높으면 비용이 증가합니다. minProvisionedTPS(기본값)의 경우 1부터 시작하는 것이 좋습니다. Amazon CloudWatch 지표를 사용하여 사용량을 추적하고 minProvisionedTPS 필요에 따라 사용량을 늘리십시오.

Amazon Personalize 캠페인을 생성할 때 캠페인에 대한 초당 최소 프로비저닝 트랜잭션 수 (minProvisionedTPS)를 지정할 수 있습니다. 이는 Amazon Personalize에서 프로비저닝한 캠페인의 기존 트랜잭션 처리량입니다. 캠페인이 활성화 상태일 때의 최소 청구 요금을 설정합니다. 트랜잭션은 단일 GetRecommendations 또는 GetPersonalizedRanking 요청으로 이루어집니다. 기본 minProvisionedTPS 값은 1입니다.

TPS가 minProvisionedTPS 이상으로 증가할 경우 Amazon Personalize는 프로비저닝 용량을 자동으로 확장 및 축소하지만, minProvisionedTPS 이하로는 확장 및 축소하지 않습니다. 용량이 증가 중에 잠시 동안의 지연으로 거래 손실이 발생할 수 있습니다. 트래픽이 감소하면 용량은 minProvisionedTPS로 돌아옵니다.

프로비저닝된 최소 TPS에 대한 요금이 부과되며, 요청이 minProvisionedTPS를 초과하는 경우 실제 TPS에 대한 요금이 부과됩니다. 실제 TPS는 추천을 요청한 총 건수입니다. 낮은 minProvisionedTPS 수치부터 시작하여 Amazon CloudWatch 측정치를 사용하여 사용량을 추적한 다음 minProvisionedTPS 필요에 따라 늘리는 것이 좋습니다.

캠페인 비용에 대한 자세한 내용은 [Amazon Personalize 요금](#)을 참조하세요.

상태

캠페인은 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

캠페인 상태를 알아보려면 전화하십시오 [DescribeCampaign](#).

Note

캠페인의 status이 ACTIVE가 될 때까지 기다렸다가 캠페인에 추천을 요청합니다.

관련 API

- [ListCampaigns](#)
- [DescribeCampaign](#)
- [UpdateCampaign](#)
- [DeleteCampaign](#)

구문 요청

```
{
  "campaignConfig": {
    "enableMetadataWithRecommendations": boolean,
    "itemExplorationConfig": {
      "string" : "string"
    },
    "syncWithLatestSolutionVersion": boolean
  },
  "minProvisionedTPS": number,
  "name": "string",
  "solutionVersionArn": "string",
  "tags": [
    {
      "tagKey": "string",
      "tagValue": "string"
    }
  ]
}
```

```
]
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[campaignConfig](#)

캠페인의 구성 세부 정보입니다.

유형: [CampaignConfig](#) 객체

필수 여부: 아니요

[minProvisionedTPS](#)

Personalize에서 지원할 초당 요청된 최소 프로비저닝 트랜잭션(추천)을 지정합니다. minProvisionedTPS가 높으면 요금이 인상됩니다. minProvisionedTPS(기본값)의 경우 1부터 시작하는 것이 좋습니다. Amazon CloudWatch 지표를 사용하여 사용량을 추적하고 minProvisionedTPS 필요에 따라 사용량을 늘리십시오.

타입: 정수

유효 범위: 최소값 1.

필수 여부: 아니요

[name](#)

새 캠페인의 이름. 캠페인 이름은 계정 내에서 고유해야 합니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

필수 사항 여부: Yes

[solutionVersionArn](#)

캠페인과 함께 배포할 학습된 모델의 Amazon 리소스 이름 (ARN). 솔루션의 최신 솔루션 버전을 지정하려면 솔루션의 ARN을 형식으로 `SolutionArn/$LATEST` 지정하십시오.

True에서 `syncWithLatestSolutionVersion` 로 설정한 경우 이 형식을 사용해야 합니다.

[CampaignConfig](#)

솔루션의 최신 솔루션 버전이 아닌 모델을 배포하려면 솔루션 버전의 ARN을 지정하세요.

자동 캠페인 업데이트에 대한 자세한 내용은 [자동 캠페인 업데이트 활성화](#)를 참조하십시오.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

필수 사항 여부: Yes

[tags](#)

캠페인에 적용할 [태그](#)의 목록입니다.

유형: [Tag](#)객체 어레이

어레이 멤버: 최소 항목 수 0개. 최대 항목 수 200개.

필수 여부: 아니요

응답 구문

```
{
  "campaignArn": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[campaignArn](#)

캠페인의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

LimitExceededException

초당 요청 수 한도를 초과했습니다.

HTTP 상태 코드: 400

ResourceAlreadyExistsException

지정된 리소스가 이미 있습니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

TooManyTagsException

이 리소스에 적용할 수 있는 최대 태그 수를 초과했습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

CreateDataDeletionJob

서비스: Amazon Personalize

Amazon Personalize 데이터 세트 그룹에서 특정 사용자에게 대한 모든 참조를 일괄적으로 삭제하는 배치 작업을 생성합니다. Amazon S3 버킷의 사용자 ID로 구성된 CSV 파일에서 삭제할 사용자를 지정합니다. 작업이 완료되면 Amazon Personalize는 더 이상 사용자 데이터를 기반으로 학습하지 않으며 사용자 세그먼트를 생성할 때 더 이상 사용자를 고려하지 않습니다. [데이터 삭제 작업 생성에 대한 자세한 내용은 사용자 삭제를 참조하십시오.](#)

- 입력 파일은 사용자 ID가 나열된 단일 USER_ID 열이 있는 CSV 파일이어야 합니다. CSV 파일 준비에 대한 자세한 내용은 [데이터 삭제 파일 준비 및 Amazon S3에 업로드](#)를 참조하십시오.
- Amazon Personalize에 사용자 ID의 입력 CSV 파일에 액세스할 수 있는 권한을 부여하려면 데이터 소스에서 읽을 권한이 있는 IAM 서비스 역할을 지정해야 합니다. 이 역할에는 버킷 GetObject 및 ListBucket 해당 콘텐츠에 대한 권한이 필요합니다. 이러한 권한은 데이터를 가져오는 것과 동일합니다. S3 버킷에 대한 액세스 권한의 부여에 대한 자세한 내용은 [S3 리소스에 대한 액세스 권한 Personalize에 부여](#) 단원을 참조하세요.

작업을 생성한 후 데이터세트와 모델에서 사용자에게 대한 모든 참조를 삭제하는 데 최대 하루가 걸릴 수 있습니다. 작업이 완료될 때까지 Amazon Personalize는 교육 시 데이터를 계속 사용합니다. 또한 사용자 세분화 레시피를 사용하면 사용자가 사용자 세그먼트에 나타날 수 있습니다.

상태

데이터 삭제 작업의 상태는 다음 중 하나일 수 있습니다.

- [보류 중] > [진행 중] > [완료] 또는 [실패]

데이터 삭제 작업의 상태를 가져오려면 [DescribeDataDeletionJob](#) API 작업을 호출하고 작업의 Amazon 리소스 이름 (ARN) 을 지정하십시오. 상태가 FAILE인 경우 응답에는 작업이 실패한 이유를 설명하는 failureReason 키가 포함됩니다.

관련 API

- [ListDataDeletionJobs](#)
- [DescribeDataDeletionJob](#)

구문 요청

```
{
  "datasetGroupArn": "string",
  "dataSource": {
    "dataLocation": "string"
  },
  "jobName": "string",
  "roleArn": "string",
  "tags": [
    {
      "tagKey": "string",
      "tagValue": "string"
    }
  ]
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

datasetGroupArn

레코드를 삭제하려는 데이터세트가 있는 데이터세트 그룹의 Amazon 리소스 이름 (ARN)

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]):personalize:.*:.*:.*

필수 사항 여부: Yes

dataSource

삭제할 사용자의 사용자 ID 목록이 들어 있는 Amazon S3 버킷입니다.

유형: [DataSource](#) 객체

필수 여부: 예

jobName

데이터 삭제 작업의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

필수 사항 여부: Yes

[roleArn](#)

Amazon S3 데이터 소스에서 읽을 권한이 있는 IAM 역할의 Amazon 리소스 이름 (ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.\@-_/\+]`

필수 사항 여부: Yes

[tags](#)

데이터 삭제 [작업에 적용할 태그](#) 목록.

유형: [Tag](#) 객체 어레이

어레이 멤버: 최소 항목 수 0개. 최대 항목 수 200개.

필수 여부: 아니요

응답 구문

```
{
  "dataDeletionJobArn": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[dataDeletionJobArn](#)

데이터 삭제 작업의 Amazon 리소스 이름 (ARN)

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

LimitExceededException

초당 요청 수 한도를 초과했습니다.

HTTP 상태 코드: 400

ResourceAlreadyExistsException

지정된 리소스가 이미 있습니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

TooManyTagsException

이 리소스에 적용할 수 있는 최대 태그 수를 초과했습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

CreateDataset

서비스: Amazon Personalize

빈 데이터세트를 생성하고 지정된 데이터세트 그룹에 추가합니다. 훈련 데이터를 데이터세트로 가져오는 [CreateDatasetImportJob](#)에 사용합니다.

다음과 같은 5가지 유형의 데이터 세트가 있습니다.

- 항목 상호 작용
- Items
- 사용자
- 작업 상호 작용
- 작업

각 데이터세트 유형에는 필수 필드 유형과 연결된 스키마가 있습니다. 솔루션 생성이라고도 하는 모델 교육을 위해서는 Item interactions 데이터세트만 필요합니다.

데이터세트는 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

데이터세트의 상태를 확인하려면 `를 호출하세요. DescribeDataset`

관련 API

- [CreateDatasetGroup](#)
- [ListDatasets](#)
- [DescribeDataset](#)
- [DeleteDataset](#)

구문 요청

```
{
  "datasetGroupArn": "string",
  "datasetType": "string",
  "name": "string",
```

```

  "schemaArn": "string",
  "tags": [
    {
      "tagKey": "string",
      "tagValue": "string"
    }
  ]
}

```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

datasetGroupArn

데이터세트를 추가할 데이터세트 그룹의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*+

필수 사항 여부: Yes

datasetType

데이터세트의 형식입니다.

다음 값 중 하나(대소문자 구분 안 함):

- 상호작용
- Items
- 사용자
- 작업
- Action_Interactions

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 예

name

데이터세트의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

필수 사항 여부: Yes

schemaArn

데이터세트와 연결할 스키마의 ARN입니다. 스키마가 데이터세트 필드를 정의합니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

필수 사항 여부: Yes

tags

데이터세트에 적용할 [태그](#)의 목록입니다.

유형: [Tag](#) 객체 어레이

어레이 멤버: 최소 항목 수 0개. 최대 항목 수 200개.

필수 여부: 아니요

응답 구문

```
{
  "datasetArn": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

datasetArn

데이터세트의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

LimitExceededException

초당 요청 수 한도를 초과했습니다.

HTTP 상태 코드: 400

ResourceAlreadyExistsException

지정된 리소스가 이미 있습니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

TooManyTagsException

이 리소스에 적용할 수 있는 최대 태그 수를 초과했습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

CreateDatasetExportJob

서비스: Amazon Personalize

데이터세트의 데이터를 S3 버킷으로 내보내는 작업을 생성합니다. Personalize에서 학습 데이터를 내보내도록 허용하려면 S3 버킷에 대한 PutObject 권한을 Personalize에 부여하는 서비스 연결 IAM 역할을 지정해야 합니다. 자세한 내용은 Personalize 개발자 안내서의 [데이터세트 내보내기](#) 단원을 참조하세요.

상태

데이터세트 내보내기 작업은 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED

내보내기 작업의 상태를 가져오려면 데이터세트 내보내기 작업의 Amazon 리소스 이름 (ARN) 을 [DescribeDatasetExportJob](#) 호출하고 지정합니다. 상태가 ACTIVE로 표시되면 데이터세트 가져오기가 완료된 것입니다, 상태가 CREATE FAILED로 표시되는 경우 응답에는 작업이 실패한 이유를 설명하는 failureReason 키가 포함됩니다.

구문 요청

```
{
  "datasetArn": "string",
  "ingestionMode": "string",
  "jobName": "string",
  "jobOutput": {
    "s3DataDestination": {
      "kmsKeyArn": "string",
      "path": "string"
    }
  },
  "roleArn": "string",
  "tags": [
    {
      "tagKey": "string",
      "tagValue": "string"
    }
  ]
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

datasetArn

가져온 데이터를 수신하는 데이터세트의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

필수 사항 여부: Yes

ingestionMode

데이터를 가져온 방식에 따라 내보낼 데이터 데이터세트 가져오기 작업을 사용하여 가져온 BULK 데이터만 내보내거나, 콘솔, PutUsers 및 PutItems 작업을 사용하여 점진적으로 가져온 PUT 데이터만 내보내거나 PutEventsALL, 두 유형 모두에 대해 내보내도록 선택할 수 있습니다. 기본 값은 PUT입니다.

타입: 문자열

유효 값: BULK | PUT | ALL

필수 여부: 아니요

jobName

데이터세트 내보내기 작업의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

필수 사항 여부: Yes

jobOutput

작업의 출력이 저장되는 S3 버킷의 경로.

유형: [DatasetExportJobOutput](#) 객체

필수 여부: 예

roleArn

출력 S3 버킷에 데이터를 추가할 권한이 있는 IAM 서비스 역할의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.\@-_/]+`

필수 사항 여부: Yes

tags

데이터세트 내보내기 작업에 적용할 [태그](#) 목록입니다.

유형: [Tag](#) 객체 어레이

어레이 멤버: 최소 항목 수 0개. 최대 항목 수 200개.

필수 여부: 아니요

응답 구문

```
{
  "datasetExportJobArn": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

datasetExportJobArn

데이터세트 내보내기 작업의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

LimitExceededException

초당 요청 수 한도를 초과했습니다.

HTTP 상태 코드: 400

ResourceAlreadyExistsException

지정된 리소스가 이미 있습니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

TooManyTagsException

이 리소스에 적용할 수 있는 최대 태그 수를 초과했습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하세요.

- [AWS Command Line Interface](#)

- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

CreateDatasetGroup

서비스: Amazon Personalize

빈 데이터세트 그룹을 생성합니다. 데이터세트 그룹은 Personalize 리소스를 위한 컨테이너입니다. 데이터세트 그룹은 데이터세트 유형별로 하나씩, 최대 3개의 데이터세트를 포함할 수 있습니다.

- 항목 상호 작용
- Items
- 사용자
- 작업
- 작업 상호 작용

데이터세트 그룹은 도메인을 지정하고 추천자와 같은 사전 구성된 리소스를 사용하는 도메인 데이터 세트 그룹이나, 캠페인과 함께 배포하는 솔루션 버전이 포함된 솔루션과 같은 사용자 지정 리소스를 사용하는 사용자 지정 데이터세트 그룹일 수 있습니다. 도메인 데이터세트 그룹으로 시작하는 경우에도 사용자 지정 사용 사례에 대한 레시피로 학습하고 캠페인과 함께 배포한 솔루션 및 솔루션 버전과 같은 사용자 지정 리소스를 추가할 수 있습니다.

데이터세트 그룹은 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- DELETE PENDING

데이터세트 그룹의 상태를 확인하려면 `DescribeDatasetGroup` 호출하세요. 상태가 CREATE FAILED로 표시되는 경우 응답에는 생성이 실패한 이유를 설명하는 `failureReason` 키가 포함됩니다.

Note

데이터세트를 그룹에 추가하기 전에 데이터세트 그룹의 `status`이 ACTIVE가 될 때까지 기다려야 합니다.

AWS Key Management Service (KMS) 키를 지정하여 그룹의 데이터세트를 암호화할 수 있습니다. KMS 키를 지정하는 경우 키에 액세스할 권한이 있는 AWS Identity and Access Management (IAM) 역할도 포함시켜야 합니다.

요청 시 데이터세트 그룹 ARN이 필요한 API

- [CreateDataset](#)
- [CreateEventTracker](#)
- [CreateSolution](#)

관련 API

- [ListDatasetGroups](#)
- [DescribeDatasetGroup](#)
- [DeleteDatasetGroup](#)

구문 요청

```
{
  "domain": "string",
  "kmsKeyArn": "string",
  "name": "string",
  "roleArn": "string",
  "tags": [
    {
      "tagKey": "string",
      "tagValue": "string"
    }
  ]
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[domain](#)

데이터세트 그룹의 도메인. 도메인 데이터세트 그룹을 생성하려면 도메인을 지정합니다. 지정하는 도메인에 따라 데이터세트의 기본 스키마와, 추천자가 사용할 수 있는 사용 사례가 결정됩니다. 도메인을 지정하지 않는 경우 캠페인과 함께 배포하는 솔루션 버전으로 사용자 지정 데이터세트 그룹을 생성합니다.

타입: 문자열

유효 값: ECOMMERCE | VIDEO_ON_DEMAND

필수 여부: 아니요

kmsKeyArn

데이터세트를 암호화하는 데 사용되는 (KMS) 키의 Amazon 리소스 이름 AWS Key Management Service (ARN).

타입: 문자열

길이 제약: 최대 길이 2048.

패턴: arn:aws.*:kms:.*:[0-9]{12}:key/.*

Required: No

name

새로운 데이터세트 그룹의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: ^[a-zA-Z0-9][a-zA-Z0-9\-_]*

필수 사항 여부: Yes

roleArn

(KMS) 키에 액세스할 수 있는 권한이 있는 AWS Identity and Access Management (IAM) 역할의 AWS Key Management Service ARN. IAM 역할을 제공하는 것은 KMS 키를 함께 지정하는 경우에 만 유효합니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):iam:.*:\d{12}:role/?[a-zA-Z_0-9+=,.\@\-_/\]+

Required: No

tags

데이터세트 그룹에 적용할 [태그](#) 목록입니다.

유형: [Tag](#)객체 배열

어레이 멤버: 최소 항목 수 0개. 최대 항목 수 200개.

필수 여부: 아니요

응답 구문

```
{
  "datasetGroupArn": "string",
  "domain": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[datasetGroupArn](#)

새 데이터세트 그룹의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

[domain](#)

새 도메인 데이터세트 그룹의 도메인.

타입: 문자열

유효 값: ECOMMERCE | VIDEO_ON_DEMAND

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

LimitExceededException

초당 요청 수 한도를 초과했습니다.

HTTP 상태 코드: 400

ResourceAlreadyExistsException

지정된 리소스가 이미 있습니다.

HTTP 상태 코드: 400

TooManyTagsException

이 리소스에 적용할 수 있는 최대 태그 수를 초과했습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

CreateDatasetImportJob

서비스: Amazon Personalize

데이터 소스(S3 버킷)에서 Personalize 데이터세트로 학습 데이터를 가져오는 작업을 생성합니다. Personalize에서 학습 데이터를 가져오도록 허용하려면 Personalize가 데이터 사본을 만들어 내부적으로 처리하므로 데이터 소스에서 읽을 수 있는 권한이 있는 IAM 서비스 역할을 지정해야 합니다. S3 버킷에 대한 액세스 권한의 부여에 대한 자세한 내용은 [S3 리소스에 대한 액세스 권한 Personalize에 부여](#) 단원을 참조하세요.

이미 추천자를 생성했거나 캠페인과 함께 사용자 지정 솔루션 버전을 배포한 경우 새 대량 레코드가 추천에 미치는 영향은 사용하는 도메인 사용 사례 또는 레시피에 따라 달라집니다. 자세한 내용은 [새 데이터가 실시간 추천에 미치는 영향](#) 단원을 참조하세요.

Important

기본 설정으로 데이터세트 가져오기 작업은 대량으로 가져온 데이터세트의 기존 데이터를 대체합니다. 기존 데이터를 바꾸지 않고 새 레코드를 추가하려면 CreateDatasetImportJob 작업의 가져오기 모드로 INCREMENTAL을 지정하십시오.

상태

데이터세트 가져오기 작업은 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED

가져오기 작업의 상태를 확인하려면 데이터세트 가져오기 작업의 Amazon 리소스 이름 (ARN) 을 제공하여 [DescribeDatasetImportJob](#)호출하십시오. 상태가 ACTIVE로 표시되면 데이터세트 가져오기가 완료된 것이고, 상태가 CREATE FAILED로 표시되는 경우 응답에는 작업이 실패한 이유를 설명하는 failureReason키가 포함됩니다.

Note

가져오기에는 시간이 필요합니다. 데이터세트를 사용하여 모델을 학습시키기 전에 상태가 ACTIVE로 나타날 때까지 기다려야 합니다.

관련 API

- [ListDatasetImportJobs](#)
- [DescribeDatasetImportJob](#)

구문 요청

```
{
  "datasetArn": "string",
  "dataSource": {
    "dataLocation": "string"
  },
  "importMode": "string",
  "jobName": "string",
  "publishAttributionMetricsToS3": boolean,
  "roleArn": "string",
  "tags": [
    {
      "tagKey": "string",
      "tagValue": "string"
    }
  ]
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[datasetArn](#)

가져온 데이터를 수신하는 데이터세트의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 사항 여부: Yes

[dataSource](#)

가져올 학습 데이터가 들어 있는 S3 버킷.

유형: [DataSource](#) 객체

필수 여부: 예

[importMode](#)

기존 데이터세트에 새 레코드를 추가하는 방법을 지정합니다. 기본 가져오기 모드는 FULL입니다. 이전에 데이터세트로 대량 레코드를 가져온 적이 없는 경우 FULL만 지정할 수 있습니다.

- 데이터세트의 기존 대량 데이터를 모두 덮어쓰려면 FULL을 지정합니다. 개별적으로 가져온 데이터는 대체되지 않습니다.
- 새 레코드를 데이터세트의 기존 데이터에 추가하려면 INCREMENTAL을 지정합니다. Personalize는 동일한 ID를 가진 모든 레코드를 새 레코드로 대체합니다.

타입: 문자열

유효 값: FULL | INCREMENTAL

필수 여부: 아니요

[jobName](#)

데이터세트 가져오기 작업의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

필수 사항 여부: Yes

[publishAttributionMetricsToS3](#)

지표 어트리뷰션을 생성한 경우 이 가져오기 작업에 대한 지표를 S3에 게시할지 여부를 지정합니다

타입: 부울

필수 항목 여부: 아니요

[roleArn](#)

S3 데이터 소스에서 읽을 권한이 있는 IAM 역할의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.\@\-_/\]+`

필수 사항 여부: Yes

[tags](#)

데이터세트 가져오기 작업에 적용할 [태그](#)의 목록입니다.

유형: [Tag](#) 객체 어레이

어레이 멤버: 최소 항목 수 0개. 최대 항목 수 200개.

필수 여부: 아니요

응답 구문

```
{
  "datasetImportJobArn": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[datasetImportJobArn](#)

데이터세트 가져오기 작업의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

LimitExceededException

초당 요청 수 한도를 초과했습니다.

HTTP 상태 코드: 400

ResourceAlreadyExistsException

지정된 리소스가 이미 있습니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

TooManyTagsException

이 리소스에 적용할 수 있는 최대 태그 수를 초과했습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)

- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

CreateEventTracker

서비스: Amazon Personalize

[PutEvents](#) API를 사용하여 지정된 데이터세트 그룹에 이벤트 데이터를 추가할 때 사용하는 이벤트 트래커를 만듭니다.

Note

이벤트 트래커 하나만 데이터세트 그룹과 연결할 수 있습니다. 기존의 이벤트 트래커와 동일한 데이터세트 그룹을 사용하여 `CreateEventTracker`를 호출하는 경우 오류가 발생합니다.

이벤트 추적기를 만들면 응답에 추적 ID가 포함되며, 추적 ID는 [PutEvents](#) 작업 사용 시 매개변수로 전달됩니다. 그러면 Amazon Personalize는 이벤트 트래커에서 지정한 데이터 세트 그룹의 항목 상호 작용 데이터 세트에 이벤트 데이터를 추가합니다.

이벤트 트래커는 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

이벤트 트래커의 상태를 확인하려면 `를 호출하십시오 DescribeEventTracker.`

Note

추적 ID를 사용하려면 이벤트 트래커가 ACTIVE 상태여야 합니다.

관련 API

- [ListEventTrackers](#)
- [DescribeEventTracker](#)
- [DeleteEventTracker](#)

구문 요청

```
{  
  "datasetGroupArn": "string",
```

```

    "name": "string",
    "tags": [
      {
        "tagKey": "string",
        "tagValue": "string"
      }
    ]
  }

```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

datasetGroupArn

이벤트 데이터를 수신하는 데이터세트 그룹의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*+

필수 사항 여부: Yes

name

이벤트 트래커의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: ^[a-zA-Z0-9][a-zA-Z0-9\-_]*

필수 사항 여부: Yes

tags

이벤트 트래커에 적용할 [태그](#)의 목록입니다.

유형: [Tag](#) 객체 어레이

어레이 멤버: 최소 항목 수 0개. 최대 항목 수 200개.

필수 여부: 아니요

응답 구문

```
{
  "eventTrackerArn": "string",
  "trackingId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

eventTrackerArn

이벤트 트래커의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*+

trackingId

이벤트 트래커의 ID. [PutEvents](#) API에 대한 요청에 이 ID를 포함시키십시오.

타입: 문자열

길이 제약: 최대 길이 256.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

LimitExceededException

초당 요청 수 한도를 초과했습니다.

HTTP 상태 코드: 400

ResourceAlreadyExistsException

지정된 리소스가 이미 있습니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

TooManyTagsException

이 리소스에 적용할 수 있는 최대 태그 수를 초과했습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

CreateFilter

서비스: Amazon Personalize

추천 필터를 생성합니다. 자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 단원을 참조하세요.

구문 요청

```
{
  "datasetGroupArn": "string",
  "filterExpression": "string",
  "name": "string",
  "tags": [
    {
      "tagKey": "string",
      "tagValue": "string"
    }
  ]
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[datasetGroupArn](#)

필터가 속할 데이터세트 그룹의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 사항 여부: Yes

[filterExpression](#)

필터 표현식은 추천에 포함되거나 추천에서 제외되는 항목을 정의합니다. 필터 표현식은 특정 형식 규칙을 따라야 합니다. 필터 표현식 구조 및 구문에 대한 자세한 정보는 [필터 표현식](#) 단원을 참조하세요.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 2500.

필수 여부: 예

name

생성할 필터의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

필수 사항 여부: Yes

tags

함수에 적용할 [태그](#)의 목록입니다.

유형: [Tag](#) 객체 어레이

어레이 멤버: 최소 항목 수 0개. 최대 항목 수 200개.

필수 여부: 아니요

응답 구문

```
{
  "filterArn": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

filterArn

새 필터의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

LimitExceededException

초당 요청 수 한도를 초과했습니다.

HTTP 상태 코드: 400

ResourceAlreadyExistsException

지정된 리소스가 이미 있습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

TooManyTagsException

이 리소스에 적용할 수 있는 최대 태그 수를 초과했습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)

- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

CreateMetricAttribution

서비스: Amazon Personalize

지표 어트리뷰션을 생성합니다. 지표 어트리뷰션은 Personalize로 가져오는 데이터에 대한 보고서를 생성합니다. 데이터를 가져온 방법에 따라 Amazon CloudWatch 또는 Amazon S3에서 보고서를 볼 수 있습니다. 자세한 내용은 [추천의 영향 측정](#) 단원을 참조하세요.

구문 요청

```
{
  "datasetGroupArn": "string",
  "metrics": [
    {
      "eventType": "string",
      "expression": "string",
      "metricName": "string"
    }
  ],
  "metricsOutputConfig": {
    "roleArn": "string",
    "s3DataDestination": {
      "kmsKeyArn": "string",
      "path": "string"
    }
  },
  "name": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

datasetGroupArn

지표 어트리뷰션에 대한 대상 데이터세트 그룹의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]):personalize:.*:.*:.*

필수 사항 여부: Yes

metrics

지표 어트리뷰션에 대한 지표 속성의 목록입니다. 각 지표 속성은 추적할 이벤트 유형과 함수를 지정합니다. 사용할 수 있는 함수는 SUM() 또는 SAMPLECOUNT()입니다. SUM() 함수의 경우 데이터 세트 유형(상호작용 또는 항목)과 집계할 열을 파라미터로 제공합니다. 예: SUM(Items.PRICE).

유형: [MetricAttribute](#) 객체 어레이

어레이 멤버: 최대 항목 수 10개.

필수 여부: 예

metricsOutputConfig

지표 어트리뷰션에 대한 출력 구성 세부 정보입니다.

유형: [MetricAttributionOutput](#) 객체

필수 여부: 예

name

지표 어트리뷰션의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

필수 항목 여부: 예

응답 구문

```
{
  "metricAttributionArn": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

metricAttributionArn

새 지표 어트리뷰션의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

LimitExceededException

초당 요청 수 한도를 초과했습니다.

HTTP 상태 코드: 400

ResourceAlreadyExistsException

지정된 리소스가 이미 있습니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

CreateRecommender

서비스: Amazon Personalize

지정한 레시피(도메인 데이터세트 그룹 사용 사례)로 추천자를 생성합니다. 도메인 데이터세트 그룹에 대한 추천자를 생성하고 요청 시 추천자의 Amazon 리소스 이름 (ARN) 을 지정합니다.

[GetRecommendations](#)

초당 최소 추천 요청 수

Important

`minRecommendationRequestsPerSecond`가 높으면 요금이 인상됩니다. `minRecommendationRequestsPerSecond`(기본값)의 경우 1부터 시작하는 것이 좋습니다. Amazon CloudWatch 지표를 사용하여 사용량을 추적하고 `minRecommendationRequestsPerSecond` 필요에 따라 사용량을 늘리십시오.

추천을 생성할 때 추천의 초당 최소 추천 요청을 구성할 수 있습니다. 초당 최소 추천 요청 (`minRecommendationRequestsPerSecond`)은 Personalize에서 프로비저닝하는 기본 추천 요청 처리량을 지정합니다. 기본값은 `minRecommendationRequestsPerSecond` 입니다¹. 추천 요청은 단일한 `GetRecommendations` 작업입니다. 요청 처리량은 초당 요청 수로 측정되며 Personalize는 초당 요청을 사용하여 시간당 요청 및 추천 사용 요금을 도출합니다.

초당 요청의 수가 `minRecommendationRequestsPerSecond` 이상으로 증가할 경우, Personalize는 프로비저닝된 용량을 자동으로 확장 및 축소하지만 `minRecommendationRequestsPerSecond` 미만으로는 절대 그렇게 하지 않습니다. 용량이 증가 중에 잠시 동안의 지연으로 요청의 손실이 발생할 수 있습니다.

요금은 시간당 최소 요청 수 (기준 `minRecommendationRequestsPerSecond`) 와 실제 요청 수 중 더 큰 금액입니다. 실제 요청 처리량은 1시간 이내의 초당 평균 요청 수로 계산됩니다. `minRecommendationRequestsPerSecond` 기본값부터 시작하여 Amazon CloudWatch 메트릭을 사용하여 사용량을 추적한 다음 `minRecommendationRequestsPerSecond` 필요에 따라 늘리는 것이 좋습니다.

상태

추천자는 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED

- STOP PENDING > STOP IN_PROGRESS > INACTIVE > START PENDING > START IN_PROGRESS > ACTIVE
- DELETE PENDING > DELETE IN_PROGRESS

추천자 상태를 확인하려면 전화하십시오 [DescribeRecommender](#).

Note

추천자에게 추천을 요청하기 전에 추천자의 status가 ACTIVE가 될 때까지 기다립니다.

관련 API

- [ListRecommenders](#)
- [DescribeRecommender](#)
- [UpdateRecommender](#)
- [DeleteRecommender](#)

구문 요청

```
{
  "datasetGroupArn": "string",
  "name": "string",
  "recipeArn": "string",
  "recommenderConfig": {
    "enableMetadataWithRecommendations": boolean,
    "itemExplorationConfig": {
      "string" : "string"
    },
    "minRecommendationRequestsPerSecond": number,
    "trainingDataConfig": {
      "excludedDatasetColumns": {
        "string" : [ "string" ]
      }
    }
  },
  "tags": [
    {
      "tagKey": "string",
```

```

    "tagValue": "string"
  }
]
}

```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

datasetGroupArn

추천자에 대한 대상 도메인 데이터세트 그룹의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

필수 사항 여부: Yes

name

추천자의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

필수 사항 여부: Yes

recipeArn

추천자가 사용할 레시피의 리소스 이름(ARN)입니다. 추천자의 경우 레시피는 도메인 데이터세트 그룹 사용 사례입니다. 도메인 데이터세트 그룹 사용 사례만 추천자를 만드는 데 사용할 수 있습니다. 사용 사례에 대한 자세한 내용은 [추천자 사용 사례 선택](#) 단원을 참조하세요.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

필수 사항 여부: Yes

[recommenderConfig](#)

추천자의 구성 세부 정보입니다.

유형: [RecommenderConfig](#) 객체

필수 여부: 아니요

[tags](#)

추천자에 적용할 [태그](#)의 목록입니다.

유형: [Tag](#) 객체 어레이

어레이 멤버: 최소 항목 수 0개. 최대 항목 수 200개.

필수 여부: 아니요

응답 구문

```
{
  "recommenderArn": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[recommenderArn](#)

추천자의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

LimitExceededException

초당 요청 수 한도를 초과했습니다.

HTTP 상태 코드: 400

ResourceAlreadyExistsException

지정된 리소스가 이미 있습니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

TooManyTagsException

이 리소스에 적용할 수 있는 최대 태그 수를 초과했습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)

- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

CreateSchema

서비스: Amazon Personalize

지정된 스키마 문자열에서 Personalize 스키마를 생성합니다. 생성하는 스키마는 Avro JSON 형식이어야 합니다.

Personalize는 3가지 스키마 변형을 인식합니다. 각 스키마는 데이터세트 유형과 연결되며 필수 필드 및 키워드 세트가 있습니다. 도메인 데이터세트 그룹의 데이터세트에 대한 스키마를 생성하는 경우 도메인 데이터세트 그룹의 도메인을 제공합니다. 호출할 때 스키마를 지정합니다 [CreateDataset](#).

스키마에 대한 자세한 내용을 알아보려면 [데이터세트 및 스키마](#)를 참조하세요.

관련 API

- [ListSchemas](#)
- [DescribeSchema](#)
- [DeleteSchema](#)

구문 요청

```
{
  "domain": "string",
  "name": "string",
  "schema": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

domain

스키마의 도메인. 도메인 데이터세트 그룹의 데이터세트에 대한 스키마를 생성하는 경우 도메인 데이터세트 그룹에서 생성할 때 선택한 도메인을 지정합니다.

타입: 문자열

유효 값: ECOMMERCE | VIDEO_ON_DEMAND

필수 여부: 아니요

name

스키마의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

필수 사항 여부: Yes

schema

Avro JSON 형식의 스키마.

타입: 문자열

길이 제약: 최대 길이는 20,000입니다.

필수 여부: 예

응답 구문

```
{
  "schemaArn": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

schemaArn

스키마의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

LimitExceededException

초당 요청 수 한도를 초과했습니다.

HTTP 상태 코드: 400

ResourceAlreadyExistsException

지정된 리소스가 이미 있습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

CreateSolution

서비스: Amazon Personalize

⚠ Important

솔루션을 만든 후에는 구성을 변경할 수 없습니다. 기본적으로 모든 새 솔루션은 자동 교육을 사용합니다. 자동 교육을 사용하면 솔루션이 활성화되는 동안 교육 비용이 발생합니다. 솔루션에 대한 자동 교육을 중단할 수는 없습니다. 불필요한 비용이 발생하지 않도록 하려면 완료 후 솔루션을 삭제해야 합니다. 교육 비용에 대한 자세한 내용은 [Amazon Personalize](#) 요금을 참조하십시오.

모델 교육을 위한 구성을 생성합니다 (솔루션 버전 생성). 이 구성에는 모델 학습에 사용할 레시피와 선택적 학습 구성 (예: 학습 및 기능 변환 매개변수에 사용할 열) 이 포함됩니다. 솔루션 구성에 대한 자세한 내용은 솔루션 [만들기 및 구성](#)을 참조하십시오.

기본적으로 새 솔루션은 자동 교육을 사용하여 7일마다 솔루션 버전을 생성합니다. 훈련 빈도를 변경할 수 있습니다. 솔루션이 활성 상태인 지 한 시간 후에 자동 솔루션 버전 생성이 시작됩니다. 한 시간 내에 솔루션 버전을 수동으로 만들면 솔루션은 첫 번째 자동 교육을 건너뛰게 됩니다. 자세한 내용은 [자동 교육 구성](#)을 참조하십시오.

자동 학습을 `performAutoTraining` 끄려면 `false`로 설정하십시오. 자동 학습을 끄는 경우 [CreateSolutionVersion](#) 작업을 호출하여 솔루션 버전을 수동으로 생성해야 합니다.

교육이 시작되면 [ListSolutionVersions](#) API 작업을 통해 솔루션 버전의 Amazon 리소스 이름 (ARN) 을 가져올 수 있습니다. 상태를 확인하려면 `l` 을 사용하십시오. [DescribeSolutionVersion](#)

훈련이 완료되면 `l` 을 [GetSolutionMetrics](#) 호출하여 모델 정확도를 평가할 수 있습니다. 솔루션 버전에 만족하면 `l` 을 사용하여 [CreateCampaign](#) 배포합니다. 캠페인은 [GetRecommendations](#) API를 통해 클라이언트에게 권장 사항을 제공합니다.

ℹ Note

Personalize는 현재 솔루션 하이퍼파라미터에 대해 `hpoObjective` 구성을 지원하지 않습니다.

상태

솔루션은 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

솔루션 상태를 확인하려면 전화하십시오 [DescribeSolution](#). 수동 교육을 사용하는 경우 전화를 걸기 전에 상태가 ACTIVE여야 CreateSolutionVersion 합니다.

관련 API

- [ListSolutions](#)
- [CreateSolutionVersion](#)
- [DescribeSolution](#)
- [DeleteSolution](#)

- [ListSolutionVersions](#)
- [DescribeSolutionVersion](#)

구문 요청

```
{
  "datasetGroupArn": "string",
  "eventType": "string",
  "name": "string",
  "performAutoML": boolean,
  "performAutoTraining": boolean,
  "performHPO": boolean,
  "recipeArn": "string",
  "solutionConfig": {
    "algorithmHyperParameters": {
      "string": "string"
    },
    "autoMLConfig": {
      "metricName": "string",
      "recipeList": [ "string" ]
    },
    "autoTrainingConfig": {
      "schedulingExpression": "string"
    },
    "eventValueThreshold": "string",
    "featureTransformationParameters": {
```

```
    "string" : "string"
  },
  "hpoConfig": {
    "algorithmHyperParameterRanges": {
      "categoricalHyperParameterRanges": [
        {
          "name": "string",
          "values": [ "string" ]
        }
      ],
      "continuousHyperParameterRanges": [
        {
          "maxValue": number,
          "minValue": number,
          "name": "string"
        }
      ],
      "integerHyperParameterRanges": [
        {
          "maxValue": number,
          "minValue": number,
          "name": "string"
        }
      ]
    },
    "hpoObjective": {
      "metricName": "string",
      "metricRegex": "string",
      "type": "string"
    },
    "hpoResourceConfig": {
      "maxNumberOfTrainingJobs": "string",
      "maxParallelTrainingJobs": "string"
    }
  },
  "optimizationObjective": {
    "itemAttribute": "string",
    "objectiveSensitivity": "string"
  },
  "trainingDataConfig": {
    "excludedDatasetColumns": {
      "string" : [ "string" ]
    }
  }
}
```

```
  },  
  "tags": [  
    {  
      "tagKey": "string",  
      "tagValue": "string"  
    }  
  ]  
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[datasetGroupArn](#)

학습 데이터를 제공하는 데이터세트 그룹의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

필수 사항 여부: Yes

[eventType](#)

여러 이벤트 유형(EVENT_TYPE 스키마 필드 사용)이 있는 경우 이 파라미터는 모델 학습에 사용되는 이벤트 유형(예: 'click' 또는 'like')입니다.

eventType이 제공되지 않으면 Personalize는 유형에 관계없이 동일한 가중치로 모든 상호작용을 학습에 사용합니다.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

[name](#)

솔루션의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

필수 사항 여부: Yes

[performAutoML](#)

Important

자동화된 기계 학습은 활성화하지 않는 것이 좋습니다. 대신 사용 사례를 사용 가능한 Personalize 레시피에 맞춥니다. 자세한 내용은 [레시피 선택](#)을 참조하세요.

자동 기계 학습(AutoML) 수행 여부. 기본값은 false입니다. 이 경우에는 recipeArn를 지정해야 합니다.

true로 설정하면 Personalize는 학습 데이터를 분석하고 최적의 USER_PERSONALIZATION 레시피와 하이퍼파라미터를 선택합니다. 이 경우에는 recipeArn을 생략해야 합니다. Personalize는 하이퍼파라미터에 대해 서로 다른 값으로 테스트를 실행하여 최적의 레시피를 결정합니다. AutoML을 사용할 경우 특정 레시피를 선택하는 것에 비해 학습 과정이 더 길어집니다.

타입: 부울

필수 항목 여부: 아니요

[performAutoTraining](#)

솔루션이 자동 학습을 사용하여 새 솔루션 버전 (학습된 모델) 을 생성하는지 여부 기본값은 True 이며 솔루션은 7일마다 새 솔루션 버전을 자동으로 생성합니다. 솔루션 구성의 AutoTrainingConfig 일부로 a를 schedulingExpression 지정하여 교육 빈도를 변경할 수 있습니다. 자동 훈련에 대한 자세한 내용은 [자동 훈련 구성](#)을 참조하십시오.

솔루션이 활성 상태인 지 한 시간 후에 자동 솔루션 버전 생성이 시작됩니다. 한 시간 내에 솔루션 버전을 수동으로 만들면 솔루션은 첫 번째 자동 교육을 건너뛰게 됩니다.

교육이 시작되면 [ListSolutionVersions](#) API 작업을 통해 솔루션 버전의 Amazon 리소스 이름 (ARN) 을 가져올 수 있습니다. 상태를 확인하려면 를 사용하십시오. [DescribeSolutionVersion](#)

타입: 부울

필수 항목 여부: 아니요

[performHPO](#)

지정 또는 선택한 레시피에서 하이퍼파라미터 최적화(HPO)를 수행할지 여부. 기본값은 false입니다.

AutoML을 수행할 때 이 파라미터는 항상 true이므로 false로 설정하지 않아야 합니다.

타입: 부울

필수 항목 여부: 아니요

[recipeArn](#)

모델 교육에 사용할 레시피의 Amazon 리소스 이름(ARN). performAutoML이 false인 경우 필수입니다. 다양한 Amazon Personalize 레시피 및 해당 ARN에 대한 자세한 내용은 [레시피 선택](#)을 참조하세요.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*+

Required: No

[solutionConfig](#)

솔루션과 함께 사용할 구성. performAutoML가 true로 설정된 경우 Personalize는 솔루션 구성의 autoMLConfig섹션만 평가합니다.

Note

Personalize는 현재 hpoObjective구성을 지원하지 않습니다.

유형: [SolutionConfig](#)객체

필수 여부: 아니요

[tags](#)

솔루션에 적용할 [태그](#)의 목록입니다.

유형: [Tag](#)객체 어레이

어레이 멤버: 최소 항목 수 0개. 최대 항목 수 200개.

필수 여부: 아니요

응답 구문

```
{  
  "solutionArn": "string"  
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

solutionArn

솔루션의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

LimitExceededException

초당 요청 수 한도를 초과했습니다.

HTTP 상태 코드: 400

ResourceAlreadyExistsException

지정된 리소스가 이미 있습니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

TooManyTagsException

이 리소스에 적용할 수 있는 최대 태그 수를 초과했습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

CreateSolutionVersion

서비스: Amazon Personalize

사용자 지정 데이터세트 그룹에서 활성 솔루션을 학습시키거나 재학습시킵니다. 솔루션은 [CreateSolution](#) 작업을 사용하여 생성되며 호출하기 전에 ACTIVE 상태여야 CreateSolutionVersion 합니다. 이 작업을 호출할 때마다 새 솔루션 버전이 생성됩니다.

상태

솔루션 버전은 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED
- CREATE STOPPING
- CREATE STOPPED

버전 상태를 확인하려면 를 호출하십시오 [DescribeSolutionVersion](#). 상태가 ACTIVE로 표시될 때까지 기다렸다가 CreateCampaign을 호출합니다.

상태가 CREATE FAILED로 표시되는 경우 응답에는 작업이 실패한 이유를 설명하는 failureReason키가 포함됩니다.

관련 API

- [ListSolutionVersions](#)
- [DescribeSolutionVersion](#)
- [ListSolutions](#)
- [CreateSolution](#)
- [DescribeSolution](#)
- [DeleteSolution](#)

구문 요청

```
{  
  "name": "string",
```

```

"solutionArn": "string",
"tags": [
  {
    "tagKey": "string",
    "tagValue": "string"
  }
],
"trainingMode": "string"
}

```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

name

솔루션 버전의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

solutionArn

학습 구성 정보가 포함된 솔루션의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

필수 사항 여부: Yes

tags

솔루션 버전에 적용할 [태그](#)의 목록입니다.

유형: [Tag](#) 객체 어레이

어레이 멤버: 최소 항목 수 0개. 최대 항목 수 200개.

필수 여부: 아니요

[trainingMode](#)

솔루션 버전을 만들 때 수행할 학습의 범위. 기본값은 FULL입니다. 이렇게 하면 데이터세트 그룹 내 데이터세트의 전체 학습 데이터를 기반으로 완전히 새로운 모델이 생성됩니다.

[사용자-개인 맞춤](#)을 사용하는 경우 학습 모드를 UPDATE로 지정할 수 있습니다. 이렇게 하면 새 항목을 추천에 고려하도록 모델이 업데이트됩니다. 이는 전체 재학습에 해당되지 않습니다. 매주 전체 재학습을 이수해야 합니다. UPDATE을 지정하는 경우 Personalize는 솔루션 버전에 대한 자동 업데이트를 중지합니다. 업데이트를 재개하려면 학습 모드를 FULL로 설정한 상태에서 새 솔루션을 생성하여 캠페인에서 배포합니다. 자동 업데이트에 대한 자세한 내용은 [자동 업데이트](#) 단원을 참조하세요.

이 UPDATE 옵션은 FULL 옵션을 사용하여 입력 솔루션에서 만든 활성 솔루션 버전이 이미 있고 입력 솔루션이 [사용자-개인 맞춤](#) 레시피 또는 레거시 [HRNN-ColdStart](#) 레시피로 학습된 경우에만 사용할 수 있습니다.

타입: 문자열

유효 값: FULL | UPDATE | AUTOTRAIN

필수 항목 여부: 아니요

응답 구문

```
{
  "solutionVersionArn": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[solutionVersionArn](#)

새 솔루션 버전의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

LimitExceededException

초당 요청 수 한도를 초과했습니다.

HTTP 상태 코드: 400

ResourceAlreadyExistsException

지정된 리소스가 이미 있습니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

TooManyTagsException

이 리소스에 적용할 수 있는 최대 태그 수를 초과했습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DeleteCampaign

서비스: Amazon Personalize

솔루션 배포를 삭제하여 캠페인을 제거합니다. 캠페인의 기반이 되는 솔루션은 삭제되지 않으며 필요할 때 재배포할 수 있습니다. 삭제된 캠페인은 더 이상 [GetRecommendations](#) 요청에 지정할 수 없습니다. 캠페인 생성에 대한 자세한 내용은 을 참조하십시오 [CreateCampaign](#).

구문 요청

```
{
  "campaignArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[campaignArn](#)

삭제할 캠페인의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DeleteDataset

서비스: Amazon Personalize

데이터세트를 삭제합니다. 관련 DatasetImportJob 또는 SolutionVersion가 CREATE PENDING 또는 IN PROGRESS 상태인 경우 데이터세트를 삭제할 수 없습니다. 데이터세트에 대한 자세한 내용은 을 참조하십시오 [CreateDataset](#).

구문 요청

```
{
  "datasetArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[datasetArn](#)

삭제할 데이터세트의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하세요.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS 파이썬용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DeleteDatasetGroup

서비스: Amazon Personalize

데이터세트 그룹을 삭제합니다. 데이터세트 그룹을 삭제하려면 다음을 삭제해야 합니다.

- 모든 연결된 이벤트 트래커.
- 모든 연결된 솔루션.
- 데이터세트 그룹 내 모든 데이터세트.

구문 요청

```
{  
  "datasetGroupArn": "string"  
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

datasetGroupArn

삭제할 데이터세트 그룹의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DeleteEventTracker

서비스: Amazon Personalize

이벤트 트래커를 삭제합니다. 데이터 세트 그룹에서 데이터 세트를 삭제하지 않습니다. 이벤트 트래커에 대한 자세한 내용은 [이 참조하십시오 `CreateEventTracker`](#).

구문 요청

```
{  
  "eventTrackerArn": "string"  
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[eventTrackerArn](#)

삭제할 이벤트 트래커의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DeleteFilter

서비스: Amazon Personalize

필터를 삭제합니다.

구문 요청

```
{  
  "filterArn": "string"  
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

filterArn

삭제할 필터의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DeleteMetricAttribution

서비스: Amazon Personalize

지표 어트리뷰션을 삭제합니다.

구문 요청

```
{  
  "metricAttributionArn": "string"  
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

metricAttributionArn

지표 어트리뷰션의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DeleteRecommender

서비스: Amazon Personalize

추천자를 비활성화하고 제거합니다. 삭제된 추천자는 더 이상 [GetRecommendations](#)요청에 지정할 수 없습니다.

구문 요청

```
{  
  "recommenderArn": "string"  
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[recommenderArn](#)

삭제할 추천자의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

`InvalidInputException`

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

`ResourceInUseException`

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DeleteSchema

서비스: Amazon Personalize

스키마를 삭제합니다. 스키마를 삭제하기 전에 해당 스키마를 참조하는 모든 데이터세트를 삭제해야 합니다. 스키마에 대한 자세한 내용은 [이 참조하십시오](#) [CreateSchema](#).

구문 요청

```
{  
  "schemaArn": "string"  
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[schemaArn](#)

삭제할 스키마의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DeleteSolution

서비스: Amazon Personalize

솔루션의 모든 버전과 Solution객체 자체를 삭제합니다. 솔루션을 삭제하기 전에 해당 솔루션에 기반한 모든 캠페인을 삭제해야 합니다. 솔루션을 사용하는 캠페인을 확인하려면 솔루션의 Amazon 리소스 이름 (ARN) 을 [ListCampaigns](#)호출하여 제공하십시오. 관련 SolutionVersion이 CREATE PENDING 또는 IN PROGRESS 상태인 경우 솔루션을 삭제할 수 없습니다. 솔루션에 대한 자세한 내용은 [참조하십시오 CreateSolution](#).

구문 요청

```
{
  "solutionArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[solutionArn](#)

삭제할 솔루션의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeAlgorithm

서비스: Amazon Personalize

주어진 알고리즘을 설명합니다.

구문 요청

```
{
  "algorithmArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

algorithmArn

설명할 알고리즘의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 항목 여부: 예

응답 구문

```
{
  "algorithm": {
    "algorithmArn": "string",
    "algorithmImage": {
      "dockerURI": "string",
      "name": "string"
    },
    "creationDateTime": number,
    "defaultHyperParameterRanges": {
      "categoricalHyperParameterRanges": [
        {
          "isTunable": boolean,
          "name": "string",

```



```

        "values": [ "string" ]
    }
],
"continuousHyperParameterRanges": [
    {
        "isTunable": boolean,
        "maxValue": number,
        "minValue": number,
        "name": "string"
    }
],
"integerHyperParameterRanges": [
    {
        "isTunable": boolean,
        "maxValue": number,
        "minValue": number,
        "name": "string"
    }
]
},
"defaultHyperParameters": {
    "string" : "string"
},
"defaultResourceConfig": {
    "string" : "string"
},
"lastUpdatedDateTime": number,
"name": "string",
"roleArn": "string",
"trainingInputMode": "string"
}
}

```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[algorithm](#)

알고리즘의 속성 목록입니다.

유형: [Algorithm](#) 객체

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeBatchInferenceJob

서비스: Amazon Personalize

이름, 리소스 이름(ARN), 상태, 입력 및 출력 구성, 추천 생성에 사용된 솔루션 버전의 ARN 등 배치 추론 작업의 속성을 가져옵니다.

구문 요청

```
{
  "batchInferenceJobArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[batchInferenceJobArn](#)

설명할 배치 추론 작업의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 항목 여부: 예

응답 구문

```
{
  "batchInferenceJob": {
    "batchInferenceJobArn": "string",
    "batchInferenceJobConfig": {
      "itemExplorationConfig": {
        "string" : "string"
      }
    },
    "batchInferenceJobMode": "string",
    "creationDateTime": number,
    "failureReason": "string",
    "filterArn": "string",
  }
}
```

```

    "jobInput": {
      "s3DataSource": {
        "kmsKeyArn": "string",
        "path": "string"
      }
    },
    "jobName": "string",
    "jobOutput": {
      "s3DataDestination": {
        "kmsKeyArn": "string",
        "path": "string"
      }
    },
    "lastUpdatedDateTime": number,
    "numResults": number,
    "roleArn": "string",
    "solutionVersionArn": "string",
    "status": "string",
    "themeGenerationConfig": {
      "fieldsForThemeGeneration": {
        "itemName": "string"
      }
    }
  }
}

```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[batchInferenceJob](#)

지정된 배치 추론 작업에 대한 정보입니다.

유형: [BatchInferenceJob](#) 객체

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeBatchSegmentJob

서비스: Amazon Personalize

이름, 리소스 이름(ARN), 상태, 입력 및 출력 구성, 세그먼트를 생성하는 데 사용된 솔루션 버전의 ARN 등 배치 세그먼트 작업의 속성을 가져옵니다.

구문 요청

```
{
  "batchSegmentJobArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

batchSegmentJobArn

설명할 배치 세그먼트 작업의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 항목 여부: 예

응답 구문

```
{
  "batchSegmentJob": {
    "batchSegmentJobArn": "string",
    "creationDateTime": number,
    "failureReason": "string",
    "filterArn": "string",
    "jobInput": {
      "s3DataSource": {
        "kmsKeyArn": "string",
        "path": "string"
      }
    },
    "jobName": "string",
```

```
"jobOutput": {
  "s3DataDestination": {
    "kmsKeyArn": "string",
    "path": "string"
  }
},
"lastUpdatedDateTime": number,
"numResults": number,
"roleArn": "string",
"solutionVersionArn": "string",
"status": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[batchSegmentJob](#)

지정된 배치 세그먼트 작업에 대한 정보입니다.

유형: [BatchSegmentJob](#) 객체

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeCampaign

서비스: Amazon Personalize

상태를 포함하여 해당 캠페인을 설명합니다.

캠페인은 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

status가 CREATE FAILED인 경우 응답에는 이유를 설명하는 failureReason키가 포함됩니다.

캠페인에 대한 자세한 내용은 [이 링크](#)를 참조하십시오 [CreateCampaign](#).

구문 요청

```
{
  "campaignArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[campaignArn](#)

캠페인의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 항목 여부: 예

응답 구문

```
{
  "campaign": {
```

```

    "campaignArn": "string",
    "campaignConfig": {
      "enableMetadataWithRecommendations": boolean,
      "itemExplorationConfig": {
        "string": "string"
      },
      "syncWithLatestSolutionVersion": boolean
    },
    "creationDateTime": number,
    "failureReason": "string",
    "lastUpdatedDateTime": number,
    "latestCampaignUpdate": {
      "campaignConfig": {
        "enableMetadataWithRecommendations": boolean,
        "itemExplorationConfig": {
          "string": "string"
        },
        "syncWithLatestSolutionVersion": boolean
      },
      "creationDateTime": number,
      "failureReason": "string",
      "lastUpdatedDateTime": number,
      "minProvisionedTPS": number,
      "solutionVersionArn": "string",
      "status": "string"
    },
    "minProvisionedTPS": number,
    "name": "string",
    "solutionVersionArn": "string",
    "status": "string"
  }
}

```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

campaign

캠페인의 속성.

유형: Campaign 객체

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeDataDeletionJob

서비스: Amazon Personalize

작업 상태를 [CreateDataDeletionJob](#) 포함하여 만든 데이터 삭제 작업에 대해 설명합니다.

구문 요청

```
{
  "dataDeletionJobArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[dataDeletionJobArn](#)

데이터 삭제 작업의 Amazon 리소스 이름 (ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

필수 항목 여부: 예

응답 구문

```
{
  "dataDeletionJob": {
    "creationDateTime": number,
    "dataDeletionJobArn": "string",
    "datasetGroupArn": "string",
    "dataSource": {
      "dataLocation": "string"
    },
    "failureReason": "string",
    "jobName": "string",
    "lastUpdatedDateTime": number,
    "numDeleted": number,
    "roleArn": "string",
  }
}
```

```
    "status": "string"  
  }  
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[dataDeletionJob](#)

상태를 포함한 데이터 삭제 작업에 대한 정보.

상태는 다음 값 중 하나일 수 있습니다.

- PENDING
- IN_PROGRESS
- 완료됨
- FAILED

유형: [DataDeletionJob](#) 객체

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)

- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeDataset

서비스: Amazon Personalize

지정된 데이터세트를 설명합니다. 데이터세트에 대한 자세한 내용은 [이 링크](#)를 참조하십시오 [CreateDataset](#).

구문 요청

```
{
  "datasetArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[datasetArn](#)

설명할 데이터세트의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

필수 항목 여부: 예

응답 구문

```
{
  "dataset": {
    "creationDateTime": number,
    "datasetArn": "string",
    "datasetGroupArn": "string",
    "datasetType": "string",
    "lastUpdatedDateTime": number,
    "latestDatasetUpdate": {
      "creationDateTime": number,
      "failureReason": "string",
      "lastUpdatedDateTime": number,
      "schemaArn": "string",
      "status": "string"
    }
  }
}
```

```
    },  
    "name": "string",  
    "schemaArn": "string",  
    "status": "string",  
    "trackingId": "string"  
  }  
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[dataset](#)

데이터세트 속성 목록입니다.

유형: [Dataset](#) 객체

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하세요.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)

- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS 파이썬용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeDatasetExportJob

서비스: Amazon Personalize

내보내기 작업 상태를 [CreateDatasetExportJob](#) 포함하여 만든 데이터세트 내보내기 작업에 대해 설명합니다.

구문 요청

```
{
  "datasetExportJobArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[datasetExportJobArn](#)

설명할 데이터세트 내보내기 작업의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*+

필수 항목 여부: 예

응답 구문

```
{
  "datasetExportJob": {
    "creationDateTime": number,
    "datasetArn": "string",
    "datasetExportJobArn": "string",
    "failureReason": "string",
    "ingestionMode": "string",
    "jobName": "string",
    "jobOutput": {
      "s3DataDestination": {
        "kmsKeyArn": "string",
        "path": "string"
      }
    }
  }
}
```

```
    },  
    "lastUpdatedDateTime": number,  
    "roleArn": "string",  
    "status": "string"  
  }  
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[datasetExportJob](#)

상태를 포함한 데이터세트 내보내기 작업에 대한 정보입니다.

상태는 다음 값 중 하나일 수 있습니다.

- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED

유형: [DatasetExportJob](#) 객체

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeDatasetGroup

서비스: Amazon Personalize

지정된 데이터세트 그룹을 설명합니다. 데이터세트 그룹에 대한 자세한 내용은 [을 참조하십시오](#)
[CreateDatasetGroup](#).

구문 요청

```
{  
  "datasetGroupArn": "string"  
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[datasetGroupArn](#)

설명할 데이터세트 그룹의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 항목 여부: 예

응답 구문

```
{  
  "datasetGroup": {  
    "creationDateTime": number,  
    "datasetGroupArn": "string",  
    "domain": "string",  
    "failureReason": "string",  
    "kmsKeyArn": "string",  
    "lastUpdatedDateTime": number,  
    "name": "string",  
    "roleArn": "string",  
    "status": "string"  
  }  
}
```

```
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[datasetGroup](#)

데이터세트 그룹 속성의 목록입니다.

유형: [DatasetGroup](#) 객체

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하세요.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)

- [AWS 루비 V3용 SDK](#)

DescribeDatasetImportJob

서비스: Amazon Personalize

에서 만든 [CreateDatasetImportJob](#) 데이터세트 가져오기 작업에 대해 가져오기 작업 상태를 설명합니다.

구문 요청

```
{
  "datasetImportJobArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[datasetImportJobArn](#)

설명할 데이터세트 가져오기 작업의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 항목 여부: 예

응답 구문

```
{
  "datasetImportJob": {
    "creationDateTime": number,
    "datasetArn": "string",
    "datasetImportJobArn": "string",
    "dataSource": {
      "dataLocation": "string"
    },
    "failureReason": "string",
    "importMode": "string",
    "jobName": "string",
    "lastUpdatedDateTime": number,
  }
}
```



```
"publishAttributionMetricsToS3": boolean,  
"roleArn": "string",  
"status": "string"  
}  
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[datasetImportJob](#)

상태를 포함한 데이터세트 가져오기 작업에 대한 정보입니다.

상태는 다음 값 중 하나일 수 있습니다.

- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED

유형: [DatasetImportJob](#) 객체

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeEventTracker

서비스: Amazon Personalize

이벤트 트래커를 설명합니다. 응답에는 이벤트 트래커의 `trackingId` 및 `status`가 포함됩니다. 이벤트 트래커에 대한 자세한 내용은 [이 링크](#)를 참조하십시오 [CreateEventTracker](#).

구문 요청

```
{
  "eventTrackerArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[eventTrackerArn](#)

설명할 이벤트 트래커의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

필수 항목 여부: 예

응답 구문

```
{
  "eventTracker": {
    "accountId": "string",
    "creationDateTime": number,
    "datasetGroupArn": "string",
    "eventTrackerArn": "string",
    "lastUpdatedDateTime": number,
    "name": "string",
    "status": "string",
    "trackingId": "string"
  }
}
```

```
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[eventTracker](#)

이벤트 트래커를 설명하는 객체입니다.

유형: [EventTracker](#) 객체

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)

- [AWS 루비 V3용 SDK](#)

DescribeFeatureTransformation

서비스: Amazon Personalize

주어진 특성 변환에 대해 설명합니다.

구문 요청

```
{
  "featureTransformationArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

featureTransformationArn

설명할 특성 변환의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 항목 여부: 예

응답 구문

```
{
  "featureTransformation": {
    "creationDateTime": number,
    "defaultParameters": {
      "string" : "string"
    },
    "featureTransformationArn": "string",
    "lastUpdatedDateTime": number,
    "name": "string",
    "status": "string"
  }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[featureTransformation](#)

FeatureTransformation 속성 목록입니다.

유형: [FeatureTransformation](#) 객체

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeFilter

서비스: Amazon Personalize

필터의 속성을 설명합니다.

구문 요청

```
{
  "filterArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

filterArn

설명할 필터의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 항목 여부: 예

응답 구문

```
{
  "filter": {
    "creationDateTime": number,
    "datasetGroupArn": "string",
    "failureReason": "string",
    "filterArn": "string",
    "filterExpression": "string",
    "lastUpdatedDateTime": number,
    "name": "string",
    "status": "string"
  }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

filter

필터 세부 정보입니다.

유형: [Filter](#) 객체

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeMetricAttribution

서비스: Amazon Personalize

지표 어트리뷰션을 설명합니다.

구문 요청

```
{
  "metricAttributionArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

metricAttributionArn

지표 어트리뷰션의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 항목 여부: 예

응답 구문

```
{
  "metricAttribution": {
    "creationDateTime": number,
    "datasetGroupArn": "string",
    "failureReason": "string",
    "lastUpdatedDateTime": number,
    "metricAttributionArn": "string",
    "metricsOutputConfig": {
      "roleArn": "string",
      "s3DataDestination": {
        "kmsKeyArn": "string",
        "path": "string"
      }
    }
  }
}
```

```
    },  
    "name": "string",  
    "status": "string"  
  }  
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[metricAttribution](#)

지표 어트리뷰션의 세부 정보입니다.

유형: [MetricAttribution](#) 객체

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)

- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeRecipe

서비스: Amazon Personalize

레시피를 설명합니다.

레시피에는 다음 세 가지 항목이 포함되어 있습니다.

- 모델을 학습시키는 알고리즘.
- 학습을 제어하는 하이퍼파라미터.
- 학습 전에 입력 데이터를 수정하기 위한 특성 변환 정보입니다.

Personalize는 미리 정의된 레시피 세트를 제공합니다. [CreateSolution](#) API로 솔루션을 만들 때 레시피를 지정합니다. [CreateSolution](#) 지정된 레시피의 알고리즘과 훈련 데이터세트를 사용하여 모델을 훈련시킵니다. 솔루션을 캠페인으로 배포하면 [GetRecommendations](#) API를 사용하여 권장 사항을 제공할 수 있습니다.

구문 요청

```
{
  "recipeArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[recipeArn](#)

설명할 레시피의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 항목 여부: 예

응답 구문

```
{
```

```
"recipe": {
  "algorithmArn": "string",
  "creationDateTime": number,
  "description": "string",
  "featureTransformationArn": "string",
  "lastUpdatedDateTime": number,
  "name": "string",
  "recipeArn": "string",
  "recipeType": "string",
  "status": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[recipe](#)

레시피를 설명하는 객체입니다.

유형: [Recipe](#) 객체

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeRecommender

서비스: Amazon Personalize

해당 추천자의 상태를 포함하여 해당 추천자에 대해 설명합니다.

추천자는 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- STOP PENDING > STOP IN_PROGRESS > INACTIVE > START PENDING > START IN_PROGRESS > ACTIVE
- DELETE PENDING > DELETE IN_PROGRESS

status가 CREATE FAILED인 경우 응답에는 이유를 설명하는 failureReason키가 포함됩니다.

추천자가 생성되거나 삭제 중일 때 modelMetrics키는 null입니다.

추천자에 대한 자세한 내용은 을 참조하십시오 [CreateRecommender](#).

구문 요청

```
{
  "recommenderArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[recommenderArn](#)

설명할 추천자의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 항목 여부: 예

응답 구문

```

{
  "recommender": {
    "creationDateTime": number,
    "datasetGroupArn": "string",
    "failureReason": "string",
    "lastUpdatedDateTime": number,
    "latestRecommenderUpdate": {
      "creationDateTime": number,
      "failureReason": "string",
      "lastUpdatedDateTime": number,
      "recommenderConfig": {
        "enableMetadataWithRecommendations": boolean,
        "itemExplorationConfig": {
          "string" : "string"
        },
        "minRecommendationRequestsPerSecond": number,
        "trainingDataConfig": {
          "excludedDatasetColumns": {
            "string" : [ "string" ]
          }
        }
      }
    },
    "status": "string"
  },
  "modelMetrics": {
    "string" : number
  },
  "name": "string",
  "recipeArn": "string",
  "recommenderArn": "string",
  "recommenderConfig": {
    "enableMetadataWithRecommendations": boolean,
    "itemExplorationConfig": {
      "string" : "string"
    },
    "minRecommendationRequestsPerSecond": number,
    "trainingDataConfig": {
      "excludedDatasetColumns": {
        "string" : [ "string" ]
      }
    }
  },
},

```

```
    "status": "string"  
  }  
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

recommender

추천자의 속성.

유형: Recommender 객체

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)

- [AWS PHP V3용 SDK](#)
- [AWS 파이썬용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeSchema

서비스: Amazon Personalize

스키마를 설명합니다. 스키마에 대한 자세한 내용은 [을 참조하십시오 CreateSchema](#).

구문 요청

```
{
  "schemaArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

schemaArn

검색할 스키마의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*+

필수 항목 여부: 예

응답 구문

```
{
  "schema": {
    "creationDateTime": number,
    "domain": "string",
    "lastUpdatedDateTime": number,
    "name": "string",
    "schema": "string",
    "schemaArn": "string"
  }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

schema

요청 받은 스키마.

유형: [DatasetSchema](#) 객체

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeSolution

서비스: Amazon Personalize

솔루션을 설명합니다. 솔루션에 대한 자세한 내용은 [CreateSolution](#)을 참조하십시오.

구문 요청

```
{
  "solutionArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

solutionArn

설명할 솔루션의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 항목 여부: 예

응답 구문

```
{
  "solution": {
    "autoMLResult": {
      "bestRecipeArn": "string"
    },
    "creationDateTime": number,
    "datasetGroupArn": "string",
    "eventType": "string",
    "lastUpdatedDateTime": number,
    "latestSolutionVersion": {
      "creationDateTime": number,
      "failureReason": "string",
      "lastUpdatedDateTime": number,
      "solutionVersionArn": "string",
    }
  }
}
```



```
"status": "string",
"trainingMode": "string",
"trainingType": "string"
},
"name": "string",
"performAutoML": boolean,
"performAutoTraining": boolean,
"performHPO": boolean,
"recipeArn": "string",
"solutionArn": "string",
"solutionConfig": {
  "algorithmHyperParameters": {
    "string" : "string"
  },
  "autoMLConfig": {
    "metricName": "string",
    "recipeList": [ "string" ]
  },
  "autoTrainingConfig": {
    "schedulingExpression": "string"
  },
  "eventValueThreshold": "string",
  "featureTransformationParameters": {
    "string" : "string"
  },
  "hpoConfig": {
    "algorithmHyperParameterRanges": {
      "categoricalHyperParameterRanges": [
        {
          "name": "string",
          "values": [ "string" ]
        }
      ],
      "continuousHyperParameterRanges": [
        {
          "maxValue": number,
          "minValue": number,
          "name": "string"
        }
      ],
      "integerHyperParameterRanges": [
        {
          "maxValue": number,
          "minValue": number,
```

```

        "name": "string"
      }
    ]
  },
  "hpoObjective": {
    "metricName": "string",
    "metricRegex": "string",
    "type": "string"
  },
  "hpoResourceConfig": {
    "maxNumberOfTrainingJobs": "string",
    "maxParallelTrainingJobs": "string"
  }
},
"optimizationObjective": {
  "itemAttribute": "string",
  "objectiveSensitivity": "string"
},
"trainingDataConfig": {
  "excludedDatasetColumns": {
    "string" : [ "string" ]
  }
}
},
"status": "string"
}
}

```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

solution

솔루션을 설명하는 객체입니다.

유형: [Solution](#) 객체

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS 파이썬용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeSolutionVersion

서비스: Amazon Personalize

솔루션의 특정 버전을 설명합니다. 솔루션에 대한 자세한 내용은 을 참조하십시오. [CreateSolution](#)

구문 요청

```
{
  "solutionVersionArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[solutionVersionArn](#)

솔루션 버전의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 항목 여부: 예

응답 구문

```
{
  "solutionVersion": {
    "creationDateTime": number,
    "datasetGroupArn": "string",
    "eventType": "string",
    "failureReason": "string",
    "lastUpdatedDateTime": number,
    "name": "string",
    "performAutoML": boolean,
    "performHPO": boolean,
    "recipeArn": "string",
    "solutionArn": "string",
    "solutionConfig": {
      "algorithmHyperParameters": {
```

```
    "string" : "string"
  },
  "autoMLConfig": {
    "metricName": "string",
    "recipeList": [ "string" ]
  },
  "autoTrainingConfig": {
    "schedulingExpression": "string"
  },
  "eventValueThreshold": "string",
  "featureTransformationParameters": {
    "string" : "string"
  },
  "hpoConfig": {
    "algorithmHyperParameterRanges": {
      "categoricalHyperParameterRanges": [
        {
          "name": "string",
          "values": [ "string" ]
        }
      ],
      "continuousHyperParameterRanges": [
        {
          "maxValue": number,
          "minValue": number,
          "name": "string"
        }
      ],
      "integerHyperParameterRanges": [
        {
          "maxValue": number,
          "minValue": number,
          "name": "string"
        }
      ]
    }
  },
  "hpoObjective": {
    "metricName": "string",
    "metricRegex": "string",
    "type": "string"
  },
  "hpoResourceConfig": {
    "maxNumberOfTrainingJobs": "string",
    "maxParallelTrainingJobs": "string"
  }
}
```


ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

GetSolutionMetrics

서비스: Amazon Personalize

지정된 솔루션 버전의 지표를 가져옵니다.

구문 요청

```
{  
  "solutionVersionArn": "string"  
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

solutionVersionArn

지표를 가져올 솔루션 버전의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*+

필수 항목 여부: 예

응답 구문

```
{  
  "metrics": {  
    "string" : number  
  },  
  "solutionVersionArn": "string"  
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

metrics

솔루션 버전에 대한 지표. 자세한 내용은 [지표를 솔루션 버전 평가 단원을 참조하세요.](#)

유형: 문자열-double 맵

맵 항목: 최대 항목 수 100개.

키 길이 제약: 최대 길이 256.

solutionVersionArn

요청에서 지정한 것과 동일한 솔루션 버전 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*+

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)

- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListBatchInferenceJobs

서비스: Amazon Personalize

솔루션 버전에서 수행된 배치 추론 작업의 목록을 가져옵니다.

구문 요청

```
{
  "maxResults": number,
  "nextToken": "string",
  "solutionVersionArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

maxResults

각 페이지에 반환되는 배치 추론 작업 결과의 최대 수. 기본 값은 100입니다.

타입: 정수

유효 범위: 최소값 1. 최대값 100.

필수 여부: 아니요

nextToken

결과에 대한 다음 페이지를 요청하기 위한 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: $\backslash p\{ASCII\}\{0,1500\}$

Required: No

solutionVersionArn

배치 추론 작업이 생성된 솔루션 버전의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

필수 여부: 아니요

응답 구문

```
{
  "batchInferenceJobs": [
    {
      "batchInferenceJobArn": "string",
      "batchInferenceJobMode": "string",
      "creationDateTime": number,
      "failureReason": "string",
      "jobName": "string",
      "lastUpdatedDateTime": number,
      "solutionVersionArn": "string",
      "status": "string"
    }
  ],
  "nextToken": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[batchInferenceJobs](#)

반환되는 각 작업에 대한 정보가 들어 있는 목록입니다.

유형: [BatchInferenceJobSummary](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

[nextToken](#)

결과에 대한 다음 페이지를 가져오기 위한 토큰. 값은 반환할 결과가 더 이상 없는 경우 null입니다.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: `\p{ASCII}{0,1500}`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

InvalidNextTokenException

토큰이 유효하지 않습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListBatchSegmentJobs

서비스: Amazon Personalize

지정한 솔루션 버전에서 수행된 배치 세그먼트 작업의 목록을 가져옵니다.

구문 요청

```
{
  "maxResults": number,
  "nextToken": "string",
  "solutionVersionArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[maxResults](#)

각 페이지에 반환되는 배치 세그먼트 작업 결과의 최대 수. 기본 값은 100입니다.

타입: 정수

유효 범위: 최소값 1. 최대값 100.

필수 여부: 아니요

[nextToken](#)

결과에 대한 다음 페이지를 요청하기 위한 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: `\p{ASCII}{0,1500}`

Required: No

[solutionVersionArn](#)

배치 세그먼트 작업이 배치 세그먼트를 생성하는 데 사용한 솔루션 버전의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*+

필수 여부: 아니요

응답 구문

```
{
  "batchSegmentJobs": [
    {
      "batchSegmentJobArn": "string",
      "creationDateTime": number,
      "failureReason": "string",
      "jobName": "string",
      "lastUpdatedDateTime": number,
      "solutionVersionArn": "string",
      "status": "string"
    }
  ],
  "nextToken": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[batchSegmentJobs](#)

반환되는 각 작업에 대한 정보가 들어 있는 목록입니다.

유형: [BatchSegmentJobSummary](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

[nextToken](#)

결과에 대한 다음 페이지를 가져오기 위한 토큰. 값은 반환할 결과가 더 이상 없는 경우 null입니다.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: `\p{ASCII}{0,1500}`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

InvalidNextTokenException

토큰이 유효하지 않습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListCampaigns

서비스: Amazon Personalize

지정된 솔루션을 사용하는 캠페인의 목록을 반환합니다. 솔루션이 지정되지 않은 경우 해당 계정과 관련된 모든 캠페인이 나열됩니다. 응답은 리소스 이름(ARN)을 비롯하여 각 캠페인의 속성을 제공합니다. 캠페인에 대한 자세한 내용은 을 참조하십시오 [CreateCampaign](#).

구문 요청

```
{
  "maxResults": number,
  "nextToken": "string",
  "solutionArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[maxResults](#)

반환할 캠페인의 최대 수.

타입: 정수

유효 범위: 최소값 1. 최대값 100.

필수 여부: 아니요

[nextToken](#)

이전 호출에서 다음 캠페인 세트 (있는 경우) 를 가져오기 [ListCampaigns](#) 위해 반환된 토큰입니다.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: `\p{ASCII}{0,1500}`

Required: No

[solutionArn](#)

캠페인을 나열할 솔루션의 리소스 이름(ARN)입니다. 솔루션이 지정되지 않은 경우 해당 계정과 관련된 모든 캠페인이 나열됩니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

필수 여부: 아니요

응답 구문

```
{
  "campaigns": [
    {
      "campaignArn": "string",
      "creationDateTime": number,
      "failureReason": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "status": "string"
    }
  ],
  "nextToken": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

campaigns

캠페인의 목록입니다.

유형: [CampaignSummary](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

nextToken

다음 번 캠페인 세트를 가져오기 위한 토큰(있는 경우).

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: `\p{ASCII}{0,1500}`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

InvalidNextTokenException

토큰이 유효하지 않습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListDataDeletionJobs

서비스: Amazon Personalize

데이터셋 그룹의 데이터 삭제 작업 목록을 생성 시간별로 정렬하여 가장 최근의 작업부터 반환합니다. 데이터셋 그룹을 지정하지 않으면 계정과 관련된 모든 데이터 삭제 작업이 나열됩니다. 응답은 Amazon 리소스 이름 (ARN) 을 포함하여 각 작업에 대한 속성을 제공합니다. 데이터 삭제 작업에 대한 자세한 내용은 [사용자](#) 삭제를 참조하십시오.

구문 요청

```
{
  "datasetGroupArn": "string",
  "maxResults": number,
  "nextToken": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

datasetGroupArn

데이터 삭제 작업을 나열할 데이터셋 그룹의 Amazon 리소스 이름 (ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

Required: No

maxResults

반환할 최대 데이터 삭제 작업 수입니다.

타입: 정수

유효 범위: 최소값 1. 최대값 100.

필수 여부: 아니요

nextToken

이전 호출에서 다음 작업 세트 (있는 경우) 를 가져오기 ListDataDeletionJobs 위해 반환된 토큰입니다.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: \p{ASCII}{0,1500}

필수 여부: 아니요

응답 구문

```
{
  "dataDeletionJobs": [
    {
      "creationDateTime": number,
      "dataDeletionJobArn": "string",
      "datasetGroupArn": "string",
      "failureReason": "string",
      "jobName": "string",
      "lastUpdatedDateTime": number,
      "status": "string"
    }
  ],
  "nextToken": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

dataDeletionJobs

데이터 삭제 작업 목록.

유형: [DataDeletionJobSummary](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

nextToken

다음 데이터 삭제 작업 세트를 가져오기 위한 토큰입니다 (있는 경우).

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: `\p{ASCII}{0,1500}`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

InvalidNextTokenException

토큰이 유효하지 않습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListDatasetExportJobs

서비스: Amazon Personalize

지정된 데이터세트를 사용하는 데이터세트 내보내기 작업의 목록을 반환합니다. 데이터세트를 지정하지 않으면 해당 계정과 관련된 모든 데이터세트 내보내기 작업이 나열됩니다. 응답은 리소스 이름(ARN)을 비롯하여 각 데이터세트 내보내기 작업의 속성을 제공합니다. 데이터세트 내보내기 작업에 대한 자세한 내용은 [참조하십시오](#) [CreateDatasetExportJob](#). 데이터세트에 대한 자세한 내용은 [참조하십시오](#) [CreateDataset](#)

구문 요청

```
{
  "datasetArn": "string",
  "maxResults": number,
  "nextToken": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[datasetArn](#)

데이터세트 내보내기 작업을 나열할 데이터세트의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

Required: No

[maxResults](#)

반환할 데이터세트 내보내기 작업의 최대 수.

타입: 정수

유효 범위: 최소값 1. 최대값 100.

필수 여부: 아니요

nextToken

다음 번 데이터세트 내보내기 작업 세트(있는 경우)를 가져오기 위해 이전에 `ListDatasetExportJobs`을 호출한 후 반환된 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: `\p{ASCII}{0,1500}`

필수 여부: 아니요

응답 구문

```
{
  "datasetExportJobs": [
    {
      "creationDateTime": number,
      "datasetExportJobArn": "string",
      "failureReason": "string",
      "jobName": "string",
      "lastUpdatedDateTime": number,
      "status": "string"
    }
  ],
  "nextToken": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

datasetExportJobs

데이터세트 내보내기 작업의 목록입니다.

유형: [DatasetExportJobSummary](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

nextToken

다음 번 데이터세트 내보내기 작업 세트(있는 경우)를 가져오기 위한 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: `\p{ASCII}{0,1500}`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

InvalidNextTokenException

토큰이 유효하지 않습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하세요.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListDatasetGroups

서비스: Amazon Personalize

데이터세트 그룹의 목록을 반환합니다. 응답은 리소스 이름(ARN)을 비롯하여 각 데이터세트 그룹의 속성을 제공합니다. 데이터세트 그룹에 대한 자세한 내용은 을 참조하십시오 [CreateDatasetGroup](#).

구문 요청

```
{
  "maxResults": number,
  "nextToken": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[maxResults](#)

반환할 최대 데이터세트 그룹 수.

타입: 정수

유효 범위: 최소값 1. 최대값 100.

필수 여부: 아니요

[nextToken](#)

다음 번 데이터세트 그룹 세트(있는 경우)를 가져오기 위해 이전에 ListDatasetGroups을 호출한 후 반환된 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: $\backslash p\{ASCII\}\{0,1500\}$

필수 여부: 아니요

응답 구문

```
{
```

```
"datasetGroups": [  
  {  
    "creationDateTime": number,  
    "datasetGroupArn": "string",  
    "domain": "string",  
    "failureReason": "string",  
    "lastUpdatedDateTime": number,  
    "name": "string",  
    "status": "string"  
  }  
],  
"nextToken": "string"  
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[datasetGroups](#)

데이터세트 그룹의 목록입니다.

유형: [DatasetGroupSummary](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

[nextToken](#)

다음 번 데이터세트 그룹 세트(있는 경우)를 가져오기 위한 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: `\p{ASCII}{0,1500}`

Errors

InvalidNextTokenException

토큰이 유효하지 않습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하세요.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListDatasetImportJobs

서비스: Amazon Personalize

지정된 데이터세트를 사용하는 데이터세트 가져오기 작업의 목록을 반환합니다. 데이터세트를 지정하지 않으면 해당 계정과 관련된 모든 데이터세트 가져오기 작업이 나열됩니다. 응답은 리소스 이름(ARN)을 비롯하여 각 데이터세트 가져오기 작업의 속성을 제공합니다. 데이터세트 가져오기 작업에 대한 자세한 내용은 [참조하십시오](#) [CreateDatasetImportJob](#). 데이터세트에 대한 자세한 내용은 [참조하십시오](#) [CreateDataset](#)

구문 요청

```
{
  "datasetArn": "string",
  "maxResults": number,
  "nextToken": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[datasetArn](#)

데이터세트 가져오기 작업을 나열할 데이터세트의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

Required: No

[maxResults](#)

반환할 데이터세트 가져오기 작업의 최대 수.

타입: 정수

유효 범위: 최소값 1. 최대값 100.

필수 여부: 아니요

nextToken

다음 번 데이터세트 가져오기 작업 세트(있는 경우)를 가져오기 위해 이전에 ListDatasetImportJobs을 호출한 후 반환된 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: \p{ASCII}{0,1500}

필수 여부: 아니요

응답 구문

```
{
  "datasetImportJobs": [
    {
      "creationDateTime": number,
      "datasetImportJobArn": "string",
      "failureReason": "string",
      "importMode": "string",
      "jobName": "string",
      "lastUpdatedDateTime": number,
      "status": "string"
    }
  ],
  "nextToken": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

datasetImportJobs

데이터세트 가져오기 작업의 목록입니다.

유형: [DatasetImportJobSummary](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

nextToken

다음 번 데이터세트 가져오기 작업 세트(있는 경우)를 가져오기 위한 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: `\p{ASCII}{0,1500}`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

InvalidNextTokenException

토큰이 유효하지 않습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하세요.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListDatasets

서비스: Amazon Personalize

지정된 데이터 세트 그룹에 포함된 데이터 세트의 목록을 반환합니다. 응답은 Amazon 리소스 이름 (ARN)을 비롯하여 각 데이터 세트의 속성을 제공합니다. 데이터세트에 대한 자세한 내용은 [참조하십시오](#) [CreateDataset](#).

구문 요청

```
{
  "datasetGroupArn": "string",
  "maxResults": number,
  "nextToken": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[datasetGroupArn](#)

나열할 데이터 세트가 포함된 데이터 세트 그룹의 Amazon 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*+

Required: No

[maxResults](#)

반환할 최대 데이터 세트 수.

타입: 정수

유효 범위: 최소값 1. 최대값 100.

필수 여부: 아니요

[nextToken](#)

다음 번 데이터세트 가져오기 작업 세트(있는 경우)를 가져오기 위해 이전에 ListDatasets을 호출한 후 반환된 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: `\p{ASCII}{0,1500}`

필수 여부: 아니요

응답 구문

```
{
  "datasets": [
    {
      "creationDateTime": number,
      "datasetArn": "string",
      "datasetType": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "status": "string"
    }
  ],
  "nextToken": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

datasets

Dataset 객체 어레이. 각 객체는 메타데이터 정보를 제공합니다.

유형: [DatasetSummary](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

nextToken

다음 번 데이터 세트 집합(있는 경우)을 가져오기 위한 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: `\p{ASCII}{0,1500}`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

InvalidNextTokenException

토큰이 유효하지 않습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하세요.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS 파이썬용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListEventTrackers

서비스: Amazon Personalize

해당 계정과 관련된 이벤트 트래커의 목록을 반환합니다. 응답은 리소스 이름(ARN) 과 추적 ID를 비롯하여 각 이벤트 트래커의 속성을 제공합니다. 이벤트 트래커에 대한 자세한 내용은 [CreateEventTracker](#)를 참조하십시오.

구문 요청

```
{
  "datasetGroupArn": "string",
  "maxResults": number,
  "nextToken": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[datasetGroupArn](#)

응답을 필터링하는 데 사용되는 데이터세트 그룹의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

Required: No

[maxResults](#)

반환할 최대 이벤트 트래커 수

타입: 정수

유효 범위: 최소값 1. 최대값 100.

필수 여부: 아니요

[nextToken](#)

다음 번 이벤트 트래커 세트(있는 경우)를 가져오기 위해 이전에 ListEventTrackers을 호출한 후 반환된 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: `\p{ASCII}{0,1500}`

필수 여부: 아니요

응답 구문

```
{
  "eventTrackers": [
    {
      "creationDateTime": number,
      "eventTrackerArn": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "status": "string"
    }
  ],
  "nextToken": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[eventTrackers](#)

이벤트 트래커의 목록입니다.

유형: [EventTrackerSummary](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

[nextToken](#)

다음 번 이벤트 트래커 세트(있는 경우)을 가져오기 위한 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: \p{ASCII}{0,1500}

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

InvalidNextTokenException

토큰이 유효하지 않습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListFilters

서비스: Amazon Personalize

지정된 데이터세트 그룹에 속하는 모든 필터를 나열합니다.

구문 요청

```
{
  "datasetGroupArn": "string",
  "maxResults": number,
  "nextToken": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

datasetGroupArn

해당 필터가 포함된 데이터세트 그룹의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]):personalize:.*:.*:.*

Required: No

maxResults

반환할 최대 필터 수.

타입: 정수

유효 범위: 최소값 1. 최대값 100.

필수 여부: 아니요

nextToken

다음 번 필터 세트(있는 경우)를 가져오기 위해 이전에 ListFilters을 호출한 후 반환된 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: `\p{ASCII}{0,1500}`

필수 여부: 아니요

응답 구문

```
{
  "Filters": [
    {
      "creationDateTime": number,
      "datasetGroupArn": "string",
      "failureReason": "string",
      "filterArn": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "status": "string"
    }
  ],
  "nextToken": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Filters

반환된 필터의 목록입니다.

유형: [FilterSummary](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

nextToken

다음 번 필터 세트(있는 경우)를 가져오기 위한 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: \p{ASCII}{0,1500}

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

InvalidNextTokenException

토큰이 유효하지 않습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListMetricAttributionMetrics

서비스: Amazon Personalize

해당 지표 어트리뷰션에 대한 지표를 나열합니다.

구문 요청

```
{
  "maxResults": number,
  "metricAttributionArn": "string",
  "nextToken": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

maxResults

한 결과 페이지에 반환할 수 있는 최대 지표 수.

타입: 정수

유효 범위: 최소값 1. 최대값 100.

필수 여부: 아니요

metricAttributionArn

속성을 검색할 지표 어트리뷰션의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

Required: No

nextToken

다음 번 결과 페이지를 검색하기 위한 이전 요청의 페이지 매김 토큰을 지정합니다.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: \p{ASCII}{0,1500}

필수 여부: 아니요

응답 구문

```
{
  "metrics": [
    {
      "eventType": "string",
      "expression": "string",
      "metricName": "string"
    }
  ],
  "nextToken": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

metrics

지정된 지표 어트리뷰션에 대한 지표.

유형: [MetricAttribute](#) 객체 어레이

어레이 멤버: 최대 항목 수 10개.

nextToken

다음 번 결과 페이지를 검색하기 위한 이전 ListMetricAttributionMetricsResponse 요청의 페이지 매김 토큰을 지정합니다.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: \p{ASCII}{0,1500}

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

InvalidNextTokenException

토큰이 유효하지 않습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListMetricAttributions

서비스: Amazon Personalize

지표 어트리뷰션을 나열합니다.

구문 요청

```
{
  "datasetGroupArn": "string",
  "maxResults": number,
  "nextToken": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

datasetGroupArn

지표 어트리뷰션의 데이터세트 그룹의 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*+

Required: No

maxResults

한 결과 페이지에 반환할 수 있는 최대 지표 어트리뷰션 수.

타입: 정수

유효 범위: 최소값 1. 최대값 100.

필수 여부: 아니요

nextToken

다음 번 결과 페이지를 검색하기 위한 이전 요청의 페이지 매김 토큰을 지정합니다.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: `\p{ASCII}{0,1500}`

필수 여부: 아니요

응답 구문

```
{
  "metricAttributions": [
    {
      "creationDateTime": number,
      "failureReason": "string",
      "lastUpdatedDateTime": number,
      "metricAttributionArn": "string",
      "name": "string",
      "status": "string"
    }
  ],
  "nextToken": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

metricAttributions

지표 어트리뷰션의 목록입니다.

유형: [MetricAttributionSummary](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

nextToken

다음 번 결과 페이지를 검색하기 위한 이전 요청의 페이지 매김 토큰을 지정합니다.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: \p{ASCII}{0,1500}

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

InvalidNextTokenException

토큰이 유효하지 않습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListRecipes

서비스: Amazon Personalize

사용 가능한 레시피의 목록을 반환합니다. 응답은 레시피의 리소스 이름(ARN)을 비롯하여 각 레시피의 속성을 제공합니다.

구문 요청

```
{  
  "domain": "string",  
  "maxResults": number,  
  "nextToken": "string",  
  "recipeProvider": "string"  
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

domain

도메인 데이터세트 그룹의 도메인별로 반환된 레시피를 필터링합니다. 이 도메인의 레시피(도메인 데이터세트 그룹 사용 사례)만 응답에 포함됩니다. 도메인을 지정하지 않으면 모든 레시피가 반환됩니다.

타입: 문자열

유효 값: ECOMMERCE | VIDEO_ON_DEMAND

필수 여부: 아니요

maxResults

반환할 최대 레시피 수.

타입: 정수

유효 범위: 최소값 1. 최대값 100.

필수 여부: 아니요

nextToken

다음 번 레시피 세트(있는 경우)를 가져오기 위해 이전에 ListRecipes을 호출한 후 반환된 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: `\p{ASCII}{0,1500}`

Required: No

[recipeProvider](#)

기본값은 SERVICE입니다.

타입: 문자열

유효 값: SERVICE

필수 항목 여부: 아니요

응답 구문

```
{
  "nextToken": "string",
  "recipes": [
    {
      "creationDateTime": number,
      "domain": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "recipeArn": "string",
      "status": "string"
    }
  ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[nextToken](#)

다음 번 레시피 세트를 가져오기 위한 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: `\p{ASCII}{0,1500}`

[recipes](#)

사용 가능한 레시피의 목록입니다.

유형: [RecipeSummary](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

InvalidNextTokenException

토큰이 유효하지 않습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)

- [AWS 루비 V3용 SDK](#)

ListRecommenders

서비스: Amazon Personalize

지정된 도메인 데이터세트 그룹의 추천자 목록을 반환합니다. 도메인 데이터세트 그룹을 지정하지 않으면 해당 계정과 연결된 모든 추천자가 나열됩니다. 응답은 리소스 이름(ARN)을 비롯하여 각 추천자의 속성을 제공합니다. 추천자에 대한 자세한 내용은 [CreateRecommender](#)를 참조하십시오.

구문 요청

```
{
  "datasetGroupArn": "string",
  "maxResults": number,
  "nextToken": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

datasetGroupArn

추천자를 나열할 도메인 데이터세트 그룹의 리소스 이름(ARN)입니다. 도메인 데이터세트 그룹을 지정하지 않으면 해당 계정과 연결된 모든 추천자가 나열됩니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

Required: No

maxResults

반환할 최대 추천자 수.

타입: 정수

유효 범위: 최소값 1. 최대값 100.

필수 여부: 아니요

nextToken

다음 번 추천자 세트(있는 경우)를 가져오기 위해 이전에 ListRecommenders를 호출한 후 반환된 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: \p{ASCII}{0,1500}

필수 여부: 아니요

응답 구문

```
{
  "nextToken": "string",
  "recommenders": [
    {
      "creationDateTime": number,
      "datasetGroupArn": "string",
      "lastUpdatedDateTime": number,
      "name": "string",
      "recipeArn": "string",
      "recommenderArn": "string",
      "recommenderConfig": {
        "enableMetadataWithRecommendations": boolean,
        "itemExplorationConfig": {
          "string" : "string"
        },
        "minRecommendationRequestsPerSecond": number,
        "trainingDataConfig": {
          "excludedDatasetColumns": {
            "string" : [ "string" ]
          }
        }
      },
      "status": "string"
    }
  ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

nextToken

다음 번 추천자 세트(있는 경우)를 가져오기 위한 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: `\p{ASCII}{0,1500}`

recommenders

추천자의 목록입니다.

유형: [RecommenderSummary](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

InvalidNextTokenException

토큰이 유효하지 않습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)

- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS 파이썬용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListSchemas

서비스: Amazon Personalize

해당 계정에 연결된 스키마의 목록을 반환합니다. 응답은 리소스 이름(ARN)을 비롯하여 각 스키마의 속성을 제공합니다. 스키마에 대한 자세한 내용은 [이 링크](#)를 참조하십시오 [CreateSchema](#).

구문 요청

```
{
  "maxResults": number,
  "nextToken": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[maxResults](#)

반환할 최대 스키마 수.

타입: 정수

유효 범위: 최소값 1. 최대값 100.

필수 여부: 아니요

[nextToken](#)

다음 번 스키마 세트(있는 경우)를 가져오기 위해 이전에 ListSchemas을 호출한 후 반환된 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: $\backslash p\{ASCII\}\{0,1500\}$

필수 여부: 아니요

응답 구문

```
{
```

```
"nextToken": "string",
"schemas": [
  {
    "creationDateTime": number,
    "domain": "string",
    "lastUpdatedDateTime": number,
    "name": "string",
    "schemaArn": "string"
  }
]
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

nextToken

다음 번 스키마 세트(있는 경우)를 가져오기 위한 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: \p{ASCII}{0,1500}

schemas

스키마의 목록입니다.

유형: [DatasetSchemaSummary](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

Errors

InvalidNextTokenException

토큰이 유효하지 않습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListSolutions

서비스: Amazon Personalize

지정된 데이터셋 그룹의 솔루션 목록을 반환합니다. 데이터셋 그룹을 지정하지 않으면 해당 계정과 관련된 모든 솔루션이 나열됩니다. 응답은 Amazon 리소스 이름(ARN)을 비롯하여 각 솔루션의 속성을 제공합니다. 솔루션에 대한 자세한 내용은 [참조하십시오](#) [CreateSolution](#).

구문 요청

```
{
  "datasetGroupArn": "string",
  "maxResults": number,
  "nextToken": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[datasetGroupArn](#)

데이터셋 그룹의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

Required: No

[maxResults](#)

반환할 최대 솔루션 수.

타입: 정수

유효 범위: 최소값 1. 최대값 100.

필수 여부: 아니요

[nextToken](#)

다음번 솔루션 세트(있는 경우)를 가져오기 위해 이전에 ListSolutions을 호출한 후 반환된 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: \p{ASCII}{0,1500}

필수 여부: 아니요

응답 구문

```
{
  "nextToken": "string",
  "solutions": [
    {
      "creationDateTime": number,
      "lastUpdatedDateTime": number,
      "name": "string",
      "recipeArn": "string",
      "solutionArn": "string",
      "status": "string"
    }
  ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

nextToken

다음번 솔루션 세트(있는 경우)를 가져오기 위한 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: \p{ASCII}{0,1500}

solutions

현재 솔루션의 목록입니다.

유형: [SolutionSummary](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

InvalidNextTokenException

토큰이 유효하지 않습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListSolutionVersions

서비스: Amazon Personalize

해당 솔루션의 솔루션 버전 목록을 반환합니다. 솔루션이 지정되지 않은 경우 해당 계정과 관련된 모든 솔루션 버전이 나열됩니다. 응답은 Amazon 리소스 이름(ARN)을 비롯하여 각 솔루션 버전의 속성을 제공합니다.

구문 요청

```
{
  "maxResults": number,
  "nextToken": "string",
  "solutionArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[maxResults](#)

반환할 최대 솔루션 버전 수.

타입: 정수

유효 범위: 최소값 1. 최대값 100.

필수 여부: 아니요

[nextToken](#)

다음번 솔루션 버전 세트(있는 경우)를 가져오기 위해 이전에 ListSolutionVersions을 호출한 후 반환된 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: $\backslash p\{ASCII\}\{0,1500\}$

Required: No

[solutionArn](#)

솔루션의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

필수 여부: 아니요

응답 구문

```
{
  "nextToken": "string",
  "solutionVersions": [
    {
      "creationDateTime": number,
      "failureReason": "string",
      "lastUpdatedDateTime": number,
      "solutionVersionArn": "string",
      "status": "string",
      "trainingMode": "string",
      "trainingType": "string"
    }
  ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

nextToken

다음번 솔루션 버전 세트(있는 경우)를 가져오기 위한 토큰.

타입: 문자열

길이 제약: 최대 길이 1500.

패턴: `\p{ASCII}{0,1500}`

solutionVersions

버전 속성을 설명하는 솔루션 버전의 목록입니다.

유형: [SolutionVersionSummary](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

InvalidNextTokenException

토큰이 유효하지 않습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListTagsForResource

서비스: Amazon Personalize

리소스에 연결된 [태그](#)의 목록을 가져옵니다.

구문 요청

```
{
  "resourceArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[resourceArn](#)

리소스의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*+

필수 항목 여부: 예

응답 구문

```
{
  "tags": [
    {
      "tagKey": "string",
      "tagValue": "string"
    }
  ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

tags

리소스의 태그.

유형: [Tag](#) 객체 어레이

어레이 멤버: 최소 항목 수 0개. 최대 항목 수 200개.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)

- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

StartRecommender

서비스: Amazon Personalize

상태가 INACTIVE인 추천자를 시작합니다. 추천자를 시작해도 새 모델이 생성되지는 않지만 추천자에 대한 청구 및 자동 재학습이 재개됩니다.

구문 요청

```
{
  "recommenderArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

recommenderArn

시작할 추천자의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 항목 여부: 예

응답 구문

```
{
  "recommenderArn": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

recommenderArn

시작한 추천자의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

StopRecommender

서비스: Amazon Personalize

ACTIVE 상태의 추천자를 중지합니다. 추천자를 중지하면 해당 추천자에 대한 청구 및 자동 재학습이 중지됩니다.

구문 요청

```
{
  "recommenderArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

recommenderArn

중지할 추천자의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 항목 여부: 예

응답 구문

```
{
  "recommenderArn": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

recommenderArn

중지한 추천자의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

StopSolutionVersionCreation

서비스: Amazon Personalize

상태가 CREATE_PENDING 또는 CREATE_IN_PROGRESS인 솔루션 버전의 생성을 중지합니다.

솔루션 버전의 현재 상태에 따라 솔루션 버전 상태가 다음과 같이 변경됩니다.

- CREATE_PENDING > CREATE_STOPPED

또는

- CREATE_IN_PROGRESS > CREATE_STOPPING > CREATE_STOPPED

솔루션 버전 생성을 중지할 때까지 완료했던 모든 학습에 대한 요금은 청구됩니다. 중지된 후에는 솔루션 버전 생성을 재개할 수 없습니다.

구문 요청

```
{
  "solutionVersionArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

solutionVersionArn

생성을 중지할 솔루션 버전의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

TagResource

서비스: Amazon Personalize

리소스에 태그 목록을 추가합니다.

구문 요청

```
{
  "resourceArn": "string",
  "tags": [
    {
      "tagKey": "string",
      "tagValue": "string"
    }
  ]
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

resourceArn

리소스의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 사항 여부: Yes

tags

리소스에 적용할 태그. 자세한 내용은 [Amazon Personalize 리소스 태그 지정](#)을 참조하십시오.

유형: [Tag](#)객체 어레이

어레이 멤버: 최소 항목 수 0개. 최대 항목 수 200개.

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

LimitExceededException

초당 요청 수 한도를 초과했습니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

TooManyTagsException

이 리소스에 적용할 수 있는 최대 태그 수를 초과했습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)

- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS 파이썬용 SDK](#)
- [AWS 루비 V3용 SDK](#)

UntagResource

서비스: Amazon Personalize

리소스에 연결된 지정된 태그를 제거합니다. 자세한 내용은 [Amazon Personalize 리소스에서 태그 제거를 참조하십시오](#).

구문 요청

```
{
  "resourceArn": "string",
  "tagKeys": [ "string" ]
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

resourceArn

리소스의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*+

필수 사항 여부: Yes

tagKeys

제거할 태그의 키입니다.

유형: 문자열 어레이

어레이 멤버: 최소 항목 수 0개. 최대 항목 수 200개.

길이 제약: 최소 길이 1. 최대 길이 128.

패턴: ^([\p{L}\p{Z}\p{N}_./+=\-\@]*)\$

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

TooManyTagKeysException

요청에 리소스에 연결할 수 있는 것보다 많은 태그 키가 포함되어 있습니다(리소스당 태그 키 50 개).

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)

- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

UpdateCampaign

서비스: Amazon Personalize

기존 캠페인과 함께 재학습된 솔루션 버전을 배포하거나, 캠페인을 변경하거나 `minProvisionedTPS`, 캠페인 구성을 수정하도록 캠페인을 업데이트합니다. 예를 들어 기존 캠페인에 대해 `enableMetadataWithRecommendations true`로 설정할 수 있습니다.

최신 솔루션 버전을 사용하여 자동으로 시작되도록 캠페인을 업데이트하려면 다음을 지정하십시오.

- `SolutionVersionArn` 파라미터에는 솔루션의 Amazon 리소스 이름 (ARN) 을 형식으로 `SolutionArn/$LATEST` 지정합니다.
- `campaignConfig`에서는 `syncWithLatestSolutionVersion` 로 설정합니다. `true`

캠페인을 업데이트하려면 캠페인 상태가 `ACTIVE` 또는 `CREATE FAILED`여야 합니다.

[DescribeCampaign](#) 작업을 사용하여 캠페인 상태를 확인합니다.

Note

업데이트가 진행 중인 동안에도 캠페인에서 추천을 받을 수 있습니다. 캠페인은 최신 캠페인 업데이트 상태가 `Active`가 될 때까지 이전 솔루션 버전과 캠페인 구성을 사용하여 추천을 생성합니다.

코드 샘플을 비롯한 캠페인 업데이트에 대한 자세한 내용은 [캠페인 업데이트](#) 단원을 참조하세요. 캠페인에 대한 자세한 내용은 [캠페인 생성](#) 단원을 참조하세요.

구문 요청

```
{
  "campaignArn": "string",
  "campaignConfig": {
    "enableMetadataWithRecommendations": boolean,
    "itemExplorationConfig": {
      "string" : "string"
    },
    "syncWithLatestSolutionVersion": boolean
  },
  "minProvisionedTPS": number,
  "solutionVersionArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[campaignArn](#)

캠페인의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

필수 여부: 예

[campaignConfig](#)

캠페인의 구성 세부 정보입니다.

유형: [CampaignConfig](#) 객체

필수 여부: 아니요

[minProvisionedTPS](#)

Personalize에서 지원할 초당 요청된 최소 프로비저닝 트랜잭션(추천)을 지정합니다. `minProvisionedTPS`가 높으면 요금이 인상됩니다. `minProvisionedTPS`(기본값)의 경우 1부터 시작하는 것이 좋습니다. Amazon CloudWatch 지표를 사용하여 사용량을 추적하고 `minProvisionedTPS` 필요에 따라 사용량을 늘리십시오.

타입: 정수

유효 범위: 최소값 1.

필수 여부: 아니요

[solutionVersionArn](#)

배포할 새 모델의 Amazon 리소스 이름 (ARN). 솔루션의 최신 솔루션 버전을 지정하려면 솔루션의 ARN을 형식으로 `SolutionArn/$LATEST` 지정하십시오. True에서 `syncWithLatestSolutionVersion` 로 설정한 경우 이 형식을 사용해야 합니다.

[CampaignConfig](#)

솔루션의 최신 솔루션 버전이 아닌 모델을 배포하려면 솔루션 버전의 ARN을 지정하세요.

자동 캠페인 업데이트에 대한 자세한 내용은 [자동 캠페인 업데이트 활성화](#)를 참조하십시오.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

필수 여부: 아니요

응답 구문

```
{
  "campaignArn": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[campaignArn](#)

요청에 제공된 것과 동일한 캠페인 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS 파이썬용 SDK](#)
- [AWS 루비 V3용 SDK](#)

UpdateDataset

서비스: Amazon Personalize

데이터세트를 업데이트하여 스키마를 새 스키마나 기존 스키마로 바꿉니다. 자세한 내용은 [데이터세트 스키마교체](#)를 참조하세요.

구문 요청

```
{
  "datasetArn": "string",
  "schemaArn": "string"
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[datasetArn](#)

업데이트하려는 데이터세트의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 사항 여부: Yes

[schemaArn](#)

사용하려는 새 스키마의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 항목 여부: 예

응답 구문

```
{
```

```
"datasetArn": "string"  
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

datasetArn

업데이트한 데이터세트 그룹의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*+

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)

- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

UpdateMetricAttribution

서비스: Amazon Personalize

지표 어트리뷰션을 업데이트합니다.

구문 요청

```
{
  "addMetrics": [
    {
      "eventType": "string",
      "expression": "string",
      "metricName": "string"
    }
  ],
  "metricAttributionArn": "string",
  "metricsOutputConfig": {
    "roleArn": "string",
    "s3DataDestination": {
      "kmsKeyArn": "string",
      "path": "string"
    }
  },
  "removeMetrics": [ "string" ]
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[addMetrics](#)

지표 어트리뷰션에 새 지표 속성을 추가합니다.

유형: [MetricAttribute](#) 객체 어레이

어레이 멤버: 최대 항목 수 10개.

필수 여부: 아니요

[metricAttributionArn](#)

업데이트할 지표 어트리뷰션의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

[metricsOutputConfig](#)

메트릭 어트리뷰션을 위한 출력 구성.

유형: [MetricAttributionOutput](#) 객체

필수 여부: 아니요

[removeMetrics](#)

지표 어트리뷰션에서 지표 속성을 제거합니다.

유형: 문자열 어레이

어레이 멤버: 최대 항목 수 10개.

길이 제약: 최대 길이 256.

필수 여부: 아니요

응답 구문

```
{
  "metricAttributionArn": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[metricAttributionArn](#)

업데이트한 지표 어트리뷰션에 대한 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceAlreadyExistsException

지정된 리소스가 이미 있습니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)

- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

UpdateRecommender

서비스: Amazon Personalize

추천자 구성을 수정하도록 추천자를 업데이트합니다. 추천자를 업데이트하여 학습에 사용된 열을 수정하면 Personalize가 추천자를 지원하는 모델의 전체 재학습을 자동으로 시작합니다. 업데이트가 완료되더라도 추천자로부터 추천을 받을 수 있습니다. 추천자는 업데이트가 완료될 때까지 이전 구성을 사용합니다. 이 업데이트의 상태를 추적하려면 [DescribeRecommender](#) 작업에서 latestRecommenderUpdate 반환된 값을 사용하십시오.

구문 요청

```
{
  "recommenderArn": "string",
  "recommenderConfig": {
    "enableMetadataWithRecommendations": boolean,
    "itemExplorationConfig": {
      "string" : "string"
    },
    "minRecommendationRequestsPerSecond": number,
    "trainingDataConfig": {
      "excludedDatasetColumns": {
        "string" : [ "string" ]
      }
    }
  }
}
```

요청 파라미터

요청은 JSON 형식으로 다음 데이터를 받습니다.

[recommenderArn](#)

수정할 추천자의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 사항 여부: Yes

recommenderConfig

추천자의 구성 세부 정보입니다.

유형: [RecommenderConfig](#) 객체

필수 여부: 예

응답 구문

```
{
  "recommenderArn": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

recommenderArn

요청에 제공된 것과 동일한 추천자 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*+

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Amazon Personalize 이벤트

Amazon Personalize 이벤트에서 지원하는 작업은 다음과 같습니다.

- [PutActionInteractions](#)
- [PutActions](#)
- [PutEvents](#)
- [PutItems](#)
- [PutUsers](#)

PutActionInteractions

서비스: Amazon Personalize Events

작업 상호 작용 이벤트 데이터를 기록합니다. 작업 상호 작용 이벤트는 사용자와 작업 간의 상호 작용입니다. 사용자가 멤버십 프로그램에 등록하거나 앱을 다운로드하는 등의 작업을 행하는 경우를 예로 들 수 있습니다.

작업 상호 작용 기록에 대한 자세한 내용은 [작업 상호 작용 이벤트 기록](#)을 참조하세요. 작업 데이터 세트의 작업에 대한 자세한 내용은 [작업 데이터 세트](#)를 참조하세요.

Request Syntax

```
POST /action-interactions HTTP/1.1
Content-type: application/json

{
  "actionInteractions": [
    {
      "actionId": "string",
      "eventId": "string",
      "eventType": "string",
      "impression": [ "string" ],
      "properties": "string",
      "recommendationId": "string",
      "sessionId": "string",
      "timestamp": number,
      "userId": "string"
    }
  ],
  "trackingId": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

[actionInteractions](#)

세션의 작업 상호 작용 이벤트 목록.

유형: [ActionInteraction](#) 객체 어레이

어레이 멤버: 최소 항목 수 1개. 최대 항목 수 10개.

필수 여부: 예

[trackingId](#)

작업 상호 작용 이벤트 트래커의 ID. 작업 상호 작용 데이터 세트를 생성하면 Amazon Personalize 가 작업 상호 작용 이벤트 트래커를 생성합니다. 자세한 내용은 [작업 상호 작용 이벤트 트래커 ID](#)를 참조하세요.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256.

필수 여부: 예

응답 구문

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 409

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

PutActions

서비스: Amazon Personalize Events

작업 데이터 세트에 작업을 하나 이상 추가합니다. 자세한 내용은 [개별적으로 작업 가져오기](#)를 참조하세요.

Request Syntax

```
POST /actions HTTP/1.1
Content-type: application/json

{
  "actions": [
    {
      "actionId": "string",
      "properties": "string"
    }
  ],
  "datasetArn": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

actions

작업 데이터 목록.

유형: [Action](#) 객체 어레이

어레이 멤버: 최소 항목 수 1개. 최대 항목 수 10개.

필수 여부: 예

datasetArn

하나 이상의 작업을 추가할 작업 데이터 세트의 Amazon 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

필수 항목 여부: 예

응답 구문

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 409

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)

- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

PutEvents

서비스: Amazon Personalize Events

항목 상호 작용 이벤트 데이터를 기록합니다. 자세한 내용은 [항목 상호 작용 이벤트 기록](#)을 참조하세요.

Note

AWS Lambda 함수를 사용하여 PutEvents 작업을 호출하는 경우 함수의 실행 역할에 Resource 요소에 와일드카드가 * 포함된 personalize:PutEvents 작업을 수행할 수 있는 권한이 있어야 합니다.

Request Syntax

```
POST /events HTTP/1.1
Content-type: application/json

{
  "eventList": [
    {
      "eventId": "string",
      "eventType": "string",
      "eventValue": number,
      "impression": [ "string" ],
      "itemId": "string",
      "metricAttribution": {
        "eventAttributionSource": "string"
      },
      "properties": "string",
      "recommendationId": "string",
      "sentAt": number
    }
  ],
  "sessionId": "string",
  "trackingId": "string",
  "userId": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

eventList

세션의 이벤트 데이터 목록입니다.

유형: [Event](#) 객체 어레이

어레이 멤버: 최소 항목 수 1개. 최대 항목 수 10개.

필수 여부: 예

sessionId

해당 사용자의 방문과 연결된 세션 ID. 사용자가 웹 사이트를 처음 방문하거나 애플리케이션을 처음 사용할 때 애플리케이션에서 세션 ID를 생성합니다. Personalize는 사용자가 로그인하기 전에 세션 ID를 사용하여 이벤트를 해당 사용자와 연결합니다. 자세한 내용은 [항목 상호 작용 이벤트 기록](#)을 참조하세요.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256.

필수 여부: 예

trackingId

이벤트의 트랜잭션 ID. ID는 [CreateEvent트래커](#) API를 호출하여 생성됩니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256.

필수 여부: 예

userId

이벤트와 연결된 사용자.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256.

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 다시 전송합니다.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

PutItems

서비스: Amazon Personalize Events

항목 데이터세트에 하나 이상의 항목을 추가합니다. 자세한 내용은 [개별적으로 항목 가져오기](#)를 참조하세요.

Request Syntax

```
POST /items HTTP/1.1
Content-type: application/json

{
  "datasetArn": "string",
  "items": [
    {
      "itemId": "string",
      "properties": "string"
    }
  ]
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

datasetArn

항목을 추가할 항목 데이터세트의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 사항 여부: Yes

items

항목 데이터의 목록입니다.

유형: [Item](#) 객체 어레이

어레이 멤버: 최소 항목 수 1개. 최대 항목 수 10개.

필수 여부: 예

응답 구문

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 409

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)

- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

PutUsers

서비스: Amazon Personalize Events

사용자 데이터세트에 사용자를 한 명 이상 추가합니다. 자세한 내용은 [개별적으로 사용자 가져오기](#)를 참조하세요.

Request Syntax

```
POST /users HTTP/1.1
Content-type: application/json

{
  "datasetArn": "string",
  "users": [
    {
      "properties": "string",
      "userId": "string"
    }
  ]
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

datasetArn

사용자를 추가할 사용자 데이터세트의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 사항 여부: Yes

users

사용자 데이터의 목록입니다.

유형: [User](#) 객체 어레이

어레이 멤버: 최소 항목 수 1개. 최대 항목 수 10개.

필수 여부: 예

응답 구문

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceInUseException

지정된 리소스가 이미 사용 중입니다.

HTTP 상태 코드: 409

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)

- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Amazon Personalize 런타임

Amazon Personalize 런타임에서 지원하는 작업은 다음과 같습니다.

- [GetActionRecommendations](#)
- [GetPersonalizedRanking](#)
- [GetRecommendations](#)

GetActionRecommendations

서비스: Amazon Personalize Runtime

예측 점수를 기준으로 내림차순으로 정렬된 추천 작업 목록을 반환합니다.

PERSONALIZED_ACTIONS 레시피로 교육된 솔루션 버전을 배포하는 사용자 지정 캠페인이 있는 경우 GetActionRecommendations API를 사용하세요.

PERSONALIZED_ACTIONS 레시피에 대한 자세한 내용은 [PERSONALIZED_ACTIONS 레시피](#)를 참조하세요. 작업 추천을 받는 방법에 대한 자세한 내용은 [작업 추천 받기](#)를 참조하세요.

Request Syntax

```
POST /action-recommendations HTTP/1.1
Content-type: application/json

{
  "campaignArn": "string",
  "filterArn": "string",
  "filterValues": {
    "string" : "string"
  },
  "numResults": number,
  "userId": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

[campaignArn](#)

작업 추천을 받기 위해 사용할 캠페인의 Amazon 리소스 이름(ARN)입니다. 이 캠페인은 PERSONALIZED_ACTIONS 레시피로 교육된 솔루션 버전을 배포해야 합니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

Required: No

filterArn

반환되는 추천에 적용할 필터의 ARN입니다. 자세한 내용은 [추천 필터링](#) 단원을 참조하세요.

이 파라미터를 사용할 때는 필터 리소스가 ACTIVE이 맞는지 확인합니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

Required: No

filterValues

추천을 필터링할 때 사용할 값. 필터 표현식의 각 자리 표시자 파라미터에 대해 파라미터 이름(대/소문자 구분)을 키로 그리고 필터 값을 해당 값으로 각각 제공합니다. 한 파라미터의 여러 값을 쉼표로 구분합니다.

INCLUDE 요소를 사용하여 작업을 포함하는 필터 표현식의 경우 표현식에 정의되어 있는 모든 파라미터의 값을 제공해야 합니다. EXCLUDE 요소를 사용하여 작업을 제외하는 표현식이 포함된 필터의 경우 filter-values를 생략할 수 있습니다. 이 경우 Personalize는 표현식의 해당 부분을 추천을 필터링하는 데 사용하지 않습니다.

자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 단원을 참조하세요.

유형: 문자열-문자열 맵

맵 항목: 최대 항목 수 25개.

키 길이 제약: 최대 길이 50.

키 패턴: [A-Za-z0-9_]+

값 길이 제약: 최대 길이 1000.

필수 여부: 아니요

numResults

반환할 결과 수. 기본값은 5입니다. 최대값은 100입니다.

유형: 정수

유효 범위: 최소값 0.

필수 여부: 아니요

userId

작업 추천을 제공할 사용자의 사용자 ID입니다.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "actionList": [
    {
      "actionId": "string",
      "score": number
    }
  ],
  "recommendationId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

actionList

예측 점수를 기준으로 내림차순으로 정렬된 작업 추천 목록입니다. 목록에는 최대 100개의 작업이 있을 수 있습니다. 작업 점수에 대한 자세한 내용은 [작업 추천 점수를 매기는 방식](#)을 참조하세요.

유형: [PredictedAction](#) 객체 어레이

recommendationId

추천의 ID.

타입: 문자열

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정한 리소스가 존재하지 않습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

GetPersonalizedRanking

서비스: Amazon Personalize Runtime

해당 사용자에게 대한 추천 항목 목록의 순위를 다시 매깁니다. 이 목록의 첫 번째 항목은 사용자가 관심을 보일 가능성이 가장 높은 항목으로 간주됩니다.

Note

캠페인을 지원하는 솔루션은 PERSONALIZED_RANKING 유형 레시피를 사용하여 생성한 것 이어야 합니다.

Request Syntax

```
POST /personalize-ranking HTTP/1.1
Content-type: application/json
```

```
{
  "campaignArn": "string",
  "context": {
    "string" : "string"
  },
  "filterArn": "string",
  "filterValues": {
    "string" : "string"
  },
  "inputList": [ "string" ],
  "metadataColumns": {
    "string" : [ "string" ]
  },
  "userId": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

campaignArn

순위를 생성하기 위해 사용할 캠페인의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

필수 사항 여부: Yes

context

추천을 받을 때 사용할 컨텍스트 메타데이터. 컨텍스트 메타데이터에는 사용자의 현재 위치 또는 기기 유형과 같이 사용자의 추천을 받을 때 관련이 있을 수 있는 모든 상호작용 정보가 포함됩니다.

유형: 문자열-문자열 맵

맵 항목: 최대 항목 수 150개.

키 길이 제약: 최대 길이 150.

키 패턴: `[A-Za-z\d_]+`

값 길이 제약: 최대 길이 1000.

필수 여부: 아니요

filterArn

특정 사용자에게 대한 추천에 항목을 포함시키거나 항목을 제외시키기 위해 만든 필터의 Amazon 리소스 이름(ARN)입니다. 자세한 내용은 [추천 필터링](#) 단원을 참조하세요.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

filterValues

추천을 필터링할 때 사용할 값. 필터 표현식의 각 자리 표시자 파라미터에 대해 파라미터 이름(대/소문자 구분)을 키로 그리고 필터 값을 해당 값으로 각각 제공합니다. 한 파라미터의 여러 값을 쉼표로 구분합니다.

INCLUDE 요소를 사용하여 항목을 포함시키는 필터 표현식의 경우 표현식에 정의되어 있는 모든 파라미터의 값을 제공해야 합니다. EXCLUDE 요소를 사용하여 항목을 제외시키는 표현식이 있는 필터의 경우 filter-values를 생략할 수 있습니다. 이 경우 Personalize는 표현식의 해당 부분을 사용하여 추천을 필터링하지 않습니다.

자세한 내용은 [추천 필터링](#) 단원을 참조하세요.

유형: 문자열-문자열 맵

맵 항목: 최대 항목 수 25개.

키 길이 제약: 최대 길이 50.

키 패턴: [A-Za-z0-9_]+

값 길이 제약: 최대 길이 1000.

필수 여부: 아니요

[inputList](#)

순위를 매길 항목이 목록(itemId 기준). 항목이 학습 데이터세트에 포함되지 않은 경우 해당 항목은 순위가 재조정된 목록 끝에 추가됩니다. 추천에 메타데이터를 포함하는 경우 최대값은 50입니다. 포함하지 않는 경우 최대값은 500입니다.

유형: 문자열 어레이

길이 제약: 최대 길이 256.

필수 여부: 예

[metadataColumns](#)

캠페인을 만들거나 업데이트할 때 추천에서 메타데이터를 활성화한 경우 개인 맞춤 순위에 포함할 항목 데이터 세트의 메타데이터 열을 지정하세요. 맵 키는 ITEMS이고, 값은 항목 데이터 세트의 열 이름 목록입니다. 제공할 수 있는 최대 열 수는 10개입니다.

캠페인용 메타데이터 활성화에 대한 자세한 내용은 [캠페인용 추천의 메타데이터 활성화](#)를 참조하세요.

유형: 문자열-문자열 어레이 맵

맵 항목: 최대 항목 수 1개입니다.

키 길이 제약: 최대 길이 256.

배열 멤버: 최대 항목 수는 99개입니다.

길이 제약: 최대 길이 150.

필수 여부: 아니요

userId

캠페인에서 순위를 제공하려는 대상 사용자.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 예

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "personalizedRanking": [
    {
      "itemId": "string",
      "metadata": {
        "string" : "string"
      },
      "promotionName": "string",
      "reason": [ "string" ],
      "score": number
    }
  ],
  "recommendationId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[personalizedRanking](#)

사용자가 가장 관심을 가질 만한 순서의 항목 목록입니다. 최대값은 500입니다.

유형: [PredictedItem](#) 객체 어레이

[recommendationId](#)

추천의 ID.

타입: 문자열

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정한 리소스가 존재하지 않습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

GetRecommendations

서비스: Amazon Personalize Runtime

추천 항목의 목록을 반환합니다. 캠페인의 경우 캠페인의 Amazon 리소스 이름(ARN)이 필요하며 필요한 사용자 및 항목 입력은 다음과 같이 캠페인을 지원하는 솔루션을 생성하는 데 사용된 레시피 유형에 따라 달라집니다.

- USER_PERSONALIZATION - `userId`필요, `itemId`는 사용되지 않음
- RELATED_ITEMS - `itemId`필요, `userId`는 사용되지 않음

Note

PERSONALIZED_RANKING 유형의 레시피를 사용하여 만든 솔루션으로 지원되는 캠페인은 [GetPersonalizedRanking](#) API를 사용합니다.

추천자의 경우 추천자의 ARN이 필요하며 필요한 항목과 사용자 입력은 해당 추천자를 지원하는 사용 사례(도메인 기반 레시피)에 따라 달라집니다. 사용 사례 요구 사항에 대한 자세한 내용은 [추천자 사용 사례 선택](#) 단원을 참조하세요.

Request Syntax

```
POST /recommendations HTTP/1.1
Content-type: application/json
```

```
{
  "campaignArn": "string",
  "context": {
    "string" : "string"
  },
  "filterArn": "string",
  "filterValues": {
    "string" : "string"
  },
  "itemId": "string",
  "metadataColumns": {
    "string" : [ "string" ]
  },
  "numResults": number,
  "promotions": [
```

```

    {
      "filterArn": "string",
      "filterValues": {
        "string" : "string"
      },
      "name": "string",
      "percentPromotedItems": number
    }
  ],
  "recommenderArn": "string",
  "userId": "string"
}

```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

campaignArn

추천을 받기 위해 사용할 캠페인의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

context

추천을 받을 때 사용할 컨텍스트 메타데이터. 컨텍스트 메타데이터에는 사용자의 현재 위치 또는 기기 유형과 같이 사용자의 추천을 받을 때 관련이 있을 수 있는 모든 상호작용 정보가 포함됩니다.

유형: 문자열-문자열 맵

맵 항목: 최대 항목 수 150개.

키 길이 제약: 최대 길이 150.

키 패턴: `[A-Za-z\d_]+`

값 길이 제약: 최대 길이 1000.

필수 여부: 아니요

filterArn

반환되는 추천에 적용할 필터의 ARN입니다. 자세한 내용은 [추천 필터링](#) 단원을 참조하세요.

이 파라미터를 사용할 때는 필터 리소스가 ACTIVE이 맞는지 확인합니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

filterValues

추천을 필터링할 때 사용할 값. 필터 표현식의 각 자리 표시자 파라미터에 대해 파라미터 이름(대/소문자 구분)을 키로 그리고 필터 값을 해당 값으로 각각 제공합니다. 한 파라미터의 여러 값을 쉼표로 구분합니다.

INCLUDE 요소를 사용하여 항목을 포함시키는 필터 표현식의 경우 표현식에 정의되어 있는 모든 파라미터의 값을 제공해야 합니다. EXCLUDE 요소를 사용하여 항목을 제외시키는 표현식이 있는 필터의 경우 `filter-values`를 생략할 수 있습니다. 이 경우 Personalize는 표현식의 해당 부분을 사용하여 추천을 필터링하지 않습니다.

자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 단원을 참조하세요.

유형: 문자열-문자열 맵

맵 항목: 최대 항목 수 25개.

키 길이 제약: 최대 길이 50.

키 패턴: `[A-Za-z0-9_]+`

값 길이 제약: 최대 길이 1000.

필수 여부: 아니요

itemId

추천을 제공할 항목 ID.

RELATED_ITEMS 레시피 유형의 경우 필수.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

metadataColumns

캠페인 또는 추천자를 만들거나 업데이트할 때 추천에서 메타데이터를 활성화한 경우 항목 데이터 세트의 메타데이터 열을 지정하여 항목 추천에 포함하세요. 맵 키는 ITEMS이고, 값은 항목 데이터 세트의 열 이름 목록입니다. 제공할 수 있는 최대 열 수는 10개입니다.

캠페인용 메타데이터 활성화에 대한 자세한 내용은 [캠페인용 추천의 메타데이터 활성화](#)를 참조하세요. 추천자용 메타데이터 활성화에 대한 자세한 내용은 [추천자용 추천의 메타데이터 활성화](#)를 참조하세요.

유형: 문자열-문자열 어레이 맵

맵 항목: 최대 항목 수 1개입니다.

키 길이 제약: 최대 길이 256.

배열 멤버: 최대 항목 수는 99개입니다.

길이 제약: 최대 길이 150.

필수 여부: 아니요

numResults

반환할 결과 수. 기본값은 25입니다. 추천에 메타데이터를 포함하는 경우 최대값은 50입니다. 포함하지 않는 경우 최대값은 500입니다.

유형: 정수

유효 범위: 최소값 0.

필수 여부: 아니요

promotions

추천 요청에 적용할 프로모션. 홍보는 추천 항목의 구성 가능한 하위 집합에 적용되는 추가 비즈니스 규칙을 정의합니다.

유형: [Promotion](#) 객체 어레이

어레이 멤버: 최대 항목 수 1개.

필수 여부: 아니요

[recommenderArn](#)

추천을 받는 데 사용할 추천자의 Amazon 리소스 이름(ARN)입니다. 도메인 사용 사례에 대한 추천자를 사용하여 도메인 데이터세트 그룹을 생성한 경우 추천자 ARN을 제공합니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

[userId](#)

추천을 제공할 사용자 ID.

USER_PERSONALIZATION 레시피 유형의 경우 필수.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "itemList": [
    {
      "itemId": "string",
      "metadata": {
        "string" : "string"
      },
      "promotionName": "string",
```

```
    "reason": [ "string" ],
    "score": number
  }
],
"recommendationId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[itemList](#)

예측 점수를 기준으로 내림차순으로 정렬된 추천 목록입니다. 목록에는 최대 500개의 항목이 있을 수 있습니다.

유형: [PredictedItem](#) 객체 어레이

[recommendationId](#)

추천의 ID.

타입: 문자열

Errors

InvalidInputException

필드 또는 파라미터에 유효한 값을 제공합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정한 리소스가 존재하지 않습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

데이터 타입

Personalize에서 지원하는 데이터 유형은 다음과 같습니다.

- [Algorithm](#)
- [AlgorithmImage](#)
- [AutoMLConfig](#)
- [AutoMLResult](#)
- [AutoTrainingConfig](#)
- [BatchInferenceJob](#)
- [BatchInferenceJobConfig](#)
- [BatchInferenceJobInput](#)
- [BatchInferenceJobOutput](#)
- [BatchInferenceJobSummary](#)
- [BatchSegmentJob](#)
- [BatchSegmentJobInput](#)
- [BatchSegmentJobOutput](#)
- [BatchSegmentJobSummary](#)
- [Campaign](#)
- [CampaignConfig](#)
- [CampaignSummary](#)

- [CampaignUpdateSummary](#)
- [CategoricalHyperParameterRange](#)
- [ContinuousHyperParameterRange](#)
- [DataDeletionJob](#)
- [DataDeletionJobSummary](#)
- [Dataset](#)
- [DatasetExportJob](#)
- [DatasetExportJobOutput](#)
- [DatasetExportJobSummary](#)
- [DatasetGroup](#)
- [DatasetGroupSummary](#)
- [DatasetImportJob](#)
- [DatasetImportJobSummary](#)
- [DatasetSchema](#)
- [DatasetSchemaSummary](#)
- [DatasetSummary](#)
- [DatasetUpdateSummary](#)
- [DataSource](#)
- [DefaultCategoricalHyperParameterRange](#)
- [DefaultContinuousHyperParameterRange](#)
- [DefaultHyperParameterRanges](#)
- [DefaultIntegerHyperParameterRange](#)
- [EventTracker](#)
- [EventTrackerSummary](#)
- [FeatureTransformation](#)
- [FieldsForThemeGeneration](#)
- [Filter](#)
- [FilterSummary](#)
- [HPOConfig](#)
- [HPOObjective](#)

- [HPOResourceConfig](#)
- [HyperParameterRanges](#)
- [IntegerHyperParameterRange](#)
- [MetricAttribute](#)
- [MetricAttribution](#)
- [MetricAttributionOutput](#)
- [MetricAttributionSummary](#)
- [OptimizationObjective](#)
- [Recipe](#)
- [RecipeSummary](#)
- [Recommender](#)
- [RecommenderConfig](#)
- [RecommenderSummary](#)
- [RecommenderUpdateSummary](#)
- [S3DataConfig](#)
- [Solution](#)
- [SolutionConfig](#)
- [SolutionSummary](#)
- [SolutionVersion](#)
- [SolutionVersionSummary](#)
- [Tag](#)
- [ThemeGenerationConfig](#)
- [TrainingDataConfig](#)
- [TunedHPOParams](#)

Personalize 이벤트에서 지원하는 데이터 유형은 다음과 같습니다.

- [Action](#)
- [ActionInteraction](#)
- [Event](#)
- [Item](#)

- [MetricAttribution](#)
- [User](#)

Personalize 런타임에서 지원하는 데이터 유형은 다음과 같습니다.

- [PredictedAction](#)
- [PredictedItem](#)
- [Promotion](#)

Personalize

Personalize에서 지원하는 데이터 유형은 다음과 같습니다.

- [Algorithm](#)
- [AlgorithmImage](#)
- [AutoMLConfig](#)
- [AutoMLResult](#)
- [AutoTrainingConfig](#)
- [BatchInferenceJob](#)
- [BatchInferenceJobConfig](#)
- [BatchInferenceJobInput](#)
- [BatchInferenceJobOutput](#)
- [BatchInferenceJobSummary](#)
- [BatchSegmentJob](#)
- [BatchSegmentJobInput](#)
- [BatchSegmentJobOutput](#)
- [BatchSegmentJobSummary](#)
- [Campaign](#)
- [CampaignConfig](#)
- [CampaignSummary](#)
- [CampaignUpdateSummary](#)
- [CategoricalHyperParameterRange](#)

- [ContinuousHyperParameterRange](#)
- [DataDeletionJob](#)
- [DataDeletionJobSummary](#)
- [Dataset](#)
- [DatasetExportJob](#)
- [DatasetExportJobOutput](#)
- [DatasetExportJobSummary](#)
- [DatasetGroup](#)
- [DatasetGroupSummary](#)
- [DatasetImportJob](#)
- [DatasetImportJobSummary](#)
- [DatasetSchema](#)
- [DatasetSchemaSummary](#)
- [DatasetSummary](#)
- [DatasetUpdateSummary](#)
- [DataSource](#)
- [DefaultCategoricalHyperParameterRange](#)
- [DefaultContinuousHyperParameterRange](#)
- [DefaultHyperParameterRanges](#)
- [DefaultIntegerHyperParameterRange](#)
- [EventTracker](#)
- [EventTrackerSummary](#)
- [FeatureTransformation](#)
- [FieldsForThemeGeneration](#)
- [Filter](#)
- [FilterSummary](#)
- [HPOConfig](#)
- [HPOObjective](#)
- [HPOResourceConfig](#)
- [HyperParameterRanges](#)

- [IntegerHyperParameterRange](#)
- [MetricAttribute](#)
- [MetricAttribution](#)
- [MetricAttributionOutput](#)
- [MetricAttributionSummary](#)
- [OptimizationObjective](#)
- [Recipe](#)
- [RecipeSummary](#)
- [Recommender](#)
- [RecommenderConfig](#)
- [RecommenderSummary](#)
- [RecommenderUpdateSummary](#)
- [S3DataConfig](#)
- [Solution](#)
- [SolutionConfig](#)
- [SolutionSummary](#)
- [SolutionVersion](#)
- [SolutionVersionSummary](#)
- [Tag](#)
- [ThemeGenerationConfig](#)
- [TrainingDataConfig](#)
- [TunedHPOParams](#)

Algorithm

서비스: Amazon Personalize

사용자 지정 알고리즘을 설명합니다.

내용

algorithmArn

알고리즘의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

algorithmImage

알고리즘 이미지에 대한 Docker 컨테이너의 URI입니다.

유형: [AlgorithmImage](#) 객체

필수 여부: 아니요

creationDateTime

알고리즘이 생성된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

defaultHyperParameterRanges

기본 하이퍼파라미터, 범위, 조정 가능 여부를 지정합니다. 조정 가능한 하이퍼파라미터는 하이퍼파라미터 최적화(HPO) 중 값을 결정할 수 있습니다.

유형: [DefaultHyperParameterRanges](#) 객체

필수 항목 여부: 아니요

defaultHyperParameters

기본 하이퍼파라미터를 지정합니다.

유형: 문자열-문자열 맵

맵 항목: 최대 항목 수 100개.

키 길이 제약: 최대 길이 256.

값 길이 제약: 최대 길이 1000.

필수 여부: 아니요

defaultResourceConfig

기본 최대 학습 작업 및 병렬 학습 작업 수를 지정합니다.

유형: 문자열-문자열 맵

맵 항목: 최대 항목 수 100개.

키 길이 제약: 최대 길이 256.

값 길이 제약: 최대 길이 1000.

필수 여부: 아니요

lastUpdatedDateTime

알고리즘이 마지막으로 업데이트된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

name

알고리즘의 이름입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

roleArn

역할의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

trainingInputMode

학습 입력 모드.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

AlgorithmImage

서비스: Amazon Personalize

알고리즘 이미지를 설명합니다.

내용

dockerURI

알고리즘 이미지에 대한 Docker 컨테이너의 URI입니다.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 예

name

알고리즘 이미지의 이름입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

AutoMLConfig

서비스: Amazon Personalize

솔루션이 AutoML (performAutoMLis true in [CreateSolution](#)) 을 수행하는 경우 Amazon Personalize 는 지정된 목록에서 주어진 지표를 최적화하는 레시피를 결정합니다. 그런 다음 Personalize는 해당 레시피를 솔루션에 사용합니다.

내용

metricName

최적화할 지표.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

recipeList

후보 레시피의 목록입니다.

유형: 문자열 어레이

어레이 멤버: 최대 항목 수 100개.

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

AutoMLResult

서비스: Amazon Personalize

솔루션에서 AutoML (performAutoMLis true in [CreateSolution](#)) 을 수행하는 경우 지정된 지표를 가장 잘 최적화한 레시피를 지정합니다.

내용

bestRecipeArn

최적 레시피의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]):personalize:.*:.*:.*

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

AutoTrainingConfig

서비스: Amazon Personalize

언제 사용할 자동 교육 구성이 `performAutoTraining true`입니다.

내용

schedulingExpression

새 솔루션 버전을 자동으로 학습시키는 빈도를 지정합니다. 비율 표현식을 비율 (값 단위) 형식으로 지정합니다. 값에는 1에서 30 사이의 숫자를 지정하십시오. 단위로는 day 또는 을 지정합니다 days. 예를 들어 5일마다 새 솔루션 버전을 자동으로 생성하려면 다음을 지정하십시오 `rate(5 days)`. 기본값은 7일마다입니다.

자동 교육에 대한 자세한 내용은 [솔루션 만들기 및 구성](#)을 참조하십시오.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 16입니다.

패턴: `rate\(\d+ days?\)`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

BatchInferenceJob

서비스: Amazon Personalize

배치 추론 작업에 대한 정보가 포함되어 있습니다.

내용

batchInferenceJobArn

배치 추론 작업의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

batchInferenceJobConfig

배치 추론 작업의 구성 세부 정보의 문자열-문자열 맵.

유형: [BatchInferenceJobConfig](#) 객체

필수 항목 여부: 아니요

batchInferenceJobMode

작업 모드.

타입: 문자열

유효 값: BATCH_INFERENCE | THEME_GENERATION

필수 여부: 아니요

creationDateTime

배치 추론 작업이 생성된 시간.

유형: 타임스탬프

필수 여부: 아니요

failureReason

배치 추론 작업이 실패한 경우 실패의 원인입니다.

타입: 문자열

필수사항: 아니요

filterArn

배치 추론 작업에 사용된 필터의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

jobInput

배치 추론 작업을 생성하는 데 사용되는 입력 데이터로 이어지는 S3 경로입니다.

유형: [BatchInferenceJobInput](#) 객체

필수 항목 여부: 아니요

jobName

배치 추론 작업의 이름입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

jobOutput

배치 추론 작업을 통해 생성된 출력 데이터가 들어 있는 S3 버킷.

유형: [BatchInferenceJobOutput](#) 객체

필수 항목 여부: 아니요

lastUpdatedDateTime

배치 추론 작업이 마지막으로 업데이트된 시간.

유형: 타임스탬프

필수 여부: 아니요

numResults

배치 추론 작업에서 생성된 추천 수. 이 숫자에는 실패한 입력 레코드에 대해 생성된 오류 메시지가 포함됩니다.

유형: 정수

필수 항목 여부: 아니요

roleArn

배치 추론 작업을 요청한 Amazon Identity and Access Management(IAM) 역할의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.\@-_/]+`

Required: No

solutionVersionArn

배치 추론 작업이 생성된 솔루션 버전의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

status

배치 추론 작업의 상태입니다. 상태는 다음 값 중 하나일 수 있습니다.

- PENDING

- IN PROGRESS
- ACTIVE
- CREATE FAILED

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

themeGenerationConfig

작업의 테마 생성 설정.

유형: [ThemeGenerationConfig](#) 객체

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

BatchInferenceJobConfig

서비스: Amazon Personalize

배치 추론 작업의 구성 세부 정보입니다.

내용

itemExplorationConfig

Personalize가 항목을 추천할 때 사용하는 항목 탐색량을 구성하는 데 사용할 탐색 구성 하이퍼파라미터(explorationWeight 및 explorationItemAgeCutOff포함)를 지정하는 문자열-문자열 맵. [사용자-개인 맞춤](#)을 참조하세요.

유형: 문자열-문자열 맵

맵 항목: 최대 항목 수 100개.

키 길이 제약: 최대 길이 256.

값 길이 제약: 최대 길이 1000.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

BatchInferenceJobInput

서비스: Amazon Personalize

배치 추론 작업의 입력 구성.

내용

s3DataSource

입력 데이터가 포함된 S3 위치의 URI입니다. S3 버킷이 호출하는 API 엔드포인트와 동일한 리전에 있어야 합니다.

유형: [S3DataConfig](#) 객체

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

BatchInferenceJobOutput

서비스: Amazon Personalize

배치 추론 작업의 출력 구성 파라미터.

내용

s3DataDestination

배치 추론 작업의 출력이 저장된 S3 버킷에 대한 정보입니다.

유형: [S3DataConfig](#) 객체

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

BatchInferenceJobSummary

서비스: Amazon Personalize

의 잘린 버전입니다. [BatchInferenceJob](#) 이 [ListBatchInferenceJobs](#)작업은 일괄 추론 작업 요약 목록을 반환합니다.

내용

batchInferenceJobArn

배치 추론 작업의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]):personalize:.*:.*:.*+

Required: No

batchInferenceJobMode

작업 모드.

타입: 문자열

유효 값: BATCH_INFERENCE | THEME_GENERATION

필수 여부: 아니요

creationDateTime

배치 추론 작업이 생성된 시간.

유형: 타임스탬프

필수 여부: 아니요

failureReason

배치 추론 작업이 실패한 경우 실패의 원인입니다.

타입: 문자열

필수사항: 아니요

jobName

배치 추론 작업의 이름입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

lastUpdatedDateTime

배치 추론 작업이 마지막으로 업데이트된 시간.

유형: 타임스탬프

필수 여부: 아니요

solutionVersionArn

배치 추론 작업에 사용된 솔루션 버전의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

status

배치 추론 작업의 상태입니다. 상태는 다음 값 중 하나일 수 있습니다.

- PENDING
- IN PROGRESS
- ACTIVE
- CREATE FAILED

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

BatchSegmentJob

서비스: Amazon Personalize

배치 세그먼트 작업에 대한 정보가 들어 있습니다.

내용

batchSegmentJobArn

배치 세그먼트 작업의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

creationDateTime

배치 세그먼트 작업이 생성된 시간.

유형: 타임스탬프

필수 여부: 아니요

failureReason

배치 세그먼트 작업이 실패한 경우 실패의 원인입니다.

타입: 문자열

필수사항: 아니요

filterArn

배치 세그먼트 작업에 사용된 필터의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

jobInput

배치 세그먼트 작업을 생성하는 데 사용되는 입력 데이터로 이어지는 S3 경로입니다.

유형: [BatchSegmentJobInput](#) 객체

필수 여부: 아니요

jobName

배치 세그먼트 작업의 이름입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

jobOutput

배치 세그먼트 작업을 통해 생성된 출력 데이터가 들어 있는 S3 버킷.

유형: [BatchSegmentJobOutput](#) 객체

필수 항목 여부: 아니요

lastUpdatedDateTime

배치 세그먼트 작업이 마지막으로 업데이트된 시간.

유형: 타임스탬프

필수 여부: 아니요

numResults

입력 데이터의 각 라인에 대해 배치 세그먼트 작업으로 생성된 예상 사용자 수입니다. 세그먼트당 최대 사용자 수는 5백만 명입니다.

유형: 정수

필수 항목 여부: 아니요

roleArn

배치 세그먼트 작업을 요청한 Amazon Identity and Access Management(IAM) 역할의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.\@-_/]+`

Required: No

solutionVersionArn

배치 세그먼트 작업이 배치 세그먼트를 생성할 때 사용한 솔루션 버전의 Amazon 리소스 이름 (ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

status

배치 세그먼트 작업의 상태입니다. 상태는 다음 값 중 하나일 수 있습니다.

- PENDING
- IN PROGRESS
- ACTIVE
- CREATE FAILED

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

BatchSegmentJobInput

서비스: Amazon Personalize

배치 세그먼트 작업의 입력 구성.

내용

s3DataSource

S3 입력 또는 출력 버킷의 구성 세부 정보입니다.

유형: [S3DataConfig](#) 객체

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

BatchSegmentJobOutput

서비스: Amazon Personalize

배치 세그먼트 작업의 출력 구성 파라미터.

내용

s3DataDestination

S3 입력 또는 출력 버킷의 구성 세부 정보입니다.

유형: [S3DataConfig](#) 객체

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

BatchSegmentJobSummary

서비스: Amazon Personalize

데이터 유형의 잘린 버전입니다. [BatchSegmentJob](#) [ListBatchSegmentJobs](#) 작업은 배치 세그먼트 작업 요약 목록을 반환합니다.

내용

batchSegmentJobArn

배치 세그먼트 작업의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

creationDateTime

배치 세그먼트 작업이 생성된 시간.

유형: 타임스탬프

필수 여부: 아니요

failureReason

배치 세그먼트 작업이 실패한 경우 실패의 원인입니다.

타입: 문자열

필수사항: 아니요

jobName

배치 세그먼트 작업의 이름입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

lastUpdatedDateTime

배치 세그먼트 작업이 마지막으로 업데이트된 시간.

유형: 타임스탬프

필수 여부: 아니요

solutionVersionArn

배치 세그먼트 작업이 배치 세그먼트를 생성할 때 사용한 솔루션 버전의 Amazon 리소스 이름 (ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

status

배치 세그먼트 작업의 상태입니다. 상태는 다음 값 중 하나일 수 있습니다.

- PENDING
- IN PROGRESS
- ACTIVE
- CREATE FAILED

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)

- [AWS 루비 V3용 SDK](#)

Campaign

서비스: Amazon Personalize

솔루션 버전의 배포를 설명하는 객체입니다. 캠페인에 대한 자세한 내용은 [을 참조하십시오](#)
[CreateCampaign](#).

내용

campaignArn

캠페인의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

Required: No

campaignConfig

캠페인의 구성 세부 정보입니다.

유형: [CampaignConfig](#) 객체

필수 항목 여부: 아니요

creationDateTime

캠페인을 만든 날짜 및 시간(Unix 형식)

유형: 타임스탬프

필수 여부: 아니요

failureReason

캠페인이 실패할 경우, 실패의 원인.

타입: 문자열

필수사항: 아니요

lastUpdatedDateTime

캠페인이 마지막으로 업데이트된 날짜 및 시간(Unix 형식).

유형: 타임스탬프

필수 여부: 아니요

latestCampaignUpdate

캠페인 업데이트 속성에 대한 요약을 제공합니다. 전체 목록을 보려면 [DescribeCampaignAPI](#)를 호출하세요.

유형: [CampaignUpdateSummary](#)객체

필수 여부: 아니요

minProvisionedTPS

초당 요청된 최소 프로비저닝 트랜잭션(추천)을 지정합니다. minProvisionedTPS가 높으면 요금이 인상됩니다. minProvisionedTPS(기본값)의 경우 1부터 시작하는 것이 좋습니다. Amazon CloudWatch 지표를 사용하여 사용량을 추적하고 minProvisionedTPS 필요에 따라 사용량을 늘리십시오.

타입: 정수

유효 범위: 최소값 1.

필수 여부: 아니요

name

캠페인의 이름입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

solutionVersionArn

캠페인에서 사용하는 솔루션 버전의 Amazon 리소스 이름 (ARN)

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*+

Required: No

status

캠페인의 상태입니다.

캠페인은 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

CampaignConfig

서비스: Amazon Personalize

캠페인의 구성 세부 정보입니다.

내용

enableMetadataWithRecommendations

추천이 포함된 메타데이터가 캠페인에 활성화되어 있는지 여부입니다. 활성화한 경우 추천 요청에 항목 데이터 세트의 열을 지정할 수 있습니다. Amazon Personalize는 추천 응답의 각 항목에 대해 이 데이터를 반환합니다. 캠페인용 메타데이터 활성화에 대한 자세한 내용은 [캠페인용 추천의 메타데이터 활성화](#)를 참조하세요.

추천에서 메타데이터를 활성화하면 추가 비용이 발생합니다. 자세한 내용은 [Amazon Personalize 요금](#)을 참조하세요.

타입: 부울

필수 항목 여부: 아니요

itemExplorationConfig

Personalize가 항목을 추천할 때 사용하는 항목 탐색량을 구성하는 데 사용할 탐색 구성 하이퍼파라미터(explorationWeight 및 explorationItemAgeCutOff포함)를 지정합니다. 솔루션에서 [사용자-개인 맞춤](#) 레시피를 사용하는 경우에만 itemExplorationConfig데이터를 제공합니다.

유형: 문자열-문자열 맵

맵 항목: 최대 항목 수 100개.

키 길이 제약: 최대 길이 256.

값 길이 제약: 최대 길이 1000.

필수 여부: 아니요

syncWithLatestSolutionVersion

캠페인이 솔루션의 최신 솔루션 버전 (학습된 모델) 을 사용하도록 자동 업데이트되는지 여부. 지정하는 True 경우 파라미터에 대한 솔루션의 ARN을 지정해야 합니다. SolutionVersionArn

SolutionArn/\$LATEST형식이어야 합니다. 기본값은 False 이며 최신 솔루션 버전을 배포하려면 캠페인을 수동으로 업데이트해야 합니다.

자동 캠페인 업데이트에 대한 자세한 내용은 [자동 캠페인 업데이트 활성화](#)를 참조하십시오.

타입: 부울

필수 항목 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

CampaignSummary

서비스: Amazon Personalize

캠페인의 속성에 대한 요약을 제공합니다. 전체 목록을 보려면 [DescribeCampaignAPI](#)를 호출하세요.

내용

campaignArn

캠페인의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*+

Required: No

creationDateTime

캠페인이 생성된 날짜 및 시간(Unix 시간)

유형: 타임스탬프

필수 여부: 아니요

failureReason

캠페인이 실패할 경우, 실패의 원인.

타입: 문자열

필수사항: 아니요

lastUpdatedDateTime

캠페인이 마지막으로 업데이트된 날짜 및 시간(Unix 시간)입니다.

유형: 타임스탬프

필수 여부: 아니요

name

캠페인의 이름입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

캠페인의 상태입니다.

캠페인은 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

CampaignUpdateSummary

서비스: Amazon Personalize

캠페인 업데이트 속성에 대한 요약を提供합니다. 전체 목록을 보려면 [DescribeCampaignAPI](#)를 호출하세요.

내용

campaignConfig

캠페인의 구성 세부 정보입니다.

유형: [CampaignConfig](#) 객체

필수 여부: 아니요

creationDateTime

캠페인 업데이트가 생성된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

failureReason

캠페인 업데이트가 실패할 경우, 실패의 원인.

타입: 문자열

필수 사항: 아니요

lastUpdatedDateTime

캠페인이 마지막으로 업데이트된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

minProvisionedTPS

Personalize에서 지원할 초당 요청된 최소 프로비저닝 트랜잭션(추천)을 지정합니다.

타입: 정수

유효 범위: 최소값 1.

필수 여부: 아니요

solutionVersionArn

배포된 솔루션 버전의 Amazon 리소스 이름(ARN)

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*+

Required: No

status

캠페인 업데이트의 상태입니다.

캠페인 업데이트는 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

CategoricalHyperParameterRange

서비스: Amazon Personalize

범주형 하이퍼파라미터의 이름과 범위를 제공합니다.

내용

name

하이퍼파라미터의 이름입니다.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

values

하이퍼파라미터의 범주 목록입니다.

유형: 문자열 어레이

어레이 멤버: 최대 항목 수 100개.

길이 제약: 최대 길이 1000.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ContinuousHyperParameterRange

서비스: Amazon Personalize

지속적 하이퍼파라미터의 이름과 범위를 제공합니다.

내용

maxValue

하이퍼파라미터에 허용되는 최대값.

유형: Double

유효 범위: 최소값 -1000000.

필수 여부: 아니요

minValue

하이퍼파라미터에 허용되는 최소값.

유형: Double

유효 범위: 최소값 -1000000.

필수 여부: 아니요

name

하이퍼파라미터의 이름입니다.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)

- [AWS 루비 V3용 SDK](#)

DataDeletionJob

서비스: Amazon Personalize

Amazon Personalize 데이터 세트 그룹에서 특정 사용자에게 대한 모든 참조를 일괄적으로 삭제하는 작업을 설명합니다. [데이터 삭제 작업 생성에 대한 자세한 내용은 사용자 삭제를 참조하십시오.](#)

내용

creationDateTime

데이터 삭제 작업의 생성 날짜 및 시간 (Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

dataDeletionJobArn

데이터 삭제 작업의 Amazon 리소스 이름 (ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

datasetGroupArn

작업이 레코드를 삭제하는 데이터세트 그룹의 Amazon 리소스 이름 (ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

dataSource

데이터 세트에 업로드할 데이터가 포함된 데이터 소스 또는 Amazon Personalize에서 삭제할 레코드 목록을 설명합니다.

유형: [DataSource](#)객체

필수 항목 여부: 아니요

failureReason

데이터 삭제 작업이 실패하는 경우 이유를 제공하십시오.

타입: 문자열

필수사항: 아니요

jobName

데이터 삭제 작업의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

lastUpdatedDateTime

데이터 삭제 작업이 마지막으로 업데이트된 날짜 및 시간 (Unix 시간)

유형: 타임스탬프

필수 여부: 아니요

numDeleted

완료된 작업으로 삭제된 레코드 수입니다.

유형: 정수

필수 항목 여부: 아니요

roleArn

Amazon S3 데이터 소스에서 읽을 권한이 있는 IAM 역할의 Amazon 리소스 이름 (ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.\@-_/\]+`

Required: No

status

데이터 삭제 작업의 상태.

데이터 삭제 작업의 상태는 다음 중 하나일 수 있습니다.

- [보류 중] > [진행 중] > [완료] 또는 [실패]

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DataDeletionJobSummary

서비스: Amazon Personalize

데이터 삭제 작업의 속성에 대한 요약을 제공합니다. 전체 목록을 보려면 [DescribeDataDeletionJobAPI](#) 작업을 호출하십시오.

내용

creationDateTime

데이터 삭제 작업의 생성 날짜 및 시간 (Unix 시간)

유형: 타임스탬프

필수 여부: 아니요

dataDeletionJobArn

데이터 삭제 작업의 Amazon 리소스 이름 (ARN)

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

datasetGroupArn

작업이 레코드를 삭제한 데이터세트 그룹의 Amazon 리소스 이름 (ARN)

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

failureReason

데이터 삭제 작업이 실패하는 경우 이유를 제공하십시오.

타입: 문자열

필수사항: 아니요

jobName

데이터 삭제 작업의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

lastUpdatedDateTime

데이터 삭제 작업이 마지막으로 업데이트된 날짜 및 시간 (Unix 시간)

유형: 타임스탬프

필수 여부: 아니요

status

데이터 삭제 작업의 상태.

데이터 삭제 작업의 상태는 다음 중 하나일 수 있습니다.

- [보류 중] > [진행 중] > [완료] 또는 [실패]

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Dataset

서비스: Amazon Personalize

데이터세트에 대한 메타데이터를 제공합니다.

내용

creationDateTime

데이터세트 생성 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

datasetArn

메타데이터를 원하는 데이터세트의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

datasetGroupArn

데이터세트 그룹의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

datasetType

다음 값 중 하나입니다.

- 상호작용
- Items

- 사용자
- 작업
- Action_Interactions

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

lastUpdatedDateTime

데이터세트가 업데이트된 시기를 보여주는 타임스탬프.

유형: 타임스탬프

필수 여부: 아니요

latestDatasetUpdate

데이터세트의 최신 업데이트를 설명합니다.

유형: [DatasetUpdateSummary](#) 객체

필수 여부: 아니요

name

데이터세트의 이름입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

schemaArn

연결된 스키마의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

status

데이터세트의 상태입니다.

데이터세트는 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

trackingId

작업 상호 작용 데이터 세트의 이벤트 트래커 ID입니다. PutActionInteractions API 작업에서 트래커의 ID를 지정합니다. Amazon Personalize는 이를 사용하여 새 데이터를 데이터 세트 그룹의 작업 상호 작용 데이터 세트로 전달합니다.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DatasetExportJob

서비스: Amazon Personalize

데이터세트를 S3 버킷으로 내보내는 작업을 설명합니다. 자세한 내용은 [이 링크](#)를 참조하십시오.

데이터세트 내보내기 작업은 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED

내용

creationDateTime

데이터세트 내보내기 작업의 생성 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

datasetArn

내보낼 데이터세트의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

Required: No

datasetExportJobArn

데이터세트 내보내기 작업의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

Required: No

failureReason

데이터세트 내보내기 작업이 실패한 경우 그 이유를 제공합니다.

타입: 문자열

필수 여부: 아니요

ingestionMode

데이터를 가져온 방식에 따라 내보낼 데이터. 데이터세트 가져오기 작업을 사용하여 가져온 BULK 데이터, 콘솔 PutEvents, PutUsers 및 PutItems 작업을 사용하여 점진적으로 가져온 PUT 데이터를 내보내거나 두 유형 ALL 모두에 대해 내보낼 수 있습니다. 기본 값은 PUT입니다.

타입: 문자열

유효 값: BULK | PUT | ALL

필수 여부: 아니요

jobName

내보내기 작업의 이름입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

jobOutput

작업의 출력이 저장되는 S3 버킷의 경로입니다. 예:

`s3://bucket-name/folder-name/`

유형: [DatasetExportJobOutput](#) 객체

필수 여부: 아니요

lastUpdatedDateTime

데이터세트 내보내기 작업의 상태가 마지막으로 업데이트된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

roleArn

출력 S3 버킷에 데이터를 추가할 권한이 있는 IAM 서비스 역할의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*+

Required: No

status

데이터세트 내보내기 작업의 상태입니다.

데이터세트 내보내기 작업은 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하세요.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DatasetExportJobOutput

서비스: Amazon Personalize

데이터세트 내보내기 작업의 출력 구성 파라미터.

내용

s3DataDestination

S3 입력 또는 출력 버킷의 구성 세부 정보입니다.

유형: [S3DataConfig](#) 객체

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DatasetExportJobSummary

서비스: Amazon Personalize

데이터세트 내보내기 작업의 속성에 대한 요약을 제공합니다. 전체 목록을 보려면 [DescribeDatasetExportJobAPI](#)를 호출하세요.

내용

creationDateTime

데이터세트 내보내기 작업이 생성된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

datasetExportJobArn

데이터세트 내보내기 작업의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

failureReason

데이터세트 내보내기 작업이 실패할 경우, 실패의 원인.

타입: 문자열

필수사항: 아니요

jobName

데이터세트 내보내기 작업의 이름입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

lastUpdatedDateTime

데이터세트 내보내기 작업 상태가 마지막으로 업데이트된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

status

데이터세트 내보내기 작업의 상태입니다.

데이터세트 내보내기 작업은 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DatasetGroup

서비스: Amazon Personalize

데이터 세트 그룹은 관련 데이터 세트(항목 상호 작용, 사용자, 항목, 작업, 작업 상호 작용)의 모음입니다. 를 [CreateDatasetGroup](#)호출하여 데이터세트 그룹을 생성합니다. 그런 다음 데이터세트를 만들고 를 호출하여 데이터세트 그룹에 추가합니다. [CreateDataset](#) 데이터세트 그룹은 호출을 통해 솔루션을 만들고 학습시키는 데 사용됩니다. [CreateSolution](#) 데이터세트 그룹에는 한 가지 유형의 데이터세트만 포함되어 있을 수 있습니다.

AWS Key Management Service (KMS) 키를 지정하여 그룹의 데이터세트를 암호화할 수 있습니다.

내용

creationDateTime

데이터세트 그룹의 생성 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

datasetGroupArn

데이터세트 그룹의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

domain

도메인 데이터세트 그룹의 도메인.

타입: 문자열

유효 값: ECOMMERCE | VIDEO_ON_DEMAND

필수 여부: 아니요

failureReason

데이터세트 그룹 생성이 실패할 경우 이유를 제공합니다.

타입: 문자열

필수사항: 아니요

kmsKeyArn

데이터세트를 암호화하는 데 사용되는 (KMS) 키의 Amazon 리소스 이름 AWS Key Management Service (ARN).

타입: 문자열

길이 제약: 최대 길이 2048.

패턴: `arn:aws.*:kms:.*:[0-9]{12}:key/.*`

Required: No

lastUpdatedDateTime

데이터세트 그룹의 마지막 업데이트 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

name

데이터세트 그룹의 이름입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

roleArn

(KMS) 키에 액세스할 권한이 있는 AWS Identity and Access Management (IAM) 역할의 AWS Key Management Service ARN. IAM 역할을 제공하는 것은 KMS 키를 함께 지정하는 경우에만 유효합니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.\@-_/]+`

Required: No

status

데이터세트 그룹의 현재 상태입니다.

데이터세트 그룹은 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- DELETE PENDING

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DatasetGroupSummary

서비스: Amazon Personalize

데이터세트 그룹의 속성에 대한 요약を提供합니다. 전체 목록을 보려면 [DescribeDatasetGroupAPI](#)를 호출하세요.

내용

creationDateTime

데이터세트 그룹이 생성된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

datasetGroupArn

데이터세트 그룹의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

domain

도메인 데이터세트 그룹의 도메인.

타입: 문자열

유효 값: ECOMMERCE | VIDEO_ON_DEMAND

필수 여부: 아니요

failureReason

데이터세트 그룹 생성이 실패할 경우, 실패의 원인.

타입: 문자열

필수사항: 아니요

lastUpdatedDateTime

데이터세트 그룹이 마지막으로 업데이트된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

name

데이터세트 그룹의 이름입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

데이터세트 그룹의 상태입니다.

데이터세트 그룹은 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- DELETE PENDING

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DatasetImportJob

서비스: Amazon Personalize

데이터 소스(S3 버킷)에서 Personalize 데이터세트로 학습 데이터를 가져오는 작업을 설명합니다. 자세한 내용은 [참조하십시오 CreateDatasetImportJob](#).

데이터세트 가져오기 작업은 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED

내용

creationDateTime

데이터세트 가져오기 작업의 생성 날짜 및 시간(Unix 시간)

유형: 타임스탬프

필수 여부: 아니요

datasetArn

가져온 데이터를 수신하는 데이터세트의 Amazon 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

datasetImportJobArn

데이터세트 가져오기 작업의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

dataSource

가져올 학습 데이터가 들어 있는 S3 버킷.

유형: [DataSource](#) 객체

필수 여부: 아니요

failureReason

데이터세트 가져오기 작업이 실패할 경우 그 이유를 제공합니다.

타입: 문자열

필수사항: 아니요

importMode

데이터세트 가져오기 작업에서 새 레코드를 가져오는 데 사용되는 가져오기 모드.

타입: 문자열

유효 값: FULL | INCREMENTAL

필수 여부: 아니요

jobName

가져오기 작업의 이름입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

lastUpdatedDateTime

데이터세트가 마지막으로 업데이트된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

publishAttributionMetricsToS3

작업이 지표 어트리뷰션을 위해 S3에 지표를 게시하는지 여부.

타입: 부울

필수 항목 여부: 아니요

roleArn

S3 데이터 소스에서 읽을 권한이 있는 IAM 역할의 ARN입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

status

데이터세트 가져오기 작업의 상태입니다.

데이터세트 가져오기 작업은 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DatasetImportJobSummary

서비스: Amazon Personalize

데이터세트 가져오기 작업의 속성에 대한 요약을 제공합니다. 전체 목록을 보려면 [DescribeDatasetImportJob](#) API를 호출하세요.

내용

creationDateTime

데이터세트 가져오기 작업이 생성된 날짜 및 시간(Unix 시간)

유형: 타임스탬프

필수 여부: 아니요

datasetImportJobArn

데이터세트 가져오기 작업의 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

Required: No

failureReason

데이터세트 가져오기 작업이 실패할 경우, 실패의 원인.

타입: 문자열

필수사항: 아니요

importMode

데이터세트 가져오기 작업이 데이터세트의 데이터를 업데이트하는 데 사용된 가져오기 모드. 자세한 내용은 [기존 대량 데이터 업데이트](#)를 참조하세요.

타입: 문자열

유효 값: FULL | INCREMENTAL

필수 여부: 아니요

jobName

데이터세트 가져오기 작업의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

lastUpdatedDateTime

데이터세트 가져오기 작업 상태가 마지막으로 업데이트된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

status

데이터세트 가져오기 작업의 상태입니다.

데이터세트 가져오기 작업은 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DatasetSchema

서비스: Amazon Personalize

데이터세트의 스키마를 설명합니다. 스키마에 대한 자세한 내용은 [여기](#)를 참조하십시오 [CreateSchema](#).

내용

creationDateTime

스키마가 생성된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

domain

도메인 데이터세트 그룹의 데이터세트에 대해 생성한 스키마의 도메인.

타입: 문자열

유효 값: ECOMMERCE | VIDEO_ON_DEMAND

필수 여부: 아니요

lastUpdatedDateTime

스키마가 마지막으로 업데이트된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

name

스키마의 이름입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

schema

스키마.

타입: 문자열

길이 제약: 최대 길이는 20,000입니다.

필수 여부: 아니요

schemaArn

스키마의 Amazon 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DatasetSchemaSummary

서비스: Amazon Personalize

데이터세트 스키마의 속성에 대한 요약을 제공합니다. 전체 목록을 보려면 [DescribeSchemaAPI](#)를 호출하세요.

내용

creationDateTime

스키마가 생성된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

domain

도메인 데이터세트 그룹의 데이터세트에 대해 생성한 스키마의 도메인.

타입: 문자열

유효 값: ECOMMERCE | VIDEO_ON_DEMAND

필수 여부: 아니요

lastUpdatedDateTime

스키마가 마지막으로 업데이트된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

name

스키마의 이름입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

schemaArn

스키마의 Amazon 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DatasetSummary

서비스: Amazon Personalize

데이터세트의 속성에 대한 요약を提供합니다. 전체 목록을 보려면 [DescribeDatasetAPI](#)를 호출하세요.

내용

creationDateTime

데이터세트가 생성된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

datasetArn

데이터세트의 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

datasetType

데이터세트 유형. 다음 값 중 하나입니다.

- 상호작용
- Items
- 사용자
- 이벤트 상호작용

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

lastUpdatedDateTime

데이터세트가 마지막으로 업데이트된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

name

데이터세트의 이름입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

데이터세트의 상태입니다.

데이터세트는 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DatasetUpdateSummary

서비스: Amazon Personalize

데이터세트 업데이트에 대해 설명합니다.

내용

creationDateTime

데이터세트 업데이트의 생성 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

failureReason

데이터세트 업데이트가 실패할 경우 그 이유를 제공합니다.

타입: 문자열

필수사항: 아니요

lastUpdatedDateTime

데이터세트의 마지막 업데이트 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

schemaArn

데이터세트의 이전 스키마를 대체한 스키마의 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

status

데이터세트 업데이트의 상태.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DataSource

서비스: Amazon Personalize

데이터 세트에 업로드할 데이터가 포함된 데이터 소스 또는 Amazon Personalize에서 삭제할 레코드 목록을 설명합니다.

내용

dataLocation

데이터세트 가져오기 작업의 경우 데이터세트에 업로드하려는 데이터가 저장되는 Amazon S3 버킷의 경로입니다. 데이터 삭제 작업의 경우, 삭제할 레코드 목록을 저장하는 Amazon S3 버킷의 경로입니다.

예:

```
s3://bucket-name/folder-name/fileName.csv
```

CSV 파일이 Amazon S3 버킷의 폴더에 있고 가져오기 작업 또는 데이터 삭제 작업에 여러 파일을 고려하려는 경우 폴더 경로를 지정할 수 있습니다. 데이터 삭제 작업의 경우 Amazon Personalize는 폴더 및 하위 폴더의 모든 파일을 사용합니다. 폴더 이름 뒤에 /가 있는 다음 구문을 사용하세요.

```
s3://bucket-name/folder-name/
```

타입: 문자열

길이 제약: 최대 길이 256.

패턴: (s3|http|https)://.+

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DefaultCategoricalHyperParameterRange

서비스: Amazon Personalize

범주형 하이퍼파라미터의 이름과 기본 범위 및 하이퍼파라미터의 조정 가능 여부를 제공합니다. 조정 가능한 하이퍼파라미터는 하이퍼파라미터 최적화(HPO) 중 값을 결정할 수 있습니다.

내용

isTunable

하이퍼파라미터의 조정 가능 여부.

타입: 부울

필수 항목 여부: 아니요

name

하이퍼파라미터의 이름입니다.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

values

하이퍼파라미터의 범주 목록입니다.

유형: 문자열 어레이

어레이 멤버: 최대 항목 수 100개.

길이 제약: 최대 길이 1000.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)

- [AWS 루비 V3용 SDK](#)

DefaultContinuousHyperParameterRange

서비스: Amazon Personalize

지속적 하이퍼파라미터의 이름과 기본 범위 및 하이퍼파라미터의 조정 가능 여부를 제공합니다. 조정 가능한 하이퍼파라미터는 하이퍼파라미터 최적화(HPO) 중 값을 결정할 수 있습니다.

내용

isTunable

하이퍼파라미터의 조정 가능 여부.

타입: 부울

필수 항목 여부: 아니요

maxValue

하이퍼파라미터에 허용되는 최대값.

유형: Double

유효 범위: 최소값 -1000000.

필수 여부: 아니요

minValue

하이퍼파라미터에 허용되는 최소값.

유형: Double

유효 범위: 최소값 -1000000.

필수 여부: 아니요

name

하이퍼파라미터의 이름입니다.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DefaultHyperParameterRanges

서비스: Amazon Personalize

하이퍼파라미터와 해당 기본 범위를 지정합니다. 하이퍼파라미터에는 범주형, 연속 또는 정수 값이 지정될 수 있습니다.

내용

categoricalHyperParameterRanges

범주형 하이퍼파라미터와 해당 기본 범위.

유형: [DefaultCategoricalHyperParameterRange](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

필수 여부: 아니요

continuousHyperParameterRanges

지속적 하이퍼파라미터와 기본 범위.

유형: [DefaultContinuousHyperParameterRange](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

필수 여부: 아니요

integerHyperParameterRanges

정수 값 하이퍼파라미터와 해당 기본 범위.

유형: [DefaultIntegerHyperParameterRange](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)

- [AWS 루비 V3용 SDK](#)

DefaultIntegerHyperParameterRange

서비스: Amazon Personalize

정수 값 하이퍼파라미터의 이름과 기본 범위, 하이퍼파라미터의 조정 가능 여부를 제공합니다. 조정 가능한 하이퍼파라미터는 하이퍼파라미터 최적화(HPO) 중 값을 결정할 수 있습니다.

내용

isTunable

하이퍼파라미터를 조정할 수 있는지 여부를 나타냅니다.

타입: 부울

필수 항목 여부: 아니요

maxValue

하이퍼파라미터에 허용되는 최대값.

유형: 정수

유효 범위: 최대값 1000000.

필수 여부: 아니요

minValue

하이퍼파라미터에 허용되는 최소값.

유형: 정수

유효 범위: 최소값 -1000000.

필수 여부: 아니요

name

하이퍼파라미터의 이름입니다.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

EventTracker

서비스: Amazon Personalize

이벤트 트래커에 대한 정보를 제공합니다.

내용

accountId

이벤트 트래커를 소유한 AWS 계정.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

creationDateTime

이벤트 트래커가 생성된 날짜 및 시간(Unix 형식).

유형: 타임스탬프

필수 여부: 아니요

datasetGroupArn

이벤트 데이터를 수신하는 데이터세트 그룹의 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

eventTrackerArn

이벤트 트래커의 ARN.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

lastUpdatedDateTime

이벤트 트래커가 마지막으로 업데이트된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

name

이벤트 트래커의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

이벤트 트래커의 상태.

이벤트 트래커는 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

trackingId

이벤트 트래커의 ID. [PutEvents](#) API에 대한 요청에 이 ID를 포함하세요.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

EventTrackerSummary

서비스: Amazon Personalize

이벤트 트래커의 속성에 대한 요약을 제공합니다. 전체 목록을 보려면 [DescribeEventTrackerAPI](#)를 호출하세요.

내용

creationDateTime

이벤트 트래커가 생성된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

eventTrackerArn

이벤트 트래커의 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

lastUpdatedDateTime

이벤트 트래커가 마지막으로 업데이트된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

name

이벤트 트래커의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

이벤트 트래커의 상태.

이벤트 트래커는 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

FeatureTransformation

서비스: Amazon Personalize

특성 변환 정보를 제공합니다. 특성 변환은 원시 입력 데이터를 모델 학습에 더 적합한 형태로 수정하는 프로세스입니다.

내용

creationDateTime

특성 변환의 생성 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

defaultParameters

특성 변환을 위한 기본 파라미터를 제공합니다.

유형: 문자열-문자열 맵

맵 항목: 최대 항목 수 100개.

키 길이 제약: 최대 길이 256.

값 길이 제약: 최대 길이 1000.

필수 여부: 아니요

featureTransformationArn

객체의 Amazon 리소스 이름 (ARN). FeatureTransformation

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

Required: No

lastUpdatedDateTime

특성 변환의 마지막 업데이트 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

name

특성 변환의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

특성 변환의 상태.

특성 변환은 다음 중 하나의 상태일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

FieldsForThemeGeneration

서비스: Amazon Personalize

테마 생성을 위한 구성 세부 정보의 문자열-문자열 맵.

내용

itemName

데이터 세트에 있는 각 항목의 이름을 저장하는 항목 데이터 세트 열의 이름.

타입: 문자열

길이 제약: 최대 길이 150.

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Filter

서비스: Amazon Personalize

ARN, 상태, 필터 표현식 등 추천 필터에 대한 정보가 들어 있습니다.

내용

creationDateTime

필터가 생성된 시간.

유형: 타임스탬프

필수 여부: 아니요

datasetGroupArn

필터가 속한 데이터세트 그룹의 ARN.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

failureReason

필터가 실패한 경우 실패 원인.

타입: 문자열

필수사항: 아니요

filterArn

필터의 ARN.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

filterExpression

추천 결과에서 필터로 걸러낼 항목 상호작용 유형을 지정합니다. 필터 표현식은 특정 형식 규칙을 따라야 합니다. 필터 표현식 구조 및 구문에 대한 정보는 [필터 표현식](#) 단원을 참조하세요.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 2500.

필수 여부: 아니요

lastUpdatedDateTime

필터가 마지막으로 업데이트된 시간.

유형: 타임스탬프

필수 여부: 아니요

name

필터의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

필터의 상태.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

FilterSummary

서비스: Amazon Personalize

필터 속성의 간략한 요약.

내용

creationDateTime

필터가 생성된 시간.

유형: 타임스탬프

필수 여부: 아니요

datasetGroupArn

필터가 속한 데이터세트 그룹의 ARN.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

failureReason

필터가 실패한 경우 실패 원인.

타입: 문자열

필수사항: 아니요

filterArn

필터의 ARN.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

lastUpdatedDateTime

필터가 마지막으로 업데이트된 시간.

유형: 타임스탬프

필수 여부: 아니요

name

필터의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

필터의 상태.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

HPOConfig

서비스: Amazon Personalize

하이퍼파라미터 최적화(HPO)에 대한 속성을 설명합니다.

내용

algorithmHyperParameterRanges

하이퍼파라미터와 해당 허용 범위.

유형: [HyperParameterRanges](#) 객체

필수 여부: 아니요

hpoObjective

HPO 중 최적화할 지표.

Note

Personalize는 현재 hpoObjective 구성을 지원하지 않습니다.

유형: [HPOObjective](#) 객체

필수 항목 여부: 아니요

hpoResourceConfig

HPO에 대한 리소스 구성을 설명합니다.

유형: [HPOResourceConfig](#) 객체

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)

- [AWS 루비 V3용 SDK](#)

HPOObjective

서비스: Amazon Personalize

하이퍼파라미터 최적화(HPO) 중 최적화할 지표.

Note

Amazon Personalize는 현재 hpoObjective 구성을 지원하지 않습니다.

내용

metricName

지표의 이름.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

metricRegex

학습 작업 로그에서 지표를 찾기 위한 정규식.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

type

지표의 유형. 유효 값은 Maximize 및 Minimize입니다.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

HPOResourceConfig

서비스: Amazon Personalize

하이퍼파라미터 최적화(HPO)를 위한 리소스 구성을 설명합니다.

내용

maxNumberOfTrainingJobs

솔루션 버전을 생성할 때 최대 학습 작업 수. maxNumberOfTrainingJobs의 최대값은 40입니다.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

maxParallelTrainingJobs

솔루션 버전을 생성할 때 최대 병행 학습 작업 수. maxParallelTrainingJobs의 최대값은 10입니다.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

HyperParameterRanges

서비스: Amazon Personalize

하이퍼파라미터와 해당 범위를 지정합니다. 하이퍼파라미터에는 범주형, 연속 또는 정수 값이 지정될 수 있습니다.

내용

categoricalHyperParameterRanges

하이퍼파라미터와 해당 범위.

유형: [CategoricalHyperParameterRange](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

필수 여부: 아니요

continuousHyperParameterRanges

연속 하이퍼파라미터와 해당 범위.

유형: [ContinuousHyperParameterRange](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

필수 여부: 아니요

integerHyperParameterRanges

정수 값 하이퍼파라미터와 해당 범위.

유형: [IntegerHyperParameterRange](#) 객체 어레이

어레이 멤버: 최대 항목 수 100개.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)

- [AWS 루비 V3용 SDK](#)

IntegerHyperParameterRange

서비스: Amazon Personalize

정수 값 하이퍼파라미터의 이름과 범위를 제공합니다.

내용

maxValue

하이퍼파라미터에 허용되는 최대값.

유형: 정수

유효 범위: 최대값 1000000.

필수 여부: 아니요

minValue

하이퍼파라미터에 허용되는 최소값.

유형: 정수

유효 범위: 최소값 -1000000.

필수 여부: 아니요

name

하이퍼파라미터의 이름입니다.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)

- [AWS 루비 V3용 SDK](#)

MetricAttribute

서비스: Amazon Personalize

지표 어트리뷰션이 보고하는 지표에 대한 정보가 들어 있습니다. 자세한 내용은 [추천 영향 측정 단원을](#) 참조하세요.

내용

eventType

지표의 이벤트 유형.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 예

expression

속성의 표현식. 사용할 수 있는 함수는 SUM() 또는 SAMPLECOUNT()입니다. SUM() 함수의 경우 데이터세트 유형(상호작용 또는 항목)과 집계할 열을 파라미터로 제공합니다. 예: SUM(Items.PRICE).

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 예

metricName

지표의 이름. 이름은 Amazon CloudWatch 또는 Amazon S3에서 메트릭을 식별하는 데 도움이 됩니다.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

MetricAttribution

서비스: Amazon Personalize

지표 어트리뷰션에 대한 정보가 들어 있습니다. 지표 어트리뷰션은 Personalize로 가져오는 데이터에 대한 보고서를 생성합니다. 데이터를 가져오는 방법에 따라 Amazon CloudWatch 또는 Amazon S3에서 보고서를 볼 수 있습니다. 자세한 내용은 [추천 영향 측정](#) 단원을 참조하세요.

내용

creationDateTime

지표 어트리뷰션의 생성 날짜 시간.

유형: 타임스탬프

필수 여부: 아니요

datasetGroupArn

지표 어트리뷰션의 데이터세트 그룹 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

Required: No

failureReason

지표 어트리뷰션의 실패 이유.

타입: 문자열

필수사항: 아니요

lastUpdatedDateTime

지표 어트리뷰션의 마지막 업데이트 날짜 시간.

유형: 타임스탬프

필수 여부: 아니요

metricAttributionArn

지표 어트리뷰션의 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

metricsOutputConfig

지표 어트리뷰션의 출력 구성.

유형: [MetricAttributionOutput](#) 객체

필수 여부: 아니요

name

지표 어트리뷰션의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

지표 어트리뷰션의 상태.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

MetricAttributionOutput

서비스: Amazon Personalize

지표 어트리뷰션에 대한 출력 구성 세부 정보.

내용

roleArn

출력 Amazon S3 버킷에 데이터를 추가하고 Amazon에 지표를 추가할 권한이 있는 IAM 서비스 역할의 Amazon 리소스 이름 (ARN). CloudWatch 자세한 내용은 [추천 영향 측정](#) 단원을 참조하세요.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.\@-_/\]+`

필수 사항 여부: Yes

s3DataDestination

S3 입력 또는 출력 버킷의 구성 세부 정보.

유형: [S3DataConfig](#) 객체

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

MetricAttributionSummary

서비스: Amazon Personalize

지표 어트리뷰션의 속성에 대한 요약을 제공합니다. 전체 목록을 보려면 로 전화하십시오 [DescribeMetricAttribution](#).

내용

creationDateTime

지표 어트리뷰션의 생성 날짜 시간.

유형: 타임스탬프

필수 여부: 아니요

failureReason

지표 어트리뷰션의 실패 이유.

타입: 문자열

필수사항: 아니요

lastUpdatedDateTime

지표 어트리뷰션의 마지막 업데이트 날짜 시간.

유형: 타임스탬프

필수 여부: 아니요

metricAttributionArn

지표 어트리뷰션의 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

Required: No

name

지표 어트리뷰션의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

지표 어트리뷰션의 상태.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

OptimizationObjective

서비스: Amazon Personalize

스트리밍 시간 최대화 또는 수익 증대와 같은 솔루션의 추가 목표를 설명합니다. 자세한 내용은 [솔루션 최적화](#) 단원을 참조하세요.

내용

itemAttribute

최적화 목표와 관련된 항목 데이터세트의 숫자 메타데이터 열. 예: VIDEO_LENGTH(스트리밍 시간 최대화) 또는 PRICE(수익 최대화).

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 150.

필수 여부: 아니요

objectiveSensitivity

Personalize가 최적화 목표의 중요성과 관련성의 균형을 맞추는 방법을 지정합니다.

타입: 문자열

유효 값: LOW | MEDIUM | HIGH | OFF

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Recipe

서비스: Amazon Personalize

레시피에 대한 정보를 제공합니다. 각 레시피는 Amazon Personalize에서 작업을 사용할 때 모델 교육에 사용하는 알고리즘을 제공합니다. [CreateSolution](#)

내용

algorithmArn

Personalize가 모형 학습에 사용하는 알고리즘의 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

creationDateTime

레시피가 생성된 날짜 및 시간(Unix 형식).

유형: 타임스탬프

필수 여부: 아니요

description

레시피에 대한 설명.

타입: 문자열

필수사항: 아니요

featureTransformationArn

객체의 ARN입니다. FeatureTransformation

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

lastUpdatedDateTime

레시피가 마지막으로 업데이트된 날짜 및 시간(Unix 형식).

유형: 타임스탬프

필수 여부: 아니요

name

레시피의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

recipeArn

레시피의 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

recipeType

다음 값 중 하나입니다.

- PERSONALIZED_RANKING
- RELATED_ITEMS
- USER_PERSONALIZATION

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

status

레시피의 상태.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

RecipeSummary

서비스: Amazon Personalize

레시피의 속성에 대한 요약을 제공합니다. 전체 목록을 보려면 [DescribeRecipeAPI](#)를 호출하세요.

내용

creationDateTime

레시피가 생성된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

domain

레시피의 도메인(레시피가 도메인 데이터세트 그룹 사용 사례인 경우).

타입: 문자열

유효 값: ECOMMERCE | VIDEO_ON_DEMAND

필수 여부: 아니요

lastUpdatedDateTime

레시피가 마지막으로 업데이트된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

name

레시피의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

recipeArn

레시피의 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

status

레시피의 상태.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Recommender

서비스: Amazon Personalize

도메인 데이터세트 그룹의 추천 생성기에 대해 설명합니다. 특정 도메인 사용 사례 (도메인 레 시피) 에 대한 도메인 데이터세트 그룹에서 추천자를 생성하고 요청에 추천자를 지정합니다.

[GetRecommendations](#)

내용

creationDateTime

추천자가 생성된 날짜 및 시간(Unix 형식).

유형: 타임스탬프

필수 여부: 아니요

datasetGroupArn

추천자가 포함된 도메인 데이터세트 그룹의 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

Required: No

failureReason

추천자가 실패할 경우 실패의 원인.

타입: 문자열

필수사항: 아니요

lastUpdatedDateTime

추천자가 마지막으로 업데이트된 날짜 및 시간(Unix 형식).

유형: 타임스탬프

필수 여부: 아니요

latestRecommenderUpdate

추천자에 대한 최신 업데이트의 요약을 제공합니다.

유형: [RecommenderUpdateSummary](#) 객체

필수 여부: 아니요

modelMetrics

추천자의 성과를 판단하는 데 도움이 되는 평가 지표를 제공합니다. 자세한 내용은 [추천자 평가 단원](#)을 참조하세요.

유형: 문자열-double 맵

맵 항목: 최대 항목 수 100개.

키 길이 제약: 최대 길이 256.

필수 여부: 아니요

name

추천자의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

recipeArn

추천자가 생성된 레시피(도메인 데이터세트 그룹 사용 사례)의 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

recommenderArn

추천자의 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

recommenderConfig

추천자의 구성 세부 정보.

유형: [RecommenderConfig](#) 객체

필수 여부: 아니요

status

추천자의 상태.

추천자는 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- STOP PENDING > STOP IN_PROGRESS > INACTIVE > START PENDING > START IN_PROGRESS > ACTIVE
- DELETE PENDING > DELETE IN_PROGRESS

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)

- [AWS 루비 V3용 SDK](#)

RecommenderConfig

서비스: Amazon Personalize

추천자의 구성 세부 정보입니다.

내용

enableMetadataWithRecommendations

추천이 포함된 메타데이터가 추천자에 활성화되어 있는지 여부입니다. 활성화한 경우 추천 요청에 항목 데이터 세트의 열을 지정할 수 있습니다. Amazon Personalize는 추천 응답의 각 항목에 대해 이 데이터를 반환합니다. 추천자용 메타데이터 활성화에 대한 자세한 내용은 [추천자용 추천의 메타데이터 활성화](#)를 참조하세요.

추천에서 메타데이터를 활성화하면 추가 비용이 발생합니다. 자세한 내용은 [Amazon Personalize 요금](#)을 참조하세요.

타입: 부울

필수 항목 여부: 아니요

itemExplorationConfig

Personalize가 항목을 추천할 때 사용하는 항목 탐색량을 구성하는 데 사용할 탐색 구성 하이퍼파라미터(explorationWeight 및 explorationItemAgeCutOff포함)를 지정합니다. 추천자가 사용자를 위한 개인 맞춤형 추천을 생성하는 경우에만 itemExplorationConfig데이터를 제공합니다(인기 항목 또는 유사 항목 제외).

유형: 문자열-문자열 맵

맵 항목: 최대 항목 수 100개.

키 길이 제약: 최대 길이 256.

값 길이 제약: 최대 길이 1000.

필수 여부: 아니요

minRecommendationRequestsPerSecond

Personalize에서 지원하는 초당 요청된 최소 프로비저닝 추천 요청을 지정합니다. minRecommendationRequestsPerSecond가 높으면 요금이 인상됩니다. minRecommendationRequestsPerSecond(기본값)의 경우 1부터 시작

하는 것이 좋습니다. Amazon CloudWatch 지표를 사용하여 사용량을 추적하고 `minRecommendationRequestsPerSecond` 필요에 따라 사용량을 늘리십시오.

타입: 정수

유효 범위: 최소값 1.

필수 여부: 아니요

trainingDataConfig

도메인 추천자를 생성할 때 사용할 학습 데이터 구성을 지정합니다.

유형: [TrainingDataConfig](#) 객체

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

RecommenderSummary

서비스: Amazon Personalize

추천자의 속성에 대한 요약을 제공합니다.

내용

creationDateTime

추천자가 생성된 날짜 및 시간(Unix 형식).

유형: 타임스탬프

필수 여부: 아니요

datasetGroupArn

추천자가 포함된 도메인 데이터세트 그룹의 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

lastUpdatedDateTime

추천자가 마지막으로 업데이트된 날짜 및 시간(Unix 형식).

유형: 타임스탬프

필수 여부: 아니요

name

추천자의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

recipeArn

추천자가 생성된 레시피(도메인 데이터세트 그룹 사용 사례)의 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

recommenderArn

추천자의 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

recommenderConfig

추천자의 구성 세부 정보.

유형: [RecommenderConfig](#) 객체

필수 여부: 아니요

status

추천자의 상태. 추천자는 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- STOP PENDING > STOP IN_PROGRESS > INACTIVE > START PENDING > START IN_PROGRESS > ACTIVE
- DELETE PENDING > DELETE IN_PROGRESS

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

RecommenderUpdateSummary

서비스: Amazon Personalize

추천자 업데이트의 속성에 대한 요약을 제공합니다. 전체 목록을 보려면 [DescribeRecommenderAPI](#)를 호출하세요.

내용

creationDateTime

추천자 업데이트가 생성된 날짜 및 시간(Unix 형식).

유형: 타임스탬프

필수 여부: 아니요

failureReason

추천자 업데이트가 실패할 경우 실패의 원인.

타입: 문자열

필수사항: 아니요

lastUpdatedDateTime

추천자 업데이트가 마지막으로 업데이트된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

recommenderConfig

추천자 업데이트 구성 세부 정보.

유형: [RecommenderConfig](#) 객체

필수 여부: 아니요

status

추천자 업데이트의 상태.

추천자는 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- STOP PENDING > STOP IN_PROGRESS > INACTIVE > START PENDING > START IN_PROGRESS > ACTIVE
- DELETE PENDING > DELETE IN_PROGRESS

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

S3DataConfig

서비스: Amazon Personalize

S3 입력 또는 출력 버킷의 구성 세부 정보.

내용

path

S3 버킷의 파일 경로.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: (s3|http|https)://.+

필수 사항 여부: Yes

kmsKeyArn

Amazon Personalize가 입력 및 출력 파일을 암호화하거나 복호화하는 데 사용하는 AWS Key Management Service (KMS) 키의 아마존 리소스 이름 (ARN).

타입: 문자열

길이 제약: 최대 길이 2048.

패턴: arn:aws.*:kms:.*:[0-9]{12}:key/.*

필수 여부: 아니요

참고

AWS 언어별 SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Solution

서비스: Amazon Personalize

Important

솔루션을 만든 후에는 구성을 변경할 수 없습니다. 기본적으로 모든 새 솔루션은 자동 교육을 사용합니다. 자동 교육을 사용하면 솔루션이 활성화되는 동안 교육 비용이 발생합니다. 솔루션에 대한 자동 교육을 중단할 수는 없습니다. 불필요한 비용이 발생하지 않도록 하려면 완료 후 솔루션을 삭제해야 합니다. 교육 비용에 대한 자세한 내용은 [Amazon Personalize](#) 요금을 참조하십시오.

솔루션에 대한 정보를 제공하는 객체. 솔루션에는 Amazon Personalize가 권장 사항을 생성하는 데 사용하는 사용자 지정 레시피, 사용자 지정 매개 변수 및 학습된 모델 (솔루션 버전) 이 포함됩니다.

솔루션을 생성한 후에는 구성을 변경할 수 없습니다. 변경이 필요한 경우 Amazon Personalize [콘솔을 사용하여 솔루션을 복제하거나](#) 새 솔루션을 생성할 수 있습니다.

내용

autoMLResult

performAutoML이 true인 경우 찾은 최상의 레시피를 지정합니다.

유형: [AutoMLResult](#) 객체

필수 여부: 아니요

creationDateTime

솔루션 생성 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

datasetGroupArn

학습 데이터를 제공하는 데이터세트 그룹의 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

eventType

모델 학습에 사용되는 이벤트 유형(예: 'click' 또는 'like'). eventType이 제공되지 않으면 Personalize는 유형에 관계없이 동일한 가중치로 모든 상호작용을 학습에 사용합니다.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

lastUpdatedDateTime

솔루션이 마지막으로 업데이트된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

latestSolutionVersion

상태 및 ARN을 포함하여 솔루션의 최신 버전을 설명합니다.

유형: [SolutionVersionSummary](#) 객체

필수 항목 여부: 아니요

name

솔루션의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

performAutoML

Important

자동화된 기계 학습은 활성화하지 않는 것이 좋습니다. 대신 사용 사례를 사용 가능한 Personalize 레시피에 맞춥니다. 자세한 내용은 [Determining your use case](#)(사용 사례 결정)를 참조하세요.

true인 경우 Personalize는 솔루션 구성에 지정된 목록에서 최상의 USER_PERSONALIZATION 레시피 검색을 수행합니다(recipeArn은 지정하면 안 됨). false(기본값)인 경우 Personalize는 학습에 recipeArn을 사용합니다.

타입: 부울

필수 항목 여부: 아니요

performAutoTraining

솔루션에서 솔루션 버전을 자동으로 생성할지 여부를 지정합니다. 기본값은 True 이며 솔루션은 7 일마다 새 솔루션 버전을 자동으로 생성합니다.

자동 교육에 대한 자세한 내용은 [솔루션 만들기 및 구성](#)을 참조하십시오.

타입: 부울

필수 항목 여부: 아니요

performHPO

선택한 레시피에서 하이퍼파라미터 최적화(HPO)를 수행할지 여부. 기본값은 false입니다.

타입: 부울

필수 항목 여부: 아니요

recipeArn

솔루션을 생성하는 데 사용된 레시피의 ARN. performAutoML이 false인 경우 필수입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

solutionArn

솔루션의 ARN.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

solutionConfig

솔루션의 구성 속성을 설명합니다.

유형: [SolutionConfig](#) 객체

필수 항목 여부: 아니요

status

솔루션의 상태.

솔루션은 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)

- [AWS 루비 V3용 SDK](#)

SolutionConfig

서비스: Amazon Personalize

솔루션의 구성 속성을 설명합니다.

내용

algorithmHyperParameters

알고리즘 하이퍼파라미터와 해당 값을 나열합니다.

유형: 문자열-문자열 맵

맵 항목: 최대 항목 수 100개.

키 길이 제약: 최대 길이 256.

값 길이 제약: 최대 길이 1000.

필수 여부: 아니요

autoMLConfig

AutoML이 수행될 때 검색할 레시피 목록이 포함된 [AutoMLConfig](#) 객체.

유형: [AutoMLConfig](#) 객체

필수 항목 여부: 아니요

autoTrainingConfig

사용할 자동 교육 구성을 지정합니다.

유형: [AutoTrainingConfig](#) 객체

필수 여부: 아니요

eventValueThreshold

값이 이 임계값보다 크거나 같은 이벤트만 모델 학습용으로 사용됩니다.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

featureTransformationParameters

특성 변환 파라미터를 나열합니다.

유형: 문자열-문자열 맵

맵 항목: 최대 항목 수 100개.

키 길이 제약: 최대 길이 256.

값 길이 제약: 최대 길이 1000.

필수 여부: 아니요

hpoConfig

하이퍼파라미터 최적화(HPO)에 대한 속성을 설명합니다.

유형: [HPOConfig](#) 객체

필수 항목 여부: 아니요

optimizationObjective

스트리밍 시간 최대화 또는 수익 증대와 같은 솔루션의 추가 목표를 설명합니다. 자세한 내용은 [솔루션 최적화](#) 단원을 참조하세요.

유형: [OptimizationObjective](#) 객체

필수 항목 여부: 아니요

trainingDataConfig

사용자 지정 솔루션 버전(학습된 모델)을 만들 때 사용할 학습 데이터 구성을 지정합니다.

유형: [TrainingDataConfig](#) 객체

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)

- [AWS 루비 V3용 SDK](#)

SolutionSummary

서비스: Amazon Personalize

솔루션의 속성에 대한 요약을 제공합니다. 전체 목록을 보려면 [DescribeSolutionAPI](#)를 호출하세요.

내용

creationDateTime

솔루션이 생성된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

lastUpdatedDateTime

솔루션이 마지막으로 업데이트된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

name

솔루션의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

recipeArn

솔루션에서 사용하는 레시피의 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

solutionArn

솔루션의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

status

솔루션의 상태.

솔루션은 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

SolutionVersion

서비스: Amazon Personalize

사용자 지정 데이터세트 그룹에 있는 [솔루션](#)의 특정 버전에 대한 정보를 제공하는 객체.

내용

creationDateTime

솔루션의 이 버전이 생성된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

datasetGroupArn

학습 데이터를 제공하는 데이터세트 그룹의 리소스 이름(ARN).

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

Required: No

eventType

모델 학습에 사용되는 이벤트 유형(예: 'click' 또는 'like').

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

failureReason

솔루션 버전의 학습이 실패할 경우 실패의 원인.

타입: 문자열

필수사항: 아니요

lastUpdatedDateTime

솔루션이 마지막으로 업데이트된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

name

솔루션 버전의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

performAutoML

true인 경우 Personalize는 솔루션 구성에 따라 가장 최적의 레시피를 검색합니다. false(기본값)인 경우 Personalize는 recipeArn을 사용합니다.

타입: 부울

필수 항목 여부: 아니요

performHPO

선택한 레시피에서 하이퍼파라미터 최적화(HPO)를 수행할지 여부. 기본값은 false입니다.

타입: 부울

필수 항목 여부: 아니요

recipeArn

솔루션에서 사용된 레시피의 ARN.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

solutionArn

솔루션의 ARN.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

solutionConfig

솔루션의 구성 속성을 설명합니다.

유형: [SolutionConfig](#) 객체

필수 여부: 아니요

solutionVersionArn

솔루션 버전의 ARN.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*+`

Required: No

status

솔루션 버전의 상태.

솔루션 버전은 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED
- CREATE STOPPING

- CREATE STOPPED

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

trainingHours

모델을 학습시키는 데 사용된 시간. 모델을 학습시키는 데 걸리는 시간에 따라 요금이 청구됩니다. 이 필드는 Personalize가 모델을 성공적으로 학습시킨 후에만 표시됩니다.

유형: Double

유효 범위: 최소값 0.

필수 여부: 아니요

trainingMode

솔루션 버전을 만들 때 수행할 학습의 범위. FULL교육에서는 데이터셋 그룹의 모든 데이터를 고려합니다. An은 최신 교육 이후 변경된 데이터만 UPDATE 처리합니다. 사용자-개인화 레시피로 생성된 솔루션 버전만 사용할 수 있습니다. UPDATE

타입: 문자열

유효 값: FULL | UPDATE | AUTOTRAIN

필수 여부: 아니요

trainingType

솔루션 버전을 자동으로 생성했는지 수동으로 생성했는지 여부.

타입: 문자열

유효 값: AUTOMATIC | MANUAL

필수 여부: 아니요

tunedHPOParams

하이퍼파라미터 최적화를 수행한 경우, 성능이 가장 좋은 모델의 하이퍼파라미터 값이 포함됩니다.

유형: [TunedHPOParams](#) 객체

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

SolutionVersionSummary

서비스: Amazon Personalize

솔루션 버전의 속성에 대한 요약을 제공합니다. 전체 목록을 보려면 [DescribeSolutionVersion](#) API를 호출하세요.

내용

creationDateTime

솔루션의 이 버전의 생성된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

failureReason

솔루션 버전이 실패할 경우 실패의 원인.

타입: 문자열

필수사항: 아니요

lastUpdatedDateTime

솔루션 버전이 마지막으로 업데이트된 날짜 및 시간(Unix 시간).

유형: 타임스탬프

필수 여부: 아니요

solutionVersionArn

솔루션 버전의 리소스 이름(ARN)

타입: 문자열

길이 제약: 최대 길이 256.

패턴: `arn:([a-z\d-]+):personalize:.*:.*:.*`

Required: No

status

솔루션 버전의 상태.

솔루션 버전은 다음 상태 중 하나일 수 있습니다.

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE 또는 CREATE FAILED

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

trainingMode

솔루션 버전을 만들 때 수행할 학습의 범위. FULL교육에서는 데이터세트 그룹의 모든 데이터를 고려합니다. An은 최신 교육 이후 변경된 데이터만 UPDATE 처리합니다. 사용자-개인화 레시피로 생성된 솔루션 버전만 사용할 수 있습니다. UPDATE

타입: 문자열

유효 값: FULL | UPDATE | AUTOTRAIN

필수 여부: 아니요

trainingType

솔루션 버전을 자동으로 생성했는지 수동으로 생성했는지 여부.

타입: 문자열

유효 값: AUTOMATIC | MANUAL

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Tag

서비스: Amazon Personalize

리소스를 분류하고 구성하는 데 도움이 되도록 리소스에 적용하는 메타데이터. 각 태그는 사용자가 정의하는 키와 선택적 값으로 구성됩니다. 자세한 내용은 [Amazon Personalize 리소스 태그 지정을 참조](#) 하십시오.

내용

tagKey

하나의 태그를 구성하는 키-값 쌍의 일부분. 키는 더 구체적인 태그 값에 대해 범주와 같은 역할을 하는 일반적인 레이블입니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: $^([\p{L}\p{Z}\p{N}_\cdot:/+=\-\@]*)\$$

필수 사항 여부: Yes

tagValue

하나의 태그를 구성하는 키-값 쌍의 선택적 부분. 하나의 값은 태그 범주(키) 내에서 서술자 역할을 수행합니다.

타입: 문자열

길이 제약: 최소 길이는 0. 최대 길이는 256입니다.

패턴: $^([\p{L}\p{Z}\p{N}_\cdot:/+=\-\@]*)\$$

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ThemeGenerationConfig

서비스: Amazon Personalize

배치 추론 작업으로 테마를 생성하기 위한 구성 세부 정보.

내용

fieldsForThemeGeneration

배치 추론 작업에 대한 설명 테마를 생성하는 데 사용되는 필드입니다.

유형: [FieldsForThemeGeneration](#) 객체

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

TrainingDataConfig

서비스: Amazon Personalize

도메인 추천자 또는 사용자 지정 솔루션 버전(학습된 모델)을 만들 때 사용할 학습 데이터 구성.

내용

excludedDatasetColumns

학습에서 제외시킬 열을 지정합니다. 각 키는 데이터세트 유형이고 각 값은 열 목록입니다. Personalize가 추천을 생성하는 데 사용하는 데이터를 제어하는 열은 제외시킵니다.

예를 들어, 추천을 필터링하는 데만 사용하려는 열이 있을 수 있습니다. 이 열을 학습에서 제외할 수 있으며, Personalize는 필터링할 때만 이 열을 고려합니다.

유형: 문자열-문자열 어레이 맵

맵 항목: 최대 항목 수 3개.

키 길이 제약: 최대 길이 256.

어레이 멤버: 최대 항목 수 50개.

길이 제약: 최대 길이 150.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

TunedHPOParams

서비스: Amazon Personalize

하이퍼파라미터 최적화(HPO)를 수행한 경우, 성능이 가장 좋은 모델의 하이퍼파라미터 값이 포함됩니다.

내용

algorithmHyperParameters

성능이 가장 좋은 모델의 하이퍼파라미터 값 목록.

유형: 문자열-문자열 맵

맵 항목: 최대 항목 수 100개.

키 길이 제약: 최대 길이 256.

값 길이 제약: 최대 길이 1000.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Amazon Personalize 이벤트

Amazon Personalize 이벤트에서 지원하는 데이터 유형은 다음과 같습니다.

- [Action](#)
- [ActionInteraction](#)
- [Event](#)
- [Item](#)
- [MetricAttribution](#)

- [User](#)

Action

서비스: Amazon Personalize Events

PutActions API를 사용하여 작업 데이터 세트에 추가된 작업 메타데이터를 나타냅니다. 자세한 내용은 [개별적으로 작업 가져오기](#)를 참조하세요.

내용

actionId

작업과 연결된 ID.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256.

필수 여부: 예

properties

작업별 메타데이터의 문자열 맵. 맵의 각 요소는 키-값 쌍으로 이루어져 있습니다. 예를 들어 {"value": "100"}입니다.

키는 작업 데이터 세트 스키마의 필드와 일치하는 낙타 대문자 이름을 사용합니다. 이전 예제에서 value는 작업 스키마에 정의된 'VALUE' 필드와 일치합니다. 범주형 문자열 데이터의 경우 단일 작업에 여러 범주를 포함시키려면 파이프 구분자(|)로 각 범주를 구분합니다. 예: \"Deluxe|Premium\".

타입: 문자열

길이 제약: 최소 길이 1. 최대 길이는 32,000입니다.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ActionInteraction

서비스: Amazon Personalize Events

PutActionInteractions API를 사용하여 전송된 작업 상호 작용 이벤트를 나타냅니다.

내용

actionId

사용자가 상호 작용한 작업의 ID. 이는 작업 상호 작용 스키마의 ACTION_ID 필드에 해당합니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256.

필수 여부: 예

eventType

작업 상호 작용 이벤트의 유형. Viewed, Taken, Not Taken 이벤트 유형을 지정할 수 있습니다. 작업 상호 작용 이벤트 유형 데이터에 대한 자세한 내용은 [이벤트 유형 데이터](#)를 참조하세요.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256.

필수 여부: 예

sessionId

해당 사용자의 방문과 연결된 ID. 애플리케이션은 사용자가 웹 사이트를 처음 방문하거나 애플리케이션을 사용할 때 고유한 sessionId을 생성합니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256.

필수 여부: 예

timestamp

작업 상호 작용 이벤트가 발생한 시점의 타임스탬프. 타임스탬프는 Unix epoch 시간 형식(초)이어야 합니다.

유형: 타임스탬프

필수 여부: 예

eventId

이벤트와 연결된 ID. 이벤트 ID가 제공되지 않은 경우 Personalize는 이벤트에 대한 고유한 ID를 생성합니다. 이벤트 ID는 모델 입력으로 사용되지 않습니다. Amazon Personalize는 이벤트 ID를 사용하여 고유한 이벤트를 구분합니다. 동일한 이벤트 ID를 가진 첫 번째 이벤트 이후의 모든 후속 이벤트는 모델 학습에 사용되지 않습니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256.

필수 여부: 아니요

impression

사용자에게 보여준 작업의 순서를 나타내는 작업 ID 목록. 예를 들어 ["actionId1", "actionId2", "actionId3"]입니다. Amazon Personalize는 작업 상호 작용 이벤트의 노출 데이터를 사용하지 않습니다. 대신 각 작업에 대해 여러 이벤트를 기록하고 Viewed 이벤트 유형을 사용합니다.

유형: 문자열 어레이

어레이 멤버: 최소 항목 수 1개. 최대 항목 수 25개.

길이 제약: 최소 길이 1. 최대 길이는 256.

필수 여부: 아니요

properties

기록하도록 선택할 수 있는 이벤트별 데이터의 문자열 맵. 예를 들어, 사용자가 작업 ID 이외의 작업을 수행하는 경우 사용자가 수행한 작업 수를 전송할 수도 있습니다.

맵의 각 항목은 키-값 쌍으로 이루어져 있습니다. 예:

```
{"numberOfActions": "12"}
```

키는 작업 상호 작용 스키마의 필드와 일치하는 낙타 대문자 이름을 사용합니다. 위 예제에서 numberOfActions는 작업 상호 작용 스키마에 정의된 'NUMBER_OF_ACTIONS' 필드와 일치합니다.

다음은 속성의 키워드로 포함할 수 없습니다(대소문자 구분 없음).

- userId
- sessionId
- eventType
- 타임스탬프
- recommendationId
- impression

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 1024.

필수 여부: 아니요

recommendationId

사용자가 상호 작용한 작업이 포함된 추천 목록의 ID.

유형: 문자열

길이 제약: 최소 길이 1자. 최대 길이 40자.

필수 여부: 아니요

userId

작업과 상호 작용한 사용자의 ID. 이는 작업 상호 작용 스키마의 USER_ID 필드에 해당합니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Event

서비스: Amazon Personalize Events

PutEvents API를 사용하여 전송된 항목 상호 작용 이벤트 정보를 나타냅니다.

내용

eventType

이벤트의 유형(예: 클릭 또는 다운로드). 이 속성은 항목 상호 작용 데이터 세트 스키마의 EVENT_TYPE 필드에 해당하며 추적 중인 이벤트 유형에 따라 달라집니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256.

필수 여부: 예

sentAt

이벤트가 발생한 시점의 클라이언트측 타임스탬프(Unix 시간).

유형: 타임스탬프

필수 여부: 예

eventId

이벤트와 연결된 ID. 이벤트 ID가 제공되지 않은 경우 Personalize는 이벤트에 대한 고유한 ID를 생성합니다. 이벤트 ID는 모델 입력으로 사용되지 않습니다. Amazon Personalize는 이벤트 ID를 사용하여 고유한 이벤트를 구분합니다. 동일한 이벤트 ID를 가진 첫 번째 이벤트 이후의 모든 후속 이벤트는 모델 학습에 사용되지 않습니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256.

필수 여부: 아니요

eventValue

항목 상호 작용 스키마의 EVENT_VALUE 필드에 해당하는 이벤트 값.

유형: Float

필수 여부: 아니요

impression

사용자에게 보여준 항목의 순서를 나타내는 항목 ID 목록. 예를 들어 ["itemId1", "itemId2", "itemId3"]입니다. 이벤트의 노출 데이터를 수동으로 기록할 항목의 목록을 제공합니다. 노출 데이터 기록에 대한 자세한 내용은 [노출 데이터 기록](#) 단원을 참조하세요.

유형: 문자열 어레이

어레이 멤버: 최소 항목 수 1개. 최대 항목 수 25개.

길이 제약: 최소 길이 1. 최대 길이는 256.

필수 여부: 아니요

itemId

항목 상호 작용 데이터 세트 스키마의 ITEM_ID 필드에 해당하는 항목 ID 키.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256.

필수 여부: 아니요

metricAttribution

이벤트와 관련된 지표 어트리뷰션에 대한 정보가 들어 있습니다. 지표 어트리뷰션에 대한 자세한 내용은 [추천 영향 측정](#) 단원을 참조하세요.

유형: [MetricAttribution](#) 객체

필수 여부: 아니요

properties

기록하도록 선택할 수 있는 이벤트별 데이터의 문자열 맵. 예를 들어 사용자가 사이트에 있는 영화 ID(itemId) 및 등급(eventValue) 이외의 영화에 등급을 매기는 경우 사용자가 만든 영화 등급 수를 보낼 수도 있습니다.

맵의 각 항목은 키-값 쌍으로 이루어져 있습니다. 예:

```
{"numberOfRatings": "12"}
```

키는 항목 상호 작용 데이터 세트 스키마의 필드와 일치하는 낙타 대문자 이름을 사용합니다. 위 예제에서 `numberOfRatings`는 항목 상호 작용 데이터 세트 스키마에 정의된 'NUMBER_OF_RATINGS' 필드와 일치합니다.

다음은 속성의 키워드로 포함할 수 없습니다(대소문자 구분 없음).

- `userId`
- `sessionId`
- `eventType`
- 타임스탬프
- `recommendationId`
- `impression`

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 1024.

필수 여부: 아니요

`recommendationId`

사용자가 상호작용한 항목이 포함된 추천 목록의 ID. Personalize가 사용자에게 보여주는 추천을 노출 데이터로 암시적으로 기록하도록 하려면 `recommendationId`를 제공합니다. 또는 지표 어트리뷰션을 사용하여 추천의 영향을 측정하는 경우 `recommendationId`를 제공합니다.

노출 데이터 기록에 대한 자세한 내용은 [노출 데이터 기록](#) 단원을 참조하세요. 지표 어트리뷰션 생성에 대한 자세한 내용은 [추천 영향 측정](#) 단원을 참조하세요.

유형: 문자열

길이 제약: 최소 길이 1자. 최대 길이 40자.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)

- [AWS 루비 V3용 SDK](#)

Item

서비스: Amazon Personalize Events

PutItems API를 사용하여 항목 데이터셋에 추가된 항목 메타데이터를 나타냅니다. 자세한 내용은 [개별적으로 항목 가져오기](#)를 참조하세요.

내용

itemId

항목과 연결된 ID.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256.

필수 여부: 예

properties

항목별 메타데이터의 문자열 맵. 맵의 각 요소는 키-값 쌍으로 이루어져 있습니다. 예를 들어 {"numberOfRatings": "12"}입니다.

키는 항목 데이터셋 스키마의 필드와 일치하는 낙타 대문자 이름을 사용합니다. 이전 예제에서 numberOfRatings는 항목 스키마에 정의된 'NUMBER_OF_RATINGS' 필드와 일치합니다. 범주형 문자열 데이터의 경우 단일 항목에 여러 범주를 포함시키려면 파이프(|)로 각 범주를 구분합니다. 예: \"Horror|Action\".

타입: 문자열

길이 제약: 최소 길이 1. 최대 길이는 32,000입니다.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

MetricAttribution

서비스: Amazon Personalize Events

이벤트와 관련된 지표 어트리뷰션에 대한 정보가 들어 있습니다. 지표 어트리뷰션에 대한 자세한 내용은 [추천 영향 측정](#) 단원을 참조하세요.

내용

eventAttributionSource

이벤트의 출처(예: 타사).

타입: 문자열

길이 제약: 최대 길이 1024.

패턴: `^[\x20-\x7E]*[\x21-\x7E]+[\x20-\x7E]*$`

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

User

서비스: Amazon Personalize Events

PutUsers API를 사용하여 사용자 데이터세트에 추가된 항목 메타데이터를 나타냅니다. 자세한 내용은 [개별적으로 사용자 가져오기](#)를 참조하세요.

내용

userId

사용자와 연결된 ID.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256.

필수 여부: 예

properties

사용자별 메타데이터의 문자열 맵. 맵의 각 요소는 키-값 쌍으로 이루어져 있습니다. 예를 들어 {"numberOfVideosWatched": "45"}입니다.

키는 사용자 데이터세트 스키마의 필드와 일치하는 낙타 대문자 이름을 사용합니다. 이전 예제에서 numberOfVideosWatched는 사용자 스키마에 정의된 'NUMBER_OF_VIDEOS_WATCHED' 필드와 일치합니다. 범주형 문자열 데이터의 경우 단일 사용자에게 여러 범주를 포함시키려면 파이프(|)로 각 범주를 구분합니다. 예: \"Member|Frequent shopper\".

타입: 문자열

길이 제약: 최소 길이 1. 최대 길이는 24,000입니다.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Amazon Personalize 런타임

Amazon Personalize 런타임에서 지원하는 데이터 유형은 다음과 같습니다.

- [PredictedAction](#)
- [PredictedItem](#)
- [Promotion](#)

PredictedAction

서비스: Amazon Personalize Runtime

작업을 식별하는 객체.

[GetActionRecommendations](#) API는 PredictedAction 목록을 반환합니다.

내용

actionId

추천 작업의 ID.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

score

추천 작업의 점수. 작업 점수에 대한 자세한 내용은 [작업 추천 점수를 매기는 방식](#)을 참조하세요.

유형: 더블

필수 항목 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

PredictedItem

서비스: Amazon Personalize Runtime

항목을 식별하는 객체.

[GetRecommendations](#) 및 [GetPersonalizedRanking](#) API는 PredictedItem 목록을 반환합니다.

내용

itemId

추천 항목 ID.

타입: 문자열

길이 제약: 최대 길이 256.

필수 여부: 아니요

metadata

항목 데이터 세트의 항목에 대한 메타데이터입니다.

유형: 문자열-문자열 맵

키 길이 제약: 최대 길이 150.

값 길이 제약: 최대 길이는 20,000입니다.

필수 여부: 아니요

promotionName

예상 항목이 포함된 홍보의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

reason

사용자 맞춤 설정-v2를 사용하는 경우 해당 항목이 권장 사항에 포함된 이유 목록 가능한 원인은 다음과 같습니다.

- 프로모션 아이템 - 추천 요청에 적용한 프로모션의 일부로 아이템이 포함되었음을 나타냅니다.
- 탐색 - 항목이 탐색에 포함되었음을 나타냅니다. 탐색의 경우 권장 사항에는 상호 작용 데이터가 적거나 사용자와 관련성이 낮은 항목이 포함됩니다. [탐색에 대한 자세한 내용은 탐색을 참조하십시오.](#)
- 인기 항목 - 해당 항목이 자리 표시자 인기 항목으로 포함되었음을 나타냅니다. 필터를 사용하는 경우 필터가 제거하는 추천 개수에 따라 Amazon Personalize에서 추천 요청에 맞는 자리 표시자 항목을 추가할 수 있습니다. numResults 이러한 항목은 상호 작용 데이터를 기반으로 하는 인기 품목으로, 필터 기준을 충족합니다. 해당 항목에는 사용자에게 대한 관련성 점수가 없습니다.

유형: 문자열 어레이

길이 제약: 최대 길이 256.

필수 여부: 아니요

score

해당 항목이 다음 번 사용자 선택이 될 것이라는 모델의 확실성을 수치로 나타냅니다. 점수 산정 로직에 대한 자세한 내용은 [추천 점수단원을 참조하세요.](#)

유형: 더블

필수 항목 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Promotion

서비스: Amazon Personalize Runtime

홍보에 대한 정보가 들어 있습니다. 홍보는 추천 항목의 구성 가능한 하위 집합에 적용되는 추가 비즈니스 규칙을 정의합니다.

내용

filterArn

홍보에 사용되는 필터의 리소스 이름(ARN). 이 필터는 홍보 항목의 기준을 정의합니다. 자세한 내용은 [홍보 필터](#) 단원을 참조하세요.

타입: 문자열

길이 제약: 최대 길이 256.

패턴: arn:([a-z\d-]+):personalize:.*:.*:.*

Required: No

filterValues

항목을 홍보할 때 사용할 값. 홍보 표현식의 각 자리 표시자 파라미터에 대해 파라미터 이름(대/소문자 구분)을 키로 그리고 필터 값을 해당 값으로 각각 제공합니다. 한 파라미터의 여러 값을 쉼표로 구분합니다.

INCLUDE 요소를 사용하여 항목을 포함시키는 필터 표현식의 경우 표현식에 정의되어 있는 모든 파라미터의 값을 제공해야 합니다. EXCLUDE 요소를 사용하여 항목을 제외하는 표현식이 포함된 필터의 경우 filter-values를 생략할 수 있습니다. 이 경우 Personalize는 표현식의 해당 부분을 추천을 필터링하는 데 사용하지 않습니다.

필터 생성에 대한 자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 단원을 참조하세요.

유형: 문자열-문자열 맵

맵 항목: 최대 항목 수 25개.

키 길이 제약: 최대 길이 50.

키 패턴: [A-Za-z0-9_]+

값 길이 제약: 최대 길이 1000.

필수 여부: 아니요

name

홍보의 이름.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이 63.

패턴: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

percentPromotedItems

홍보를 적용할 추천 항목의 비율.

타입: 정수

유효 범위: 최소값 1. 최대값 100.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

일반적인 오류

이 단원에는 모든 AWS 서비스의 API 작업에 대한 일반 오류가 나와 있습니다. 이 서비스의 API 작업에 대한 오류는 해당 API 작업 항목을 참조하십시오.

AccessDeniedException

이 작업을 수행할 수 있는 충분한 액세스 권한이 없습니다.

HTTP 상태 코드: 400

IncompleteSignature

요청 서명이 AWS 표준을 준수하지 않습니다.

HTTP 상태 코드: 400

InternalFailure

알 수 없는 오류, 예외 또는 장애 때문에 요청 처리가 실패했습니다.

HTTP 상태 코드: 500

InvalidAction

요청된 동작 또는 작업이 유효하지 않습니다. 작업을 올바르게 입력했는지 확인합니다.

HTTP 상태 코드: 400

InvalidClientTokenId

제공된 X.509 인증서 또는 AWS 액세스 키 ID가 AWS의 레코드에 존재하지 않습니다.

HTTP 상태 코드: 403

NotAuthorized

이 작업을 수행하려면 권한이 있어야 합니다.

HTTP 상태 코드: 400

OptInRequired

AWS 액세스 키 ID는 서비스에 대한 구독이 필요합니다.

HTTP 상태 코드: 403

RequestExpired

요청이 요청상의 날짜 스탬프로부터 15분 이상, 또는 요청 만료 날짜(예: 미리 서명된 URL)로부터 15분 이상 경과한 후 서비스에 도달했거나, 요청상의 날짜 스탬프가 15분 이상 미래입니다.

HTTP 상태 코드: 400

ServiceUnavailable

서버의 일시적 장애로 인해 요청이 실패하였습니다.

HTTP 상태 코드: 503

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

ValidationError

입력이 AWS 서비스에서 지정한 제약에 충족되지 않습니다.

HTTP 상태 코드: 400

공통 파라미터

다음 목록에는 모든 작업이 쿼리 문자열을 사용하여 Signature Version 4 요청에 서명하는 데 사용하는 파라미터가 포함되어 있습니다. 작업별 파라미터는 그 작업에 대한 항목에 나열되어 있습니다. Signature Version 4에 대한 자세한 내용은 IAM 사용 설명서의 [AWS API 요청에 서명](#)을 참조하세요.

Action

수행할 작업입니다.

유형: 문자열

필수 항목 여부: 예

Version

요청이 작성되는 API 버전으로 YYYY-MM-DD 형식으로 표시됩니다.

유형: 문자열

필수 항목 여부: 예

X-Amz-Algorithm

요청 서명을 생성하는 데 사용된 해시 알고리즘입니다.

조건: HTTP 권한 부여 헤더 대신 쿼리 문자열에 인증 정보를 포함하는 경우 이 파라미터를 지정합니다.

유형: 문자열

유효한 값: AWS4-HMAC-SHA256

필수 항목 여부: 조건부

X-Amz-Credential

자격 증명 범위 값이며 액세스 키, 날짜, 대상으로 하는 리전, 요청하는 서비스 및 종료 문자열("aws4_request")이 포함된 문자열입니다. 값은 다음 형식으로 표시됩니다. access_key/YYYYMMDD/region/service/aws4_request.

자세한 내용은 IAM 사용 설명서의 [서명된 AWS API 요청 생성](#)을 참조하세요.

조건: HTTP 권한 부여 헤더 대신 쿼리 문자열에 인증 정보를 포함하는 경우 이 파라미터를 지정합니다.

유형: 문자열

필수 항목 여부: 조건부

X-Amz-Date

서명을 만드는 데 사용되는 날짜입니다. 형식은 ISO 8601 기본 형식('YYYYMMDD'T'HHMMSS'Z')이어야 합니다. 예를 들어 다음 날짜 시간은 유효한 X-Amz-Date 값: 20120325T120000Z.

조건: X-Amz-Date는 모든 요청에서 옵션이지만 서명 요청에 사용되는 날짜보다 우선할 때 사용됩니다. 날짜 헤더가 ISO 8601 기본 형식으로 지정된 경우 X-Amz-Date가 필요하지 않습니다. X-Amz-Date를 사용하는 경우 항상 Date 헤더의 값을 재정의합니다. 자세한 내용은 IAM 사용 설명서의 [AWS API 요청 서명의 요소](#)를 참조하세요.

유형: 문자열

필수 항목 여부: 조건부

X-Amz-Security-Token

AWS Security Token Service(AWS STS)에 대한 호출을 통해 받은 임시 보안 토큰입니다. AWS STS의 임시 보안 인증 정보를 지원하는 서비스 목록은 IAM 사용 설명서의 [IAM으로 작업하는 AWS 서비스](#)를 참조하세요.

조건: AWS STS의 임시 보안 인증 정보를 사용하는 경우 보안 토큰을 포함시켜야 합니다.

유형: 문자열

필수 항목 여부: 조건부

X-Amz-Signature

서명할 문자열과 파생된 서명 키에서 계산된 16진수로 인코딩된 서명을 지정합니다.

조건: HTTP 권한 부여 헤더 대신 쿼리 문자열에 인증 정보를 포함하는 경우 이 파라미터를 지정합니다.

유형: 문자열

필수 항목 여부: 조건부

X-Amz-SignedHeaders

표준 요청의 일부로 포함된 모든 HTTP 헤더를 지정합니다. 서명된 헤더 지정에 대한 자세한 내용은 IAM 사용 설명서의 [서명된 AWS API 요청 생성](#)을 참조하세요.

조건: HTTP 권한 부여 헤더 대신 쿼리 문자열에 인증 정보를 포함하는 경우 이 파라미터를 지정합니다.

유형: 문자열

필수 항목 여부: 조건부

Personalize의 문서 이력

다음 표는 Personalize 개발자 안내서의 각 릴리스에서 변경된 중요 사항에 대해 설명합니다. 이 설명서에 대한 업데이트 알림을 받으려면 RSS 피드를 구독하세요.

변경 사항	설명	날짜
새로운 특성	Amazon Personalize는 이제 데이터세트 그룹에서 메타데이터 및 상호 작용 데이터를 비롯한 사용자 및 데이터를 삭제하는 기능을 지원합니다. 자세한 내용은 사용자 삭제 를 참조하십시오.	2024년 5월 2일
새 기능	Amazon Personalize는 이제 더 빠른 교육, 더 짧은 추천 지연 시간, 더 관련성 높은 추천을 통해 최대 5백만 개의 항목을 고려할 수 있습니다. 사용자 개인화-v2와 맞춤형 순위-v2라는 두 가지 새로운 사용자 지정 레시피가 이 향상된 교육을 사용합니다.	2024년 5월 1일
새 기능	모든 새로운 Amazon Personalize 사용자 지정 솔루션은 이제 자동 교육을 사용합니다. 자동 교육 구성에 대한 자세한 내용은 자동 교육 구성 을 참조하십시오.	2024년 4월 19일
새 기능	Amazon Personalize는 이제 사용자의 행동을 기반으로 사용자에게 대한 차선택 작업을 추천하는 기능을 지원합니다. 자	2023년 11월 26일

세한 내용은 [차선택-작업 레시피](#)를 참조하세요.

새 기능

Amazon Personalize는 이제 생성형 AI의 도움을 받아 배치 추천에 설명적 주제를 포함하는 기능을 지원합니다. 테마가 있는 배치 추천 생성에 대한 자세한 내용은 [테마가 있는 배치 추천](#)을 참조하세요. Amazon Personalize를 생성형 AI와 함께 사용하는 방법에 대한 자세한 내용은 [Amazon Personalize 및 생성형 AI](#)를 참조하세요.

2023년 11월 26일

새 기능

Amazon Personalize는 이제 추천에 항목 데이터 세트의 메타데이터를 포함하는 기능을 지원합니다. 캠페인의 메타데이터를 활성화하는 방법에 대한 자세한 내용은 [권장 사항의 항목 메타데이터](#)를 참조하십시오. 추천자용 메타데이터 활성화에 대한 자세한 내용은 [추천자용 추천의 메타데이터 활성화](#)를 참조하세요.

2023년 11월 26일

새 기능

Amazon Personalize는 이제 서비스에서 검색 결과를 개인화하는 기능을 지원합니다. OpenSearch 자세한 내용은 [검색 결과 개인 설정을 참조하십시오](#). OpenSearch

2023년 10월 16일

새로운 특성

Personalize는 이제 항목 데이터 세트에 대해 최대 100개의 메타데이터 열(50개에서 증가), 사용자 데이터 세트에 대해 최대 25개(5개에서 증가)의 메타데이터 열을 가져올 수 있는 기능을 지원합니다. Personalize 제한에 대한 자세한 내용은 [Personalize 엔드포인트 및 할당량](#) 단원을 참조하세요.

2023년 9월 5일

새로운 특성

Amazon Personalize는 이제 검색 결과를 개인화하는 기능을 지원합니다. OpenSearch 자세한 내용은 [검색 결과 개인 설정 OpenSearch \(자체 관리\)](#) 을 참조하십시오.

2023년 7월 25일

새로운 특성

Personalize는 이제 데이터 세트의 스키마를 새 스키마 또는 기존 스키마로 교체하는 기능을 지원합니다. 자세한 내용은 [데이터 세트 스키마 교체](#)를 참조하세요.

2023년 7월 13일

새로운 특성

사용자 개인화 또는 맞춤형 순위 레시피를 사용하는 경우, Personalize 배치 추론 작업에서 이제 재교육 없이 점진적으로 가져오는 데이터를 사용할 수 있습니다. 자세한 내용은 [배치 추천 가져오기](#)를 참조하세요.

2023년 6월 30일

새로운 특성

Personalize는 이제 관련 항목 추천 요청에서 지정한 항목을 기준으로 항목을 필터링하는 기능을 지원합니다. 필터에 대한 자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 단원을 참조하세요.

2023년 6월 21일

새로운 특성

Personalize는 이제 관련 항목 추천 요청에서 지정한 항목을 기준으로 항목을 필터링하는 기능을 지원합니다. 필터에 대한 자세한 내용은 [추천 및 사용자 세그먼트 필터링](#) 단원을 참조하세요.

2023년 6월 21일

새로운 특성

이제 Personalize는 인터페이스 VPC 엔드포인트를 사용하여 가상 프라이빗 클라우드 (VPC)와 Personalize 간에 프라이빗 연결을 지원합니다. 자세한 내용은 [Personalize 및 인터페이스 VPC 엔드포인트 \(AWS PrivateLink\)](#)를 참조하세요.

2023년 6월 12일

새로운 특성

Personalize는 이제 추천 솔루션 또는 사용자 지정 솔루션을 생성할 때 교육 시 사용되는 열을 구성할 수 있도록 지원합니다. 추천을 생성할 때 열을 구성하는 방법에 대한 자세한 내용은 [추천 생성](#)을 참조하세요. 솔루션을 만들 때 열을 구성하는 방법에 대한 자세한 내용은 [교육 시 사용되는 열 구성](#)을 참조하세요.

2023년 5월 30일

[새로운 문서 특성](#)

Personalize 개발자 안내서에
는 이제 Personalize 추천을 사
용한 A/B 테스트 수행에 대한
정보가 포함되어 있습니다. 자
세한 내용은 [A/B 테스트를 통한
추천 영향 측정](#)을 참조하세요.

2023년 5월 5일

[새로운 특성](#)

Personalize는 이제 인기도가
유사 항목 레시피로 생성된 추
천에 미치는 영향을 구성하는
것을 지원합니다. 자세한 내용
은 [유사 항목 레시피](#)를 참조하
세요.

2023년 4월 21일

[새로운 특성](#)

Amazon Personalize는 이제 아
마존 SageMaker 데이터 랭글
러를 사용하여 40개 이상의 소
스에서 Amazon Personalize 데
이터 세트로 데이터를 가져올
수 있도록 지원합니다. 자세한
내용은 [Amazon SageMaker 데
이터 랭글러를 사용하여 데이
터 가져오기](#)를 참조하십시오.

2023년 4월 14일

[새로운 문서 특성](#)

Personalize 개발자 안내서에
이제 새로운 준비 체크리스
트가 포함되어 있습니다. 이
체크리스트는 자체 데이터로
Personalize를 사용할 준비를
하는 데 도움이 됩니다. 자세한
내용은 [준비 체크리스트](#)를 참
조하세요.

2023년 2월 9일

새로운 특성	Personalize는 이제 데이터세트로 가져오는 데이터에 대한 인사이트 및 통계 생성을 지원합니다. 자세한 내용은 데이터 세트에서 데이터 분석 을 참조하세요.	2023년 1월 25일
새로운 특성	Personalize는 이제 사용자 지정 데이터세트 그룹을 위한 새로운 지금-유행 레시피를 지원합니다. 자세한 내용은 지금-유행 레시피 를 참조하세요.	2023년 1월 6일
새로운 특성	Personalize는 이제 IAM 정책에서 태그 사용을 지원합니다. 자세한 내용은 IAM 정책에서 태그 사용 을 참조하세요.	2022년 12월 28일
새로운 특성	Personalize는 이제 교육 진행 중 모델이 고려하는 최대 상호작용의 수 할당량에 대한 조정을 지원합니다. 또한 교육 진행 중 모델이 고려하는 최대 사용자의 수 한도 할당량은 더 이상 적용되지 않습니다. 자세한 내용은 Personalize 엔드포인트 및 할당량 을 참조하세요.	2022년 12월 15일
새로운 특성	Personalize는 이제 추천이 비즈니스에 미치는 영향을 측정하기 위한 지표 속성 생성을 지원합니다. 자세한 내용은 추천 영향 측정 단원을 참조하세요.	2022년 11월 17일

새로운 특성

활성 솔루션, 활성 캠페인, 추천자 및 필터의 총 수에 대한 Personalize 할당량이 이제 늘어났습니다. 이제 이러한 각 할당량은 계정별로가 아니라 데이터세트 그룹별로 적용됩니다. 할당량에 대한 자세한 내용은 [Personalize](#) 엔드포인트 및 할당량을 참조하세요.

2022년 9월 7일

새로운 특성

Personalize 필터는 이제 이벤트 유형별로 사용자당 최대 100개의 상호작용을 고려합니다. 필터에 대한 자세한 내용은 [추천 및 사용자 세그먼트 필터링 단원을 참조하세요](#).

2022년 8월 29일

새로운 특성

Personalize는 이제 VIDEO_ON_DEMAND 도메인에 대한 새로운 Trending 사용 사례를 지원합니다. 자세한 내용은 [VIDEO_ON_DEMAND 사용 사례](#)를 참조하세요.

2022년 8월 17일

새로운 특성

Personalize는 이제 별도의 프로모션 필터를 사용하여 추천 항목의 프로모션을 지원합니다. 항목 홍보에 대한 자세한 내용은 [추천에서 항목 홍보를 참조하세요](#).

2022년 8월 12일

새로운 특성

Personalize는 이제 자리 표시자 파라미터가 있는 필터 표현식에서 비교 연산자를 사용할 수 있도록 지원합니다. 필터 표현식에 대한 자세한 내용은 [필터 표현식 단원을 참조하세요](#).

2022년 8월 12일

새로운 특성

Personalize는 이제 데이터세트에 대한 점진적 대량 업데이트를 지원합니다. 이제 기존 데이터를 바꾸지 않고서도 데이터세트 가져오기 작업을 사용하여 데이터세트를 업데이트할 수 있습니다. 자세한 내용은 [기존 대량 데이터 업데이트](#)를 참조하세요.

2022년 8월 2일

새로운 특성

Personalize는 이제 다양한 언어의 비정형 텍스트 메타데이터를 사용할 수 있습니다. 자세한 내용은 [비정형 텍스트 메타데이터](#)를 참조하세요.

2022년 6월 6일

새로운 특성

Personalize 추천자는 이제 오프라인 지표를 생성할 수 있습니다. 이러한 지표를 사용하여 추천자의 성과를 평가할 수 있습니다. 자세한 내용은 [추천자 평가](#) 단원을 참조하세요.

2022년 5월 24일

새로운 특성

Personalize는 이제 추천자를 중지하고 나중에 다시 시작하는 기능을 지원합니다. 이렇게 하면 추천인 청구를 일시 중지하고 사용한 경우에만 비용을 지불할 수 있습니다. 자세한 내용은 [추천자 중단 및 시작](#)을 참조하세요.

2022년 4월 20일

새로운 특성	Personalize는 이제 태그를 사용하여 아마존 Personalize 리소스를 분류하고 관리할 수 있도록 지원합니다. 자세한 내용은 Personalize 리소스 태그 지정 을 참조하세요.	2022년 4월 7일
새로운 특성	Amazon Personalize는 이제 리소스 지정을 지원합니다. AWS CloudFormation 자세한 내용은 리소스 지정 을 참조하십시오. AWS CloudFormation	2022년 3월 11일
새로운 문서 특성	Personalize 개발자 안내서에는 이제 Personalize에서 발생할 수 있는 오류 메시지에 대한 공통 질문에 대한 답변과 문제 해결 조언을 제공하는 새로운 문제 해결 주제가 포함되어 있습니다. 자세한 설명은 문제 해결 을 참조하십시오.	2022년 2월 15일
새로운 특성	Personalize는 이제 온디맨드 비디오 또는 전자 상거래 도메인을 위한 사용 사례에 최적화된 리소스가 포함된 도메인 데이터세트 그룹 생성을 지원합니다. 자세한 내용을 알아보려면 데이터세트 그룹 을 참조하세요.	2021년 11월 29일

새로운 특성

Personalize는 이제 새로운 사용자_세분화 레시피를 사용하여 사용자 세그먼트 생성을 지원합니다. 사용자_세분화 레시피는 항목 입력 데이터를 기반으로 사용자 세그먼트를 생성합니다. 자세한 내용은 [USER_Segmentation 레시피](#)를 참조하세요.

2021년 11월 29일

새로운 특성

Personalize는 이제 새로운 RELATED_ITEMS 추천 레시피 유사 항목을 지원합니다. 유사 항목 레시피를 사용하여 상호 작용 데이터와 항목 메타데이터를 기반으로 유사 항목에 대한 추천을 생성할 수 있습니다. 자세한 내용은 [유사 항목 레시피](#)를 참조하세요.

2021년 10월 5일

새로운 문서 특성

Personalize 개발자 안내서에는 이제 Java 2.x용 SDK와 함께 Personalize를 사용하기 위한 시작 자습서가 포함되어 있습니다. 자세한 내용은 [시작하기\(Java 2.x용 SDK\)](#)를 참조하세요.

2021년 8월 25일

새로운 특성

Personalize는 이제 항목 데이터세트의 비정형 텍스트 메타데이터에서 의미 있는 정보를 추출할 수 있습니다. 자세한 내용은 [항목 데이터세트](#) 단원을 참조하세요.

2021년 6월 9일

새로운 특성	Personalize는 이제 솔루션 버전 생성을 중지(모델 교육 중지)하는 기능을 지원합니다. 자세한 내용은 솔루션 버전 생성 중단 을 참조하세요.	2021년 5월 20일
새로운 특성(미리보기 릴리스)	Personalize는 이제 수익 극대화 같은 관련성을 극대화하는 것 이외에도 목표에 맞게 솔루션을 최적화할 수 있습니다. 이 특성은 미리보기 릴리스에 있습니다. 자세한 내용은 추가 목표를 위한 솔루션 최적화 를 참조하세요.	2021년 5월 18일
새로운 특성	Personalize는 이제 분석 및 추적을 위해 Personalize 데이터 세트의 레코드를 S3 버킷으로 내보낼 수 있습니다. 자세한 내용은 데이터세트 내보내기 를 참조하세요.	2021년 4월 26일
새로운 특성	Personalize는 이제 사용자 개인화를 통해 학습한 최신 모델(솔루션 버전)을 2시간마다 새 데이터를 포함하도록 자동으로 업데이트할 수 있습니다. 자세한 내용은 사용자 개인화 레시피 를 참조하세요.	2020년 11월 17일
새로운 특성	Personalize는 이제 추천을 받을 때 지정하는 기준에 따라 추천을 필터링할 수 있습니다. 자세한 내용은 추천 필터링 단원 을 참조하세요.	2020년 11월 10일

<u>새로운 특성</u>	Personalize는 이제 사용자 및 항목을 점진적으로 가져오는 기능을 지원합니다. 자세한 내용은 <u>점진적으로 레코드 가져오기</u> 를 참조하세요.	2020년 10월 2일
<u>새로운 특성</u>	Personalize는 이제 새로운 USER_PERSONALIZATION 추천 레시피를 지원합니다. USER_PERSONALIZATION 특성에는 노출 데이터 모델링, 자동 항목 탐색 및 자동 콜드 항목 선택이 포함됩니다. 자세한 내용은 <u>사용자 개인화 레시피</u> 를 참조하세요.	2020년 8월 5일
<u>새로운 특성</u>	Personalize는 이제 사용자 지정 필터 표현식을 사용하여 항목 및 사용자 메타데이터를 기반으로 추천을 필터링할 수 있습니다. 자세한 내용은 <u>추천 필터링 단원을 참조하세요</u> .	2020년 7월 31일
<u>새로운 특성</u>	Personalize에서는 이제 사용자가 상호작용한 항목을 기준으로 결과를 필터링할 수 있습니다. 자세한 내용은 <u>추천 필터링 단원을 참조하세요</u> .	2020년 6월 3일
<u>새로운 특성</u>	Personalize에는 이제 권장 항목에 대한 점수가 노출됩니다. 점수는 Personalize 모델에서 사용자가 다음에 특정 항목을 선택할 확실성을 나타냅니다. 자세한 내용은 <u>추천 가져오기</u> 를 참조하세요.	2020년 4월 3일

새로운 리전	Personalize에서는 아시아 태평양(서울) 리전에 대한 지원을 추가하였습니다. Personalize에서 지원하는 AWS 리전의 전체 목록은 Web Services 일반 참조의 AWS 리전 표 또는 AWS 리전 및 엔드포인트를 참조하세요.	2020년 1월 21일
새로운 특성	Personalize에서 이제 컨텍스트 메타데이터를 기반으로 추천을 가져올 수 있습니다. 자세한 내용은 추천 가져오기 를 참조하세요.	2019년 12월 19일
새로운 리전	Personalize에서 아시아 태평양(뭄바이), 아시아 태평양(시드니), 캐나다(중부) 리전에 대한 지원이 추가됩니다. Amazon Personalize에서 지원하는 AWS 지역의 전체 목록은 Amazon Web Services 일반 참조의 AWS 지역 표 또는 AWS 지역 및 엔드포인트를 참조하십시오.	2019년 12월 18일
새로운 특성	Personalize에서 이제 배치 권장 워크플로우를 지원합니다. 자세한 내용은 배치 추천 가져오기 를 참조하세요.	2019년 11월 14일
Personalize 정식 출시	이제 Personalize를 일반적인 용도로 사용할 수 있습니다.	2019년 6월 10일
Personalize 미리 보기 릴리스	이 문서는 Personalize에 대한 설명서의 최초 미리 보기 릴리스입니다.	2018년 11월 28일

AWS 용어집

최신 AWS 용어는 AWS 용어집 참조서의 [AWS 용어집](#)을 참조하세요.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.