



開発者ガイド

# Amazon Forecast



# Amazon Forecast: 開発者ガイド

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

Amazon の商標とトレードドレスは、Amazon 以外の製品またはサービスとの関連において、顧客に混乱を招いたり、Amazon の名誉または信用を毀損するような方法で使用することはできません。Amazon が所有しない他の商標はすべてそれぞれの所有者に帰属します。所有者は必ずしも Amazon との提携や関連があるわけではありません。また、Amazon の支援を受けているとはかぎりません。

# Table of Contents

Amazon Forecast とは .....	1
Amazon Forecast の使用 .....	2
Amazon Forecast の特長 .....	2
Amazon Forecast の料金 .....	3
Amazon Forecast を初めてお使いになる方向けの情報 .....	3
AWS SDKsの使用 .....	3
Amazon Forecast の仕組み .....	5
セットアップ .....	6
AWS へのサインアップ .....	6
AWS CLI のセットアップ .....	6
アクセス権限の設定 .....	7
Amazon Forecast の IAM ロールを作成する (IAM コンソール) .....	8
Amazon Forecast の IAM ロールを作成する (AWS CLI) .....	10
サービス間の混乱した代理の防止 .....	14
使用開始 .....	15
入力データを準備する .....	16
開始方法 (コンソール) .....	17
開始方法 (AWS CLI) .....	31
開始方法 (Python ノートブック) .....	45
高度なチュートリアル .....	46
リソースのクリーンアップ .....	47
チュートリアル .....	49
AWS CloudFormation による自動化 .....	49
前提条件 .....	50
Forecast のオートメーションのための AWS CloudFormation テンプレートのデプロイ .....	51
クリーンアップ .....	53
データセットのインポート .....	54
データセット .....	54
データセットドメインとデータセットタイプ .....	55
データセットスキーマ .....	57
データセットグループ .....	59
データ収集頻度の競合の解決 .....	59
関連する時系列 .....	59
履歴およびフォワードルックの関連する時系列 .....	60

関連する時系列データセットの検証 .....	61
例: フォワードルックの関連する時系列ファイル .....	62
例: 詳細度の予測 .....	64
レガシー予測子および関連する時系列 .....	64
項目メタデータ .....	65
例: 項目メタデータファイルとスキーマ .....	66
レガシー予測子およびアイテムメタデータ .....	68
その他の参照資料 .....	68
事前定義済みのデータセットドメインとデータセットタイプ .....	68
RETAIL ドメイン .....	71
CUSTOM ドメイン .....	72
INVENTORY_PLANNING ドメイン .....	74
EC2 CAPACITY ドメイン .....	75
WORK_FORCE ドメイン .....	76
WEB_TRAFFIC ドメイン .....	78
METRICS ドメイン .....	79
データの更新 .....	80
インポートモード .....	81
既存のデータセットの更新 .....	82
予想の更新 .....	83
欠落した値の処理 .....	83
フィルロジックの選択 .....	84
ターゲット時系列および関連する時系列フィルロジック .....	85
欠落した値の構文 .....	86
データセットのガイドライン .....	87
予測子のトレーニング .....	91
予測子の作成 .....	91
へのアップグレード AutoPredictor .....	95
データ集計 .....	96
集計の仕組み .....	97
時間境界 .....	98
データ集計の想定 .....	102
追加のデータセットの使用 .....	103
レガシー予測子の使用 .....	103
予測子のメトリクス .....	104
精度メトリクスの解釈 .....	105

重み付き分位損失 (wQL) .....	106
重み付き絶対誤差率 (WAPE) .....	108
二乗平均平方根誤差 (RMSE) .....	108
平均絶対パーセント誤差 (MAPE) .....	109
平均絶対スケーリング誤差 (MASE) .....	110
精度メトリクスのエクスポート .....	111
Forecast タイプの選択 .....	113
レガシー予測子の使用 .....	115
予測子の再トレーニング .....	118
Weather Index .....	120
Weather Index の有効化 .....	121
データセットへの geolocation の追加 .....	122
タイムゾーンの指定 .....	132
条件と制限 .....	137
Holidays の特徴量化 .....	139
Holidays の特徴量化の有効化 .....	139
国コード .....	140
追加の祝祭日カレンダー .....	153
Predictor Explainability .....	154
影響スコアの解釈 .....	155
Predictor Explainability の作成 .....	156
Predictor Explainability のエクスポート .....	159
制約事項とベストプラクティス .....	160
予測子のモニタリング .....	161
予測子モニタリングワークフロー .....	162
予測子モニタリングを有効にする .....	163
モニタリング結果の表示 .....	166
制約事項とベストプラクティス .....	169
Forecast アルゴリズム .....	170
ビルトイン Forecast アルゴリズム .....	170
Forecast アルゴリズムの比較 .....	171
ARIMA .....	173
CNN-QR .....	174
DeepAR+ .....	181
ETS .....	190
NPTS .....	191

Prophet .....	195
予測の生成 .....	197
予想を作成する .....	197
時系列の指定 .....	199
予想をエクスポートする .....	200
予想をクエリする .....	203
コールドスタート予想 .....	203
Forecast Explainability .....	205
影響スコアの解釈 .....	206
Forecast Explainability の作成 .....	207
時系列の指定 .....	207
時間ポイントの指定 .....	210
Forecast Explainability の視覚化 .....	212
Forecast Explainability のエクスポート .....	213
制約事項とベストプラクティス .....	215
What-If 分析 .....	216
What-If 分析の作成 .....	216
What-If 分析の作成 .....	217
What-If 予想の作成 .....	218
What-If 予想を比較する .....	221
What-If 予想をエクスポートする .....	222
What-If 予想をクエリする .....	222
変換関数 .....	223
代替データセット .....	230
予測ディメンション .....	234
リソースの管理 .....	236
リソースの停止 .....	236
リソースの削除 .....	238
リソースツリーについて .....	238
個々のリソースの削除 .....	240
リソースツリーの削除 .....	241
リソースのタグ付け .....	242
タグを管理する .....	243
IAM ポリシーでのタグの使用 .....	244
リソースにタグを追加する .....	245
追加情報 .....	247

通知の受信 .....	247
Forecast リソースジョブのモニタリング .....	248
ジョブステータス通知用の EventBridge ルールの作成 .....	250
ジョブステータス通知用の CloudWatch Events ルールの作成 .....	251
ガイドラインとクォータ .....	252
AWS でサポートされているリージョン .....	252
コンプライアンス .....	252
Service Quotas .....	252
条件と制限 .....	257
予約フィールド名 .....	260
コードの例 .....	290
アクション .....	290
CreateDataset .....	291
CreateForecast .....	294
DeleteDataset .....	296
DeleteForecast .....	298
DescribeForecast .....	300
ListDatasetGroups .....	302
ListForecasts .....	303
セキュリティ .....	306
データ保護 .....	307
保管時の暗号化 .....	308
転送時と処理時の暗号化 .....	308
Amazon Forecast が AWS KMS で許可を使用する方法 .....	308
カスタマーマネージドキーを作成する .....	309
Amazon Forecast Service の暗号化キーを監視する .....	311
Identity and Access Management .....	315
対象者 .....	316
アイデンティティを使用した認証 .....	317
ポリシーを使用したアクセスの管理 .....	320
IAM を使用する Amazon Forecast の仕組み .....	323
アイデンティティベースポリシーの例 .....	330
トラブルシューティング .....	339
ログ記録とモニタリング .....	341
を使用した Forecast API コールのログ記録 AWS CloudTrail .....	341
CloudWatch Amazon Forecast のメトリクス .....	344

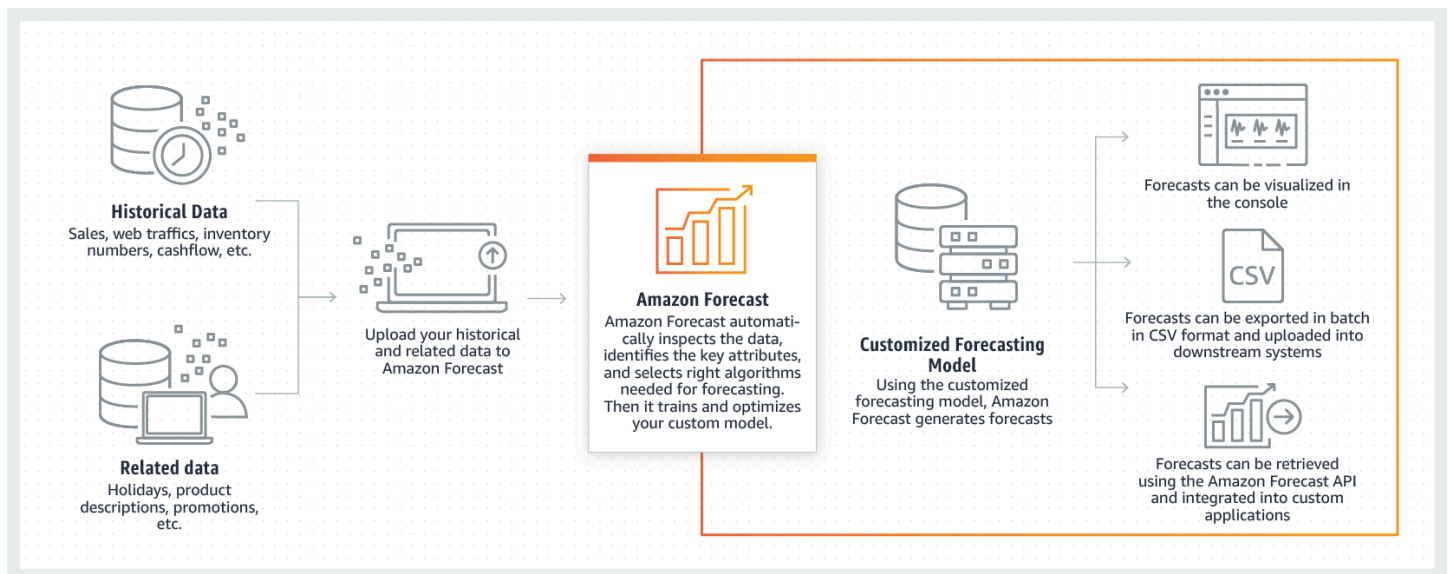
コンプライアンス検証 .....	346
耐障害性 .....	347
インフラストラクチャセキュリティ .....	347
VPC エンドポイントAWS PrivateLink .....	348
Forecast VPC エンドポイントの考慮事項 .....	349
Forecast 用のインターフェイス VPC エンドポイントの作成 .....	349
Forecast 用の VPC エンドポイントポリシーの作成 .....	349
API リファレンス .....	352
アクション .....	352
Amazon Forecast サービス .....	354
Amazon Forecast クエリサービス .....	626
データ型 .....	634
Amazon Forecast サービス .....	637
Amazon Forecast クエリサービス .....	778
共通エラー .....	781
共通パラメータ .....	782
ドキュメント履歴 .....	785
AWS 用語集 .....	789
.....	dccxc



# Amazon Forecast とは

Amazon Forecast は、統計アルゴリズムと機械学習アルゴリズムを使用して、非常に正確な時系列予測を実現するフルマネージドサービスです。Forecast は、Amazon.com で時系列予測に使用されるのと同じテクノロジーに基づいて、履歴データに基づいて将来の時系列データを予測するための state-of-the-art アルゴリズムを提供し、機械学習の経験を必要としません。

時系列の予測は、小売、金融、物流、ヘルスケアなどの複数の分野で有益です。Forecast を使用して、在庫、労働力、ウェブトラフィック、サーバー容量、および財務に関するドメイン固有のメトリクスを予測することもできます。



Amazon Forecast の技術的側面の詳細については、「[Time Series Forecasting Principles with Amazon Forecast](#)」(Amazon Forecast を使用した時系列予測プリンシプル) を参照してください。

## トピック

- [Amazon Forecast の使用](#)
- [Amazon Forecast の特長](#)
- [Amazon Forecast の料金](#)
- [Amazon Forecast を初めてお使いになる方向けの情報](#)
- [AWS SDK での Forecast の使用](#)

# Amazon Forecast の使用

[API](#)、[AWS Command Line Interface \(AWS CLI\)](#)、[Python Software Development Kit \(SDK\)](#)、および [Amazon Forecast コンソール](#) を使用して、時系列データセットをインポートし、予測子をトレーニングし、予想を生成できます。

ここでは、Amazon Forecast の一般的なユースケースを示します。

- 小売の需要計画 – 製品需要を予測し、さまざまな店舗の場所に合わせて、在庫と価格をより正確に設定できるようにします。
- サプライチェーン計画 – 製造に必要な原材料、サービス、またはその他の項目の量を予測します。
- リソース計画 – 人員配置、広告、エネルギー消費、およびサーバー容量の要件を予測します。
- 運用計画 – ウェブトラフィック、AWS 使用状況、IoT センサーの使用状況のレベルを予測します。

## Amazon Forecast の特長

Amazon Forecast は、時系列予測プロセスの多くを自動化し、データセットの準備と予測の解釈に注力できるようにします。

Forecast には次の機能があります。

- 自動機械学習 – Forecast は、データセットに最適な機械学習アルゴリズムの組み合わせを見つけ、複雑な機械学習タスクを自動化します。
- State-of-the-art アルゴリズム – Amazon.com で使用されているのと同じテクノロジーに基づく機械学習アルゴリズムの組み合わせを適用します。Forecast は、一般的に使用される統計手法から複雑なニューラルネットワークまで、幅広いトレーニングアルゴリズムを提供します。
- 欠落値のサポート – Forecast では、データセット内の欠落値を自動的に処理するためのいくつかの filling メソッドが用意されています。
- 追加のビルトインデータセット – Forecast は、モデルを改善するためにビルトインデータセットを自動的に組み込むことができます。これらのデータセットは既に特徴エンジニアリングされており、追加の設定は必要ありません。

## Amazon Forecast の料金

Amazon Forecast では、実際に使用した分の料金のみをお支払いいただきます。最低料金や前払いの義務は発生しません。Amazon Forecast のコストは、生成された予測の数、データストレージ、およびトレーニング時間によって異なります。

[AWS 無料利用枠](#)により、1 か月あたり、最大 10,000 の時系列の予想、最大 10 GB のストレージ、および最長 10 時間のトレーニング時間の制限内でご利用いただけます。Amazon Forecast の無料利用枠は、使用を開始してから最初の 2 か月間有効です。

課金および料金の詳細な一覧については、[Amazon Forecast の料金](#)を参照してください。

## Amazon Forecast を初めてお使いになる方向けの情報

Amazon Forecast を初めて使用する方には、以下のページから開始することをお勧めします。

1. [Amazon Forecast の仕組み](#) – データセットのインポート、予測子の作成、および予測の生成の主要な概念とプロセスについて説明します。
2. [使用開始](#) – いずれかのチュートリアルに従って、最初の Amazon Forecast の予測子を作成します。
3. [API リファレンス](#) – Amazon Forecast API のアクションとデータ型についてご確認いただけます。

## AWS SDK での Forecast の使用

AWS Software Development Kit (SDKs) は、多くの一般的なプログラミング言語で使用できます。各 SDK には、デベロッパーが好みの言語でアプリケーションを簡単に構築できるようにする API、コード例、およびドキュメントが提供されています。

SDK ドキュメント	コード例
<a href="#">AWS SDK for C++</a>	<a href="#">AWS SDK for C++ コード例</a>
<a href="#">AWS CLI</a>	<a href="#">AWS CLI コード例</a>
<a href="#">AWS SDK for Go</a>	<a href="#">AWS SDK for Go コード例</a>

SDK ドキュメント	コード例
<a href="#">AWS SDK for Java</a>	<a href="#">AWS SDK for Java コード例</a>
<a href="#">AWS SDK for JavaScript</a>	<a href="#">AWS SDK for JavaScript コード例</a>
<a href="#">AWS SDK for Kotlin</a>	<a href="#">AWS SDK for Kotlin コード例</a>
<a href="#">AWS SDK for .NET</a>	<a href="#">AWS SDK for .NET コード例</a>
<a href="#">AWS SDK for PHP</a>	<a href="#">AWS SDK for PHP コード例</a>
<a href="#">AWS Tools for PowerShell</a>	<a href="#">PowerShell コード例のツール</a>
<a href="#">AWS SDK for Python (Boto3)</a>	<a href="#">AWS SDK for Python (Boto3) コード例</a>
<a href="#">AWS SDK for Ruby</a>	<a href="#">AWS SDK for Ruby コード例</a>
<a href="#">AWS SDK for Rust</a>	<a href="#">AWS SDK for Rust コード例</a>
<a href="#">AWS SDK for SAP ABAP</a>	<a href="#">AWS SDK for SAP ABAP コード例</a>
<a href="#">AWS SDK for Swift</a>	<a href="#">AWS SDK for Swift コード例</a>

### 可用性の例

必要なものが見つからなかった場合。このページの下側にある [Provide feedback (フィードバックを送信)] リンクから、コードの例をリクエストしてください。

# Amazon Forecast の仕組み

Amazon Forecast で予測プロジェクトを作成するときは、次のリソースを使用します。

- [データセットのインポート](#) – [Datasets] (データセット) は、入力データのコレクションです。データセットグループは、補足情報を含むデータセットのコレクションです。Forecast アルゴリズムはデータセットグループを使用して、予測子と呼ばれるカスタムの予測モデルをトレーニングします。
- [予測子のトレーニング](#) – [Predictors] (予測子) は、データに基づいてトレーニングされたカスタムモデルです。予測子をトレーニングするには、事前に構築されたアルゴリズムを選択するか、Amazon Forecast で最適なアルゴリズムが選択されるように AutoML オプションを選択します。
- [予測の生成](#) – 時系列データの予測を生成したり、[QueryForecast](#) API を使用して予測したり、コンソールで視覚化したりできます。

# セットアップ

Amazon Forecast を使用して時系列データを評価または予測する前に、AWS アカウントを作成して、アクセス権限を設定し、AWS Command Line Interface (AWS CLI) をセットアップします。

トピック

- [AWS へのサインアップ](#)
- [AWS CLI のセットアップ](#)
- [Amazon Forecast のアクセス許可を設定する](#)

## AWS へのサインアップ

Amazon Web Services (AWS) にサインアップすると、Amazon Forecast を含む AWS のすべてのサービスに、AWS アカウントが自動的にサインアップされます。料金は、使用するサービスの料金のみが請求されます。

## AWS CLI のセットアップ

AWS Command Line Interface (AWS CLI) は、Amazon Forecast を含む AWS のサービスを管理するための統合デベロッパーツールです。このツールをインストールして使用することをお勧めします。

1. AWS CLI をインストールするには、「AWS Command Line Interface ユーザーガイド」の「[AWS Command Line Interface のインストール](#)」の手順に従ってください。
2. AWS CLI を設定し、呼び出すプロファイルを設定するには、「AWS Command Line Interface ユーザーガイド」の「[Configuring the AWS CLI](#)」の手順に従います。
3. AWS CLI プロファイルが適切に設定されていることを確認するには、コマンドウィンドウで次のコマンドを実行します。

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

プロファイルが正しく設定されている場合は、次のような出力が表示されます。

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

4. AWS CLI が Amazon Forecast 用に設定されていることを確認するには、以下のコマンドを実行します。

```
aws forecast help
```

```
aws forecastquery help
```

AWS CLI が正しく設定されていれば、Amazon Forecast または Amazon Forecast クエリでサポートされている AWS CLI コマンドのリストが表示されます。

## Amazon Forecast のアクセス許可を設定する

Amazon Forecast は、Amazon Simple Storage Service (Amazon S3) を使用してターゲットの時系列データを保存します。このデータは、予測を生成できる予測子をトレーニングするために使用されます。Amazon Forecast がお客様に代わって Amazon S3 にアクセスするには、許可を設定する必要があります。

お客様の代わりに Amazon S3 を使用するための許可を Amazon Forecast に付与するには、アカウントで AWS Identity and Access Management (IAM) ロールと IAM ポリシーを設定する必要があります。IAM ポリシーは必要な許可を指定します。また、IAM ロールにアタッチする必要があります。

IAM ロールとポリシーを作成し、ロールにポリシーをアタッチするには、IAM コンソールまたは AWS Command Line Interface (AWS CLI) を使用できます。

### Note

Forecast は Amazon Virtual Private Cloud と通信せず、Amazon S3 の VPCE ゲートウェイをサポートできません。VPC アクセスのみを許可する S3 バケットを使用すると、AccessDenied エラーが発生します。

### トピック

- [Amazon Forecast の IAM ロールを作成する \(IAM コンソール\)](#)
- [Amazon Forecast の IAM ロールを作成する \(AWS CLI\)](#)
- [サービス間の混乱した代理の防止](#)

## Amazon Forecast の IAM ロールを作成する (IAM コンソール)

AWS IAM コンソールを使用して以下を実行できます。

- 信頼できるエンティティとして Amazon Forecast を使用して IAM ロールを作成する
- Amazon Forecast に Amazon S3 バケットのデータの表示、読み取り、書き込みを許可する許可を使用して IAM ポリシーを作成する
- IAM ポリシーを IAM ロールにアタッチします。

Amazon S3 へのアクセスを Amazon Forecast に許可する IAM ロールとポリシーを作成するには (IAM コンソール)

1. IAM コンソール (<https://console.aws.amazon.com/iam>) にサインインします。
2. [ポリシー] を選択して次の手順を実行し、必要なポリシーを作成します。
  - a. [Create policy] (ポリシーの作成) をクリックします。
  - b. [ポリシーの作成] ページのポリシーエディタで [JSON] タブを選択します。
  - c. 次のポリシーをコピーし、エディタ内に貼り付けてテキストを置き換えます。*bucket-name* を必ず S3 バケットの名前に置き換えてから、[ポリシーの確認] を選択します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:Get*",
        "s3:List*",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}
```

[次へ: タグ] をクリックします



- d. オプションで、このポリシーにタグを割り当てることができます。[Next: Review] (次: 確認) をクリックします。
  - e. [ポリシーの確認] ページで [名前] にポリシーの名前を入力します。例えば、AWS3BucketAccess です。必要に応じて、このポリシーの説明を入力してから、[ポリシーの作成] を選択します。
3. ナビゲーションペインで、[ロール] を選択します。次の手順に従って IAM ロールを作成します。
- a. ロールの作成 を選択します。
  - b. 信頼できるエンティティタイプ で、AWS のサービス を選択します。

[ユースケース] では、[一般的なユースケース] セクションまたは [その他の AWS のサービスのユースケース] ドロップダウンリストから [Forecast] を選択します。[Forecast] が見つからない場合は、[EC2] を選択してください。

[次へ] をクリックします。

- c. [権限の追加] セクションで [次へ] をクリックします。
- d. [名前、レビュー、および作成] セクションで、[ロール名] にロールの名前 (ForecastRole など) を入力します。[ロールの説明] で、ロールの説明を更新し、[ロールの作成] を選択します。
- e. これで [ロール] ページに戻るはずですが、新しいロールを選択して、ロールの詳細ページを開きます。
- f. [概要] で、[ロール ARN] の値をコピーして保存します。データセットを Amazon Forecast 内にインポートするときに必要になります。
- g. このロールを使用するサービスとして [Amazon Forecast] を選択しなかった場合は、[信頼関係] を選択し、[信頼関係の編集] を選択して、信頼ポリシーを次のように更新します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "forecast.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
```

```
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "account-id"
      },
      "ArnLike": {
        "aws:SourceArn": "arn:aws:forecast:region:account-id:*"
      }
    }
  }
]
```

- h. [オプション] KMS キーを使用して暗号化を有効にする場合は、KMS キーと ARN をアタッチしてください。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ForecastKMS",
      "Effect": "Allow",
      "Action": "kms:*",
      "Resource": "arn:aws:kms:region:account-id:key/KMS-key-id"
    }
  ]
}
```

## Amazon Forecast の IAM ロールを作成する (AWS CLI)

AWS CLI は、以下のことを実行するために使用できます。

- 信頼できるエンティティとして Amazon Forecast を使用して IAM ロールを作成する
- Amazon Forecast に Amazon S3 バケットのデータの表示、読み取り、書き込みを許可する許可を使用して IAM ポリシーを作成する
- IAM ポリシーを IAM ロールにアタッチします。

## Amazon S3 へのアクセスを Amazon Forecast に許可する IAM ロールとポリシーを作成するには (AWS CLI)

1. 信頼されたエンティティとして Amazon Forecast を使用して、お客様の代わりとなる IAM ロールを作成します。

```
aws iam create-role \  
--role-name ForecastRole \  
--assume-role-policy-document '{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "",  
      "Effect": "Allow",  
      "Principal": {  
        "Service": "forecast.amazonaws.com"  
      },  
      "Action": "sts:AssumeRole",  
      "Condition": {  
        "StringEquals": {  
          "aws:SourceAccount": "account-id"  
        },  
        "ArnLike": {  
          "aws:SourceArn": "arn:aws:forecast:region:account-id:"  
        }  
      }  
    }  
  ]  
}'
```

このコマンドは、デフォルトの AWS 設定プロファイルが、Amazon Forecast でサポートされている AWS リージョン をターゲットとしていることを前提としています。Amazon Forecast でサポートされていない AWS リージョン をターゲットとするように別のプロファイル (例: `aws-forecast`) を設定した場合は、コマンドに `profile` パラメータを含めることによってその設定を明示的に指定する必要があります (例: `--profile aws-forecast`)。AWS CLI 設定プロファイルの設定の詳細については、AWS CLI の [configure](#) コマンドを参照してください。

コマンドが正常にロールを作成すると、それが次のような出力として返されます。

```
{  
  "Role": {
```

```

    "Path": "/",
    "RoleName": "ForecastRole",
    "RoleId": your-role-ID,
    "Arn": "arn:aws:iam::your-acct-ID:role/ForecastRole",
    "CreateDate": "creation-date",
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Sid": "",
          "Effect": "Allow",
          "Principal": {
            "Service": "forecast.amazonaws.com"
          },
          "Action": "sts:AssumeRole",
          "Condition": {
            "StringEquals": {
              "aws:SourceAccount": "your-acct-ID"
            },
            "ArnLike": {
              "aws:SourceArn": "arn:aws:forecast:region:your-acct-ID:"
            }
          }
        }
      ]
    }
  }
}

```

ロールの ARN をメモします。これは、Amazon Forecast 予測子をトレーニングするためのデータセットをインポートする際に必要です。

2. Amazon S3 のデータの一覧表示、読み取り、書き込みを行うための許可を指定した IAM ポリシーを作成し、ステップ 1 で作成した IAM ロールに付与します。

```

aws iam put-role-policy \
  --role-name ForecastRole \
  --policy-name ForecastBucketAccessPolicy \
  --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
      {

```

```

    "Effect": "Allow",
    "Action": [
      "s3:Get*",
      "s3:List*",
      "s3:PutObject"
    ],
    "Resource": [
      "arn:aws:s3:::bucket-name",
      "arn:aws:s3:::bucket-name/*"
    ]
  }
]
}'

```

3. [オプション] KMS キーを使用して暗号化を有効にする場合は、KMS キーと ARN をアタッチしてください。

```

aws iam put-role-policy \
  --role-name ForecastRole \
  --policy-name ForecastBucketAccessPolicy \
  --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "s3:Get*",
          "s3:List*",
          "s3:PutObject"
        ],
        "Resource": [
          "arn:aws:s3:::bucket-name",
          "arn:aws:s3:::bucket-name/*"
        ]
      }
    ]
  }'
aws iam put-role-policy \
  --role-name ForecastRole \
  --policy-name ForecastKMSAccessPolicy \
  --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
      {

```

```
    "Effect": "Allow",
    "Action": [
      "kms:DescribeKey",
      "kms:CreateGrant",
      "kms:RetireGrant"
    ],
    "Resource": [
      "arn:aws:kms:region:account-id:key/KMS-key-id"
    ]
  }
]
```

## サービス間の混乱した代理の防止

混乱した代理問題とは、あるアクションを実行する許可を持たないエンティティが、より多くの特権を持つエンティティにアクションの実行を強制できることで生じるセキュリティ上の問題です。AWS では、サービス間でのなりすましが、混乱した代理問題を生じさせる可能性があります。サービス間でのなりすましは、あるサービス (呼び出し元サービス) が、別のサービス (呼び出し対象サービス) を呼び出すときに発生する可能性があります。呼び出し元サービスが操作され、それ自身のアクセス許可が使用されて、本来アクセス許可が付与されるべきではない方法により、別の顧客のリソースに影響を与える可能性があります。これを防ぐために、AWS には、アカウント内のリソースへのアクセス権が付与されたサービスプリンシパルですべてのサービスのデータを保護するために役立つツールが用意されています。

リソースポリシーの `aws:SourceArn` および `aws:SourceAccount` のグローバル条件コンテキストキーを使用して、Identity and Access Management (IAM) が Amazon Forecast に付与する、リソースへのアクセス許可を制限することをお勧めします。両方のグローバル条件コンテキストキーを同じポリシーステートメントで使用する場合は、`aws:SourceAccount` 値、および `aws:SourceArn` 値の中のアカウントで、同じアカウント ID を使用する必要があります。

# 使用開始

Amazon Forecast の使用を開始するには、次の手順に従います。

- Forecast データセットを作成してトレーニングデータをインポートします。
- Forecast 予測子を作成して、時系列データに基づいて予想を生成できます。Forecast は、データセット内の各時系列にアルゴリズムの最適な組み合わせを適用します。
- 予測を生成します。

この演習では、公開されている利用可能な電力使用量データセットの修正版を使用して予測子をトレーニングします。データセットの削除の詳細については、「[ElectricityLoadDiagrams20112014 データセット](#)」を参照してください。以下はデータセットのサンプル行です。

```
2014-01-01 01:00:00, 2.53807106598985, client_0
2014-01-01 01:00:00, 23.648648648648624, client_1
2014-01-01 02:00:00, 9.648648648612345, client_0
```

この演習では、データセットを使用して予測子をトレーニングしてから、クライアントによる時間単位の電力使用量を予測します。

この演習では、Forecast コンソールまたは AWS Command Line Interface (AWS CLI) を使用できます。Amazon Forecast リソースはリージョン間で共有されないため、Amazon Forecast コンソール、AWS CLI、および Amazon Forecast SDK のデフォルトリージョンに注意してください。

## Important

始める前に、AWS アカウント を持っており、AWS CLI をインストールしたことを確認してください。詳細については、「[セットアップ](#)」を参照してください。[Amazon Forecast の仕組み](#) についても確認することをお勧めします。

## トピック

- [入力データを準備する](#)
- [開始方法 \(コンソール\)](#)
- [開始方法 \(AWS CLI\)](#)

- [開始方法 \(Python ノートブック\)](#)
- [リソースのクリーンアップ](#)

## 入力データを準備する

Amazon Forecast コンソールと AWS Command Line Interface (AWS CLI) のいずれかを使用して予想プロジェクトを設定するにかかわらず、入力データを設定する必要があります。データを準備するには、次の操作を実行します。

- トレーニングデータをコンピュータにダウンロードして、AWS アカウントの Amazon Simple Storage Service (Amazon S3) バケットにアップロードします。データを Amazon Forecast データセットにインポートするには、データを Amazon S3 バケットに保存する必要があります。
- AWS Identity and Access Management(IAM) ロールを作成します。Amazon Forecast に IAM ロールを使用して S3 バケットにアクセスする許可を付与します。IAM ロールの詳細については、IAM ユーザーガイドの[IAM ロール](#)を参照してください。

トレーニングデータを準備するには

1. zip ファイル、[electricityusagedata.zip](#) をダウンロードします。

この演習では、個々の家庭の電力消費量データセットの修正版を使用します。(Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository <http://archive.ics.uci.edu/ml>. Irvine, CA: University of California, School of Information and Computer Science.) 使用状況データは時間単位で集計されます。

2. コンテンツを解凍し、electricityusagedata.csv としてローカルに保存します。
3. S3 バケットにデータファイルをアップロードします。

ステップバイステップの手順については、Amazon Simple Storage Service ユーザーガイドの「[Uploading Files and Folders by Using Drag and Drop](#)」(ドラッグアンドドロップを使用したファイルとフォルダのアップロード)を参照してください。

4. IAM ロールを作成します。

「開始方法」の演習で AWS CLI を使用する場合は、IAM ロールを作成する必要があります。コンソールを使用している場合は、それを使ってロールを作成できます。手順については、[Amazon Forecast のアクセス許可を設定する](#)を参照してください。



Amazon S3 へのデータのアップロードが完了すると、Amazon Forecast コンソールまたは AWS CLI を使用して、トレーニングデータのインポート、予測子の作成、予想の生成や確認ができます。

- [開始方法 \(コンソール\)](#)
- [開始方法 \(AWS CLI\)](#)

## 開始方法 (コンソール)

この演習では、Amazon Forecast コンソールを使用して電力使用量の時系列データをインポートし、入力データセットに基づいて予測子を作成します。また予想期間に基づいて将来の電力使用量を予測します。

この演習では、個々の家庭の電力消費量データセットの修正版を使用します。(Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository <http://archive.ics.uci.edu/ml>. Irvine, CA: University of California, School of Information and Computer Science.) 使用状況データは時間単位で集計されます。変更されたデータは zip ファイル [electricityusagedata.zip](#) として入手できます。

### 前提条件

- AWS アカウント。AWS アカウントをまだお持ちでない場合は、「[AWS へのサインアップ](#)」の説明に従ってアカウントを作成します。
- Amazon Simple Storage Service (Amazon S3) バケットのトレーニングデータ。詳細については、「[入力データを準備する](#)」を参照してください。
- Amazon Forecast による S3 バケットの読み書きを許可する AWS Identity and Access Management (IAM) ロール。詳細については、「[Amazon Forecast の IAM ロールを作成する \(IAM コンソール\)](#)」を参照してください。

この演習には、完了するまでに数分から数時間かかる手順がいくつかあることに注意してください。

### ステップ 1: トレーニングデータをインポートする

時系列データを Amazon Forecast にインポートするには、データセットグループを作成し、データセットグループのドメインを選択します。データの詳細を指定して、Amazon Forecast にデータの S3 の場所を指定します。この例で使用されているターゲット時系列は、[過去の電力使用量](#)データです。

**Note**

この演習では、データセットグループを作成していないことを前提としています。以前にデータセットグループを作成した場合、表示される内容は、次のスクリーンショットや手順とは若干異なります。

時系列データをインポートして予測するには

1. Amazon Forecast コンソール (<https://console.aws.amazon.com/forecast/>) を開きます。
2. Amazon Forecast ホームページで、[Create dataset group] (データセットグループを作成) を選択します。
3. [Create dataset group (データセットグループの作成)] ページの [Dataset group details (データセットグループの詳細)] に、次の情報を入力します。
  - データセットグループ名 - データセットグループの名前を入力します。
  - [Forecasting domain] (ドメインの予測) - ドロップダウンメニューから [カスタム] を選択します。予想ドメインの選択方法の詳細については「[データセットドメインとタイプ](#)」を参照してください。

[タグ] セクションはそのままにしておきます。画面の表示は次のようになります。

# Create dataset group [Info](#)

## Dataset group details

### Dataset group name

The name can help you distinguish this dataset group from other dataset groups on the dataset groups dashboard.

The dataset group name must have 1 to 63 characters. Valid characters: a-z, A-Z, 0-9, and \_

### Forecasting domain [Info](#)

A forecasting domain defines a forecasting use case. You can choose a predefined domain, or you can create your own domain.

Choose this domain if none of the other domains are applicable to yo...

## ► Tags - optional [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.

Cancel

Next

4. [次へ] をクリックします。
5. [Create target time series dataset (ターゲット時系列データセットの作成)] ページの [Dataset details (データセットの詳細)] に、次の情報を入力します。
  - [Dataset name] (データセット名) - データセットグループの名前を入力します。
  - [Frequency of your data] (データの頻度) - デフォルト値の **1** のままにして、ドロップダウンメニューから [hour] (時間) を選択します。この設定は、入力時系列データの設定と一致している必要があります。サンプルの電気使用量データの時間間隔は 1 時間です。
  - データスキーマ — [スキーマビルダー] を選択し、時系列データの順序に合わせて列コンポーネントを上から下にドラッグします。

1. タイムスタンプ - デフォルトのタイムスタンプ形式である **yyyy-MM-dd HH:mm:ss** を使用します。
2. target\_value
3. item\_id

電力使用量入力データは、タイムスタンプ、指定時刻の電力使用量 (target\_value)、その電力使用量に対して課金されたお客様の ID (文字列) に列が対応しています。ここで指定する列の順序とタイムスタンプ形式は、入力時系列データと一致している必要があります。

[データセットの詳細] パネルは次のように表示されます。

### Dataset details

**Dataset name**  
The name can help you distinguish this dataset from other datasets on your Datasets dashboard.

The dataset name must have 1 to 63 characters. Valid characters: a-z, A-Z, 0-9, and \_

**Frequency of your data**  
This is the frequency at which entries are registered into your data file.

Your data entries have a time interval of

**Data schema** [Info](#)  
Use the data schema section to specify the attribute types for each column in your dataset. You can specify the schema in two ways:

**Schema builder**  
Specify your Attribute Name, Attribute Type, and attribute order in the text boxes provided.

**JSON schema**  
Specify AttributeName and AttributeType in the JSON format.

**Schema Builder** [Info](#)  
The attributes below are required for your chosen domain. You may add additional attributes. All attributes displayed must exist in your CSV file and must be ordered in the same order that they appear in your CSV file. To reorder the attributes, simply drag and drop each attribute to the correct position.

Column

1	<input type="text" value="timestamp"/>	<input type="text" value="timestamp"/>	<input type="text" value="yyyy-MM-dd HH:mm:ss"/>
2	<input type="text" value="target_value"/>	<input type="text" value="float"/>	
3	<input type="text" value="item_id"/>	<input type="text" value="string"/>	

You can add up to 10 more attributes.

6. [データセットのインポートの詳細]に、以下の情報を入力します。

- [データセットのインポート名] - データセットの名前を入力します。
- [タイムゾーンの選択] — デフォルトを選択したままにします (タイムゾーンは使用しないでください)。
- [Data location] (データの場所) - 以下の形式を使用して、Amazon S3 上の .csv ファイルの場所を入力してください。

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

- [IAM role] (IAM ロール) – デフォルトの [Enter a custom IAM role ARN] (カスタム IAM ロールの ARN を入力) のままにします。

または、ドロップダウンメニューから [Create a new role] (新しいロールを作成) を選択し、画面の指示に従って、Amazon Forecast に必要な IAM ロールを作成させることもできます。

- [Custom IAM role ARN] (カスタム IAM ロールの ARN) - [Amazon Forecast の IAM ロールを作成する \(IAM コンソール\)](#) で作成した IAM ロールの Amazon リソースネーム (ARN) を入力します。

[データセットインポートの詳細] パネルは次のように表示されます。

### Dataset import details

**Dataset import name**  
The name can help you distinguish this dataset import from other imports on your dataset detail page.

The dataset import name must have 1 to 63 characters. Valid characters: a-z, A-Z, 0-9, and \_

**Select time zone** [Info](#)  
Select a time zone for your dataset.

Do not use time zone ▼

**Data location** [Info](#)  
The location is the path to the file in your S3 bucket that contains your data.

View
Browse S3

Your files must be in CSV format.

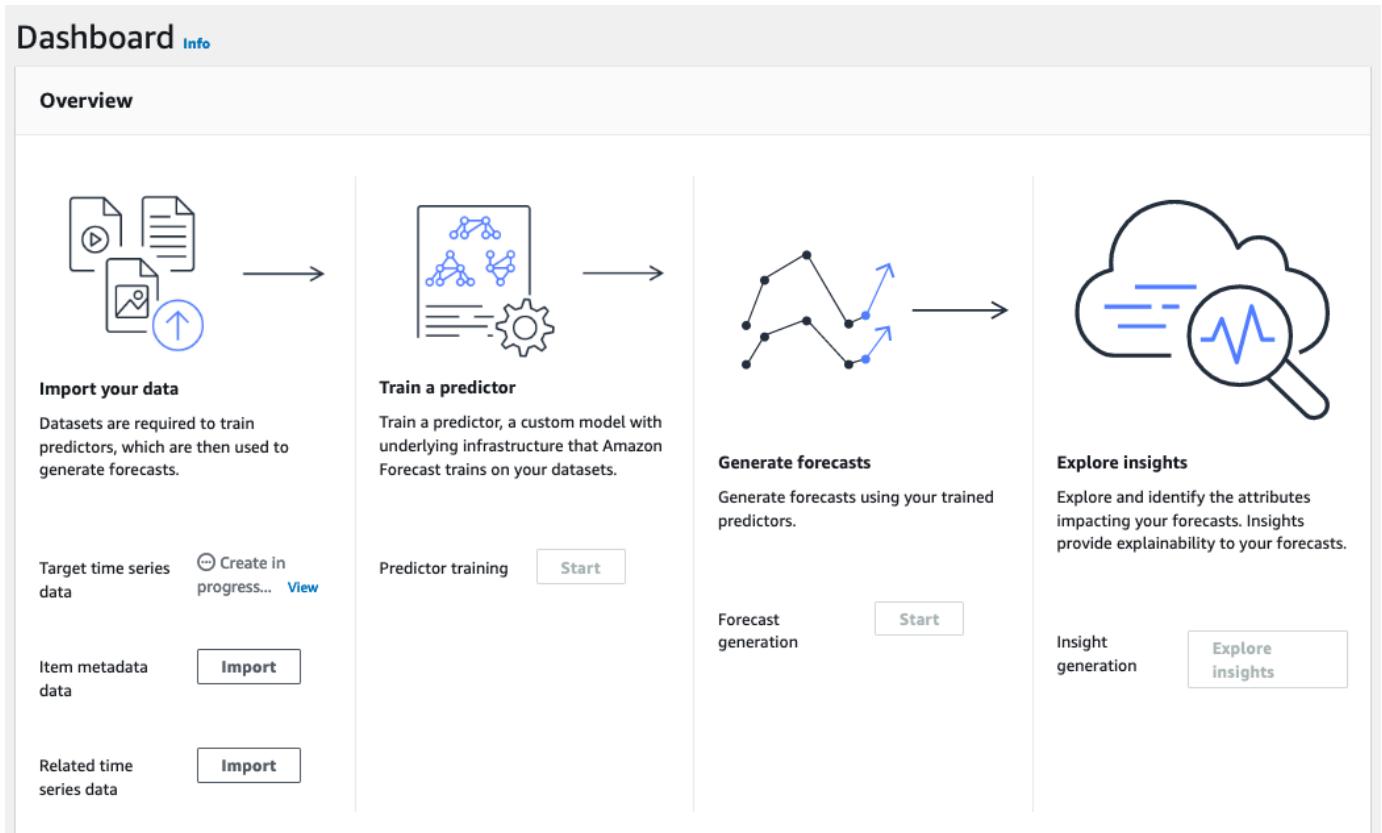
**IAM role** [Info](#)  
Dataset groups require permissions from IAM to read your dataset files in S3. Choose or create a role using this control.

Enter a custom IAM role ARN ▼

**Custom IAM role ARN**

Cancel
Previous
Start

7. [開始] を選択します。Amazon Forecast ホームページに戻された場合、[データセットグループを表示] を選択します。
8. 作成したデータセットグループの名前をクリックします。データセットグループの [ダッシュボード] ページが表示されます。画面の表示は次のようになります。



[ターゲット時系列データ] に、インポートジョブのステータスが表示されます。Amazon Forecast で、時系列データのインポートが終了するまで待ちます。このプロセスには、数分以上かかることがあります。データセットがインポートされると、ステータスが [アクティブ] に変わり、ダッシュボード上部のバナーにデータが正常にインポートされたことが通知されます。

ターゲット時系列データセットがインポートされました。これで、予測子を作成できます。

## ステップ 2: 予測子を作成する

次に、予測子を作成して、時系列データに基づいて予想を生成するために使用します。Forecast は、データセット内の各時系列にアルゴリズムの最適な組み合わせを適用します。

Forecast コンソールで予測子を作成するには、予測子名、予想頻度を指定し、予想期間を定義します。設定できる追加フィールドの詳細については、「[予測子のトレーニング](#)」を参照してください。

## 予測子を作成するには

1. ターゲット時系列データセットのインポートが完了すると、データセットグループの [ダッシュボード] は次のようになります。

The screenshot shows the Amazon Forecast dashboard with the following sections:

- Import your data:** Datasets are required to train predictors, which are then used to generate forecasts. It includes a 'Target time series data' section with a status of 'Active' and buttons for 'View' and 'Edit'. Below are 'Item metadata data' and 'Related time series data' sections, each with an 'Import' button.
- Train a predictor:** Train a predictor, a custom model with underlying infrastructure that Amazon Forecast trains on your datasets. It features a 'Predictor training' section with a prominent orange 'Start' button.
- Generate forecasts:** Generate forecasts using your trained predictors. It includes a 'Forecast generation' section with a 'Start' button.
- Explore insights:** Explore and identify the attributes impacting your forecasts. Insights provide explainability to your forecasts. It includes an 'Insight generation' section with an 'Explore insights' button.

[予測子をトレーニングする] で、[Start (開始)] を選択します。[Train predictor (予測子をトレーニングする)] ページが表示されます。

### Note

予測子をトレーニングする前に、[Target time series data] (ターゲット時系列データ) の Status が Active である必要があります。これは、インポートが正常に終了したことを意味します。

2. [予測子をトレーニングする] ページで、[予測子の設定] に次の情報を入力します。
  - [Predictor name] (予測子名) - 予測子の名前を入力します。
  - [Forecast frequency] (予測頻度) - デフォルト値の **1** のままにします。ドロップダウンメニューから [hour] を選択します。この設定は、入力時系列データの設定と一致している必要があります。サンプルの電気使用量データの時間間隔は 1 時間です。

- [Forecast horizon] (予測期間) - 今後予測を実行する期間を選択します。この数値に、「Step 1: Import the Training Data」で指定したデータ入力頻度 (hourly) を乗算して、今後予測を実行する期間が決定されます。この演習では、この数値を 36 に設定し、36 時間の予測を行います。
- [予想ディメンション] と [予想分位数] — これらのフィールドはデフォルト値のままにします。

残りの [入力データ設定] セクションと [タグ] セクションはオプションなので、デフォルト値のままにしておきます。[予測子設定] セクションは次のように表示されます。



### Predictor settings

**Predictor name**  
The name can help you distinguish this predictor from your other predictors.

The predictor name must have 1 to 63 characters. Valid characters: a-z, A-Z, 0-9, and \_

---

**Forecast configuration**

**Forecast frequency**  
This is the frequency at which your forecasts are generated.

Your forecast frequency is

**Forecast horizon [Info](#)**  
This number tells Amazon Forecast how far into the future to predict your data at the specified forecast frequency.

**Forecast dimensions - *optional***  
Item id is used in training by default. Select additional keys you would like to use to generate a forecast. These keys are fields in your dataset.

**Forecast quantiles - *optional* [Info](#)**  
Specify the quantiles used to create forecasts and evaluate predictors. Choose up to 5 quantiles between 0.01 and 0.99 (by increments of 0.01). You can also include the mean forecast with 'mean'.

Forecast quantiles	Value	
<input type="text" value="Forecast quantile 1"/>	<input type="text" value="0.10"/>	<input type="button" value="Remove"/>
<input type="text" value="Forecast quantile 2"/>	<input type="text" value="0.50"/>	<input type="button" value="Remove"/>
<input type="text" value="Forecast quantile 3"/>	<input type="text" value="0.90"/>	<input type="button" value="Remove"/>

You can add up to 2 more forecast quantiles.

---

### Predictor settings

**Optimization metric - *optional* [Info](#)**  
Use a specific accuracy metric to optimize your predictor.

**Enable explainability [Info](#)**

Enable explainability

3. [作成] を選択します。データセットグループの [ダッシュボード] ページが表示されます。画面の表示は次のようになります。

## Dashboard Info

### Overview

**Import your data**

Datasets are required to train predictors, which are then used to generate forecasts.

Target time series data ✔ Active [View](#) [Edit](#)

Item metadata data [Import](#)

Related time series data [Import](#)

**Train a predictor**

Train a predictor, a custom model with underlying infrastructure that Amazon Forecast trains on your datasets.

[View predictors](#)

[Train predictor](#)

**Generate forecasts**

Generate forecasts using your trained predictors.

Forecast generation [Start](#)

**Explore insights**

Explore and identify the attributes impacting your forecasts. Insights provide explainability to your forecasts.

Insight generation [Explore insights](#)

- 予測子のステータスを確認するには、[予測子を表示] を選択します。
- [予測子] ページの [トレーニングステータス] 列で予測子のステータスを確認します。画面の表示は次のようになります。

### Predictors (1) Info

This page gives you an overview of your predictors, and provides information about their status and performance.

[Manage notifications](#) [Stop](#) [Retrain](#) [Delete](#) [Create new forecast](#) [Train new predictor](#)

Find predictor name < 1 >

Predictor name	Training status	Forecast types	WAPE	RMSE	AutoPredictor Info	Date created
<a href="#">gs_predictor</a>	⊖ Create in progress... 2 hr 3 mins est. remaining	-	-	-	True	Fri, 25 Feb 2022 23:33:46 GMT

Amazon Forecast で予測子のトレーニングが終了するまで待ちます。このプロセスには、数分以上かかることがあります。予測子がトレーニングされると、ステータスが [アクティブ] に変わり、予想の生成を開始できることを知らせるバナーが表示されます。

## ステップ 3: 予測を作成する

予測子がアクティブになると、予想を作成できます。予測は予測のグループであり、ターゲットデータセット内のすべての項目に対して 1 つです。完全な予測を取得するには、エクスポートジョブを作成します。

予測を取得および表示するには

1. データセットグループの [ダッシュボード] の [予想の生成] で、[開始] を選択します。[Create a forecast (予測の作成)] ページが表示されます。

### Note

予測を生成する前に、[Predictor training] (予測子のトレーニング) の Status が Active である必要があります。

2. [Create a forecast (予測の作成)] ページの [Forecast details (予測の詳細)] に次の情報を入力します。
  - [Forecast name] (予測名) - 予測の名前を入力します。
  - [Predictor] (予測子) - ドロップダウンメニューから、 で作成した予測子を選択します。Step 2: Train a Predictor

[予想の分位数] と [タグ] フィールドはオプションなので、デフォルト値のままにしておきます。画面の表示は次のようになります。

## Forecast details

### Forecast name

The name can help you distinguish this forecast from your other forecasts.

The forecast name must have 1 to 63 characters. Valid characters: a-z, A-Z, 0-9, and \_

### Predictor Info

The predictor that you want to use to create forecasts.

### Forecast types - optional Info

Enter up to 5 quantile values between .01 to .99. You can also enter 'mean'. By default, Amazon Forecast will generate forecasts for .10, .50 and .90 quantiles.

Separate forecast types with commas.

[開始] をクリックします。

3. [予想] ページが表示されます。画面の表示は次のようになります。

Forecast name	Status	Forecast created	Predictor used
gs_forecast	Create in progress... 35 mins est. remaining	Sat, 26 Feb 2022 00:02:01 GMT	gs_predictor_123

[ステータス] 列には、予想のステータスが一覧表示されます。Amazon Forecast で、予測の作成が終了するまで待ちます。このプロセスには、数分以上かかることがあります。予想が作成されると、ステータスは [アクティブ] に変わります。

予想が作成されたので、予想をエクスポートできます。

## ステップ 4: 予想をエクスポートする

予想が作成されたら、完全に予想をエクスポートできます。

## 完全な予測をエクスポートするには

1. [データセットグループ] のページで、Step 1: Import Training Data で作成したデータセットグループを選択します。
2. 画面の左上隅にある  
☰  
をクリックして、ナビゲーションペインを開きます。データセットグループの下にある [予想] をクリックします。
3. Step 3: Create a Forecast で作成した予測の横にあるラジオボタンを選択します。
4. [Create forecast export (予測エクスポートの作成)] を選択します。[Create forecast export (予測エクスポートの作成)] ページが表示されます。
5. [Create forecast export (予測エクスポートの作成)] ページの [エクスポートの詳細] に次の情報を入力します。
  - [Export name] (エクスポート名) - 予測エクスポートジョブの名前を入力します。
  - [IAM role] (IAM ロール) – デフォルトの [Enter a custom IAM role ARN] (カスタム IAM ロールの ARN を入力) のままにします。  
  
または、ドロップダウンメニューから [Create a new role] (新しいロールを作成) を選択し、画面の指示に従って、Amazon Forecast に必要な IAM ロールを作成させることもできます。
  - [Custom IAM role ARN] (カスタム IAM ロールの ARN) - [Amazon Forecast の IAM ロールを作成する \(IAM コンソール\)](#) で作成した IAM ロールの Amazon リソースネーム (ARN) を入力します。
  - [S3 予測エクスポートの場所] - 次の形式を使用して、Amazon Simple Storage Service (Amazon S3) バケットまたはバケットのフォルダの場所を入力します。

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

画面の表示は次のようになります。

## Create forecast export [Info](#)

### Export details

#### Export name

The name can help you distinguish this export job from your other exports.

The export name must have 1 to 63 characters. Valid characters: a-z, A-Z, 0-9, and \_

#### IAM role [Info](#)

Amazon forecast requires permissions to store the exported forecasts in S3. Choose or create a role that has permissions to write to S3. If you created an IAM role when you imported a dataset and specified it in the Any S3 bucket field, choose that IAM role.

#### Custom IAM role ARN

#### KMS key ARN - *optional*

The ARN of the IAM role that Amazon Forecast uses to access the AWS KMS key.

The KMS key must have 1 to 256 characters. Valid characters: a-z, A-Z, 0-9, -, ., /, and :

#### Export file type - *optional*

Files will be exported to CSV by default. If you wish to export to Parquet, choose Parquet below.

 CSV PARQUET

#### S3 forecast export location [Info](#)

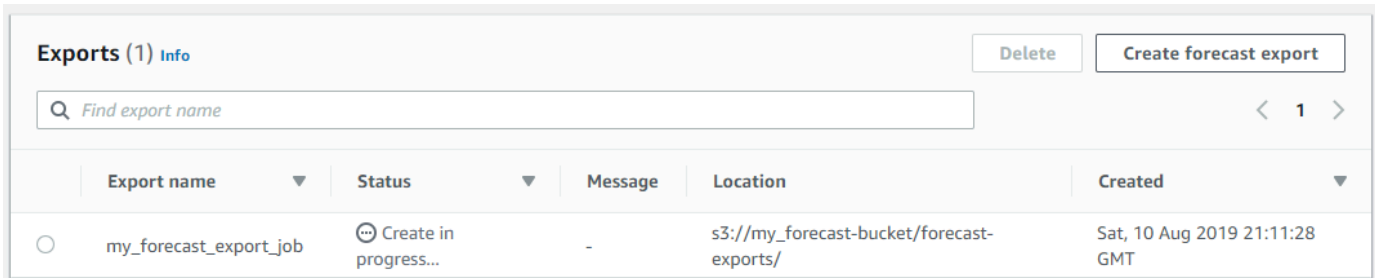
This is the path to the S3 bucket or folder in the bucket where you want to store your exported forecasts.

Your forecast export will be one or more CSV files.

### ► Tags - *optional* [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.

6. [開始] をクリックします。[予想] ページが表示されます。
7. Step 3: Create a Forecast で作成した予想をクリックします。[エクスポート] セクションを見つけてください。画面の表示は次のようになります。



Export name	Status	Message	Location	Created
my_forecast_export_job	Create in progress...	-	s3://my_forecast-bucket/forecast-exports/	Sat, 10 Aug 2019 21:11:28 GMT

ステータスの進行状況が表示されます。Amazon Forecast で、予測のエクスポートが終了するまで待ちます。このプロセスには、数分以上かかることがあります。予測がエクスポートされたら、ステータスが [Active] に変わり、S3 バケットで予測ファイルを見つけることができます。

## 開始方法 (AWS CLI)

この演習では、AWS Command Line Interface (AWS CLI) を使用して Amazon Forecast を詳しく学びます。Amazon Forecast データセットを作成し、予測子をトレーニングして、結果の予測子を使用して予想を生成します。開始する前に、AWS アカウントがあり、AWS CLI が設定されていることを確認してください。詳細については、「[セットアップ](#)」を参照してください。

### Note

この演習の AWS CLI コマンドは Linux でテストされています。Windows での AWS CLI コマンドの使用については、「AWS Command Line Interface ユーザーガイド」の「[AWS Command Line Interface のパラメータ値の指定](#)」を参照してください。

## ステップ 1: トレーニングデータをインポートする

まずデータセットを作成し、そこに電力使用量データをインポートすることから始めます。

Amazon Forecast データセットを作成するには

1. 適切なドメインとデータセットのタイプを決定します。

データセットにインポートするトレーニングデータは、データセットのドメインとタイプの選択に影響します。それでは、電力使用量データのいくつかのサンプル行を確認しましょう。

```
2014-01-01 01:00:00, 2.53807106598985, client_0
2014-01-01 01:00:00, 23.648648648648624, client_1
```

```
2014-01-01 02:00:00, 9.648648648612345, client_0
```

データ形式は CSV (カンマ区切り値) で、(タイムスタンプで示されるように) 1 時間ごとに収集されます。これらの列が含まれています。

- 列 1 – 電力使用量がいつ記録されたかを示すタイムスタンプ。
- 列 2 – 時間ごとの電力使用量の値 (タイムスタンプ値が 1 時間ごとに増加する様子に注意してください)。
- 列 3 – 電気を使用しているお客様を識別するクライアント ID の値。

このデータに対して、次の事前定義されているデータセットドメインとデータセットタイプを選択します。

- カスタムドメイン – METRICS、RETAIL、または WEB\_TRAFFIC などのデータセットドメインはこのデータには適用されないため、カスタムドメインを選択します。
- ターゲット時系列タイプ – 時間の経過とともに、電力使用量を追跡するため、データは時系列です。また、予測するターゲット (列 2、電力使用量) も含まれています。したがって、ターゲット時系列データセットタイプを選択します。

このタイプを選択する理由を理解するには、「[事前定義済みのデータセットドメインとデータセットタイプ](#)」を参照してください。

## 2. データセットのスキーマを決定します。

[CUSTOM ドメイン](#) のターゲット時系列タイプには、timestamp、target\_value、および item\_id などのフィールドが必要です。target\_value フィールドはターゲットです。Amazon Forecast は、このフィールドの予測を生成します。

必須フィールドをデータ内の列にマップするには、スキーマを作成します。スキーマの各属性が、データ内のフィールドにマッピングされます。

### Important

スキーマ内の属性の順序は、トレーニングデータ内のフィールドの順序と一致している必要があります。

```
{
```



```
"Attributes":[
  {
    "AttributeName": "timestamp",
    "AttributeType": "timestamp"
  },
  {
    "AttributeName": "target_value",
    "AttributeType": "float"
  },
  {
    "AttributeName": "item_id",
    "AttributeType": "string"
  }
]
```

これで、データセットを作成してデータをインポートするために必要な情報が得られました。

### 3. データセットを作成します。

```
aws forecast create-dataset \
--dataset-name electricity_demand_ds \
--domain CUSTOM \
--dataset-type TARGET_TIME_SERIES \
--data-frequency H \
--schema '{
  "Attributes": [
    {
      "AttributeName": "timestamp",
      "AttributeType": "timestamp"
    },
    {
      "AttributeName": "target_value",
      "AttributeType": "float"
    },
    {
      "AttributeName": "item_id",
      "AttributeType": "string"
    }
  ]
}'
```

このリクエストでは、data-frequency 値 H は 1 時間ごとのデータ収集頻度を表します。以下に、応答の例を示します。

```
{
  "DatasetArn": "arn:aws:forecast:us-west-2:acct-id:dataset/
electricity_demand_ds"
}
```

このオペレーションの詳細については、「[CreateDataset](#)」を参照してください。

#### 4. (オプション) データセットの説明を取得します。

```
aws forecast describe-dataset \
--dataset-arn arn:aws:forecast:us-west-2:acct-id:dataset/electricity_demand_ds
```

以下に、応答の例を示します。

```
{
  "DatasetName": "electricity_demand_ds",
  "DatasetArn": "arn:aws:forecast:us-west-2:acct-id:dataset/
electricity_demand_ds",
  "CreationTime": 1564533087.907,
  "LastModificationTime": 1564533087.907,
  "Domain": "CUSTOM",
  "DatasetType": "TARGET_TIME_SERIES",
  "DataFrequency": "H",
  "Schema": { ... },
  "EncryptionConfig": {},
  "Status": "ACTIVE"
}
```

#### Note

レスポンスのキーと値のペアの順序は任意です。

#### 5. データセットグループを作成し、データセットを追加します。domain パラメータの値は、データセットの domain と一致する必要があります。

```
aws forecast create-dataset-group \
--dataset-group-name electricity_ds_group \
```

```
--dataset-arns arn:aws:forecast:us-west-2:acct-id:dataset/electricity_demand_ds \  
--domain CUSTOM
```

以下に、応答の例を示します。

```
{  
  "DatasetGroupArn": "arn:aws:forecast:us-west-2:acct-id:dataset-group/  
electricity_ds_group"  
}
```

このオペレーションの詳細については、「[CreateDatasetGroup](#)」を参照してください。

## 6. (オプション) データセットグループの説明を取得します。

```
aws forecast describe-dataset-group \  
--dataset-group-arn arn:aws:forecast:us-west-2:acct-id:dataset-group/  
electricity_ds_group
```

以下に、応答の例を示します。

```
{  
  "DatasetGroupName": "electricity_ds_group",  
  "DatasetGroupArn": "arn:aws:forecast:us-west-2:acct-id:dataset-group/  
electricity_ds_group",  
  "DatasetArns": [  
    "arn:aws:forecast:us-west-2:acct-id:dataset-group/electricity_ds_group"  
  ],  
  "Domain": "CUSTOM",  
  "CreationTime": 1564533719.852,  
  "LastModificationTime": 1564533719.852,  
  "Status": "ACTIVE"  
}
```

## 7. Amazon S3 バケットからデータセットに電力使用量トレーニングデータをインポートします。

指定した IAM ロールには、S3 バケットのデータを読み込むための許可が必要です。IAM ロールの作成方法については、「[Amazon Forecast の IAM ロールを作成する \(AWS CLI\)](#)」を参照してください。

```
aws forecast create-dataset-import-job \  
--dataset-arn arn:aws:forecast:us-west-2:acct-id:dataset/electricity_demand_ds \  
--dataset-import-job-name electricity_ds_import_job \  

```

```
--data-source '{
  "S3Config": {
    "Path": "s3://bucket/electricityusagedata.csv",
    "RoleArn": "arn:aws:iam::acct-id:role/Role"
  }
}'
```

以下は、data-source パラメータの短縮構文です。

```
--data-source S3Config="{Path='s3://bucket/
electricityusagedata.csv',RoleArn='arn:aws:iam::acct-id:role/Role'}"
```

以下に、応答の例を示します。

```
{
  "DatasetImportJobArn": "arn:aws:forecast:us-west-2:acct-id:dataset-import-job/
electricity_demand_ds/electricity_ds_import_job"
}
```

このオペレーションの詳細については、「[CreateDatasetImportJob](#)」を参照してください。

## 8. インポートのステータスを確認します。

```
aws forecast describe-dataset-import-job \
--dataset-import-job-arn arn:aws:forecast:us-west-2:acct-id:dataset-import-job/
electricity_demand_ds/electricity_ds_import_job
```

以下に、応答の例を示します。

```
{
  "DatasetImportJobName": "electricity_ds_import_job",
  "DatasetImportJobArn": "arn:aws:forecast:us-west-2:acct-id:dataset-import-job/
electricity_demand_ds/electricity_ds_import_job",
  "DatasetArn": "arn:aws:forecast:us-west-2:acct-id:dataset/
electricity_demand_ds",
  "DataSource": {
    "S3Config": {
      "Path": "s3://bucket/electricityusagedata.csv",
      "RoleArn": "arn:aws:iam::acct-id:role/ForecastRole"
    }
  },
}
```

```
"DataSize": 0.14639010466635227,  
"TimeStampFormat": "yyyy-MM-dd HH:mm:ss",  
"CreationTime": 1564537011.114,  
"LastModificationTime": 1564537028.223,  
"Status": "CREATE_IN_PROGRESS"  
}
```

データがすべてインポートされると、ステータスは ACTIVE に変わり、次の例に示すように、データの統計がレスポンスに含まれます。

```
{  
  "DatasetArn": "arn:aws:forecast:us-west-2:acct-id:dataset/  
electricity_demand_ds",  
  "Status": "ACTIVE",  
  "FieldStatistics": {  
    "date": {  
      "Min": "2014-01-01T01:00:00Z",  
      "Max": "2015-01-01T00:00:00Z",  
      "Count": 3241200,  
      "CountDistinct": 8760,  
      "CountNull": 0  
    },  
    "target": {  
      "Min": "0.0",  
      "Max": "168200.0",  
      "Avg": 606.5167610461679,  
      "Stddev": 3518.405223972031,  
      "Count": 3241200,  
      "CountDistinct": 1196961,  
      "CountNull": 0,  
      "CountNan": 0  
    },  
    "item": {  
      "Count": 3241200,  
      "CountDistinct": 370,  
      "CountNull": 0  
    }  
  },  
  ...  
}
```

**⚠ Important**

データセットグループで予測子を作成するには、ステータスが ACTIVE になるまで待つ必要があります。

このオペレーションの詳細については、「[DescribeDatasetImportJob](#)」を参照してください。

## ステップ 2: Predictor を作成する

予測子を作成するには、[CreateAutoPredictor](#) オペレーションを使用し、次の情報を入力します。

- 予測子名 — 他の予測子と区別できるように予測子に名前を付けます
- データセットグループ — 前のステップでデータセットグループを作成しました。
- 予測頻度 — 予測の粒度 (毎時、毎日、毎週など)。
- 予測期間 — 予測されるタイムステップの数。

予測子が作成されたら、Amazon Forecast によって生成された精度メトリクスを確認します。メトリクスは、予測の生成に予測子を使用するかどうかを決定するのに役立ちます。予測子の詳細については、「[予測子のトレーニング](#)」を参照してください。

予測子を作成して、精度メトリクスを確認するには

### 1. 予測子を作成します。

```
aws forecast create-predictor \  
--predictor-name electricitypredictor \  
--input-data-config DatasetGroupArn="arn:aws:forecast:us-west-2:acct-id:dsgroup/  
electricity_ds_group" \  
--forecast-horizon 36 \  
--forecast-frequency D
```

以下に、応答の例を示します。

```
{  
  "PredictorArn": "arn:aws:forecast:us-west-2:acct-id:predictor/  
electricitypredictor"
```

```
}
```

## 2. 予測子のステータスを取得します。

```
aws forecast describe-predictor \  
--predictor-arn arn:aws:forecast:us-west-2:acct-id:predictor/electricitypredictor
```

以下に、応答の例を示します。

```
{  
  "PredictorArn": "arn:aws:forecast:<region>:<acct-num>:predictor/  
electricitypredictor",  
  "PredictorName": "electricitypredictor",  
  "ForecastHorizon": 36,  
  "ForecastTypes": [  
    "0.1",  
    "0.5",  
    "0.9"  
  ],  
  "ForecastFrequency": "D",  
  "DatasetImportJobArns": [  
    "arn:aws:forecast:<region>:<acct-num>:dataset-import-job/  
getting_started_dataset/gs_import"  
  ],  
  "DataConfig": {  
    "DatasetGroupArn": "arn:aws:forecast:<region>:<acct-num>:dataset-group/  
getting_started",  
    "AttributeConfigs": [  
      {  
        "AttributeName": "target_value",  
        "Transformations": {  
          "aggregation": "sum",  
          "backfill": "zero",  
          "frontfill": "none",  
          "middlefill": "zero"  
        }  
      }  
    ]  
  },  
  "EstimatedTimeRemainingInMinutes": 97,  
  "Status": "CREATE_IN_PROGRESS",  
  "CreationTime": "2022-02-23T09:26:24.643000-08:00",  
  "LastModificationTime": "2022-02-23T09:49:26.899000-08:00",
```

```
"ExplainabilityInfo": {
  "Status": "NOT_AVAILABLE"
}
}
```

### ⚠ Important

モデルトレーニングには時間がかかります。トレーニングが完了し、予測子の状況が ACTIVE になるまで続行しないでください。

### 3. 予測子の精度メトリクスを取得します。

```
aws forecast get-accuracy-metrics \
--predictor-arn arn:aws:forecast:us-west-2:acct-id:predictor/electricitypredictor
```

以下に、応答の例を示します。

```
{
  "PredictorEvaluationResults": [
    {
      "TestWindows": [
        {
          "EvaluationType": "SUMMARY",
          "Metrics": {
            "RMSE": 448.19602551622864,
            "WeightedQuantileLosses": [
              {
                "Quantile": 0.9,
                "LossValue": 0.11574311406253326
              },
              {
                "Quantile": 0.5,
                "LossValue": 0.1706269067283527
              },
              {
                "Quantile": 0.1,
                "LossValue": 0.11724164222477837
              }
            ]
          }
        }
      ]
    }
  ],
}
```



```
{
  "EvaluationType": "COMPUTED",
  "Metrics": {
    "RMSE": 448.19602551622864,
    "WeightedQuantileLosses": [
      {
        "Quantile": 0.9,
        "LossValue": 0.11574311406253326
      },
      {
        "Quantile": 0.5,
        "LossValue": 0.1706269067283527
      },
      {
        "Quantile": 0.1,
        "LossValue": 0.11724164222477837
      }
    ]
  },
  "TestWindowEnd": 1420070400.0,
  "TestWindowStart": 1420002000.0
}
```

メトリクスは各分位数のエラー損失を示します。たとえば、最初の分位に 11.7% の誤差がありました。メトリクスには二乗平均平方根誤差 (RMSE) の値も表示されます。

要約メトリクスは、すべてのテストウィンドウで計算されたメトリクスの平均を示します。テストウィンドウは 1 つのみのため、要約と計算されたメトリクスは等しくなります。

このオペレーションの詳細については、「[GetAccuracyMetrics](#)」を参照してください。

### ステップ 3: 予測を作成する

Amazon Forecast は、データセット内の一意の `item_id` ごとに、`target_value` フィールド (データセットドメインとタイプによって決定される) の予測を作成します。この例では、`target_value` フィールドは電力使用量を提供し、`item_id` はクライアント ID を提供します。お客様ごとの時間ごとの電気使用量の予測を取得します。

予測が作成されたら、単一のアイテムをクエリするか、完全な予測をエクスポートできます。

予測を作成、取得、エクスポートするには

## 1. 予測を作成します。

```
aws forecast create-forecast \  
--forecast-name electricityforecast \  
--predictor-arn arn:aws:forecast:us-west-2:acct-id:predictor/electricitypredictor
```

このオペレーションでは、予測子を使用して予測を作成します。レスポンスで、予測の Amazon リソースネーム (ARN) を取得します。この ARN を使用して予測を取得およびエクスポートします。以下に、応答の例を示します。

```
{  
  "ForecastArn": "arn:aws:forecast:us-west-2:acct-id:forecast/  
electricityforecast"  
}
```

このオペレーションの詳細については、「[CreateForecast](#)」を参照してください。

## 2. client\_1 の予測の最初の 2 時間を取得します。

### Note

サービス名 forecastquery は、他で使用されているサービス名とは異なります。

```
aws forecastquery query-forecast \  
--forecast-arn arn:aws:forecast:us-west-2:acct-id:forecast/electricityforecast \  
--start-date 2015-01-01T00:00:00 \  
--end-date 2015-01-01T02:00:00 \  
--filters '{"item_id":"client_1"}'
```

このオペレーションには次のパラメータが含まれます。

- `start-date` および `end-date` – 予測を取得するためのオプションの日付範囲を指定します。これらのパラメータを指定しないと、このオペレーションは `client_1` の予測全体を返します。

- `filters - client_1` の電力予測を取得する `item_id` フィルターを指定します。

以下は、`filters` パラメータの短縮構文です。

```
--filters item_id="client_1"
```

以下に、応答の例を示します。

```
{
  "Forecast": {
    "Predictions": {
      "mean": [
        {
          "Timestamp": "2015-01-01T01:00:00",
          "Value": 20.952411651611328
        },
        {
          "Timestamp": "2015-01-01T02:00:00",
          "Value": 19.11078453063965
        }
      ],
      "p90": [
        {
          "Timestamp": "2015-01-01T01:00:00",
          "Value": 24.524038314819336
        },
        {
          "Timestamp": "2015-01-01T02:00:00",
          "Value": 22.319091796875
        }
      ],
      "p50": [
        {
          "Timestamp": "2015-01-01T01:00:00",
          "Value": 20.7841739654541
        },
        {
          "Timestamp": "2015-01-01T02:00:00",
          "Value": 19.237524032592773
        }
      ],
      "p10": [
```

```
    {
      "Timestamp": "2015-01-01T01:00:00",
      "Value": 18.507278442382812
    },
    {
      "Timestamp": "2015-01-01T02:00:00",
      "Value": 16.15062141418457
    }
  ]
}
}
```

これは 1 時間ごとの予測であるため、レスポンスには 1 時間ごとの予測値が表示されます。レスポンスで、以下の点に注意してください。

- mean – 特定の日時の平均値は、そのお客様の予測平均電力使用量です。
- p90、p50、p10 - 指定された日時に実際の値が記載された値を下回る信頼レベルを指定します。例えば、2015-01-01T01:00:00 の時点で、Amazon Forecast は、電気の使用量が 24.5 未満になると 90% 確信しています。Amazon Forecast は、使用量が 20.8 未満になることについて 50% の確信しており、使用量が 18.5 未満になることについて 10% 確信しています。

このオペレーションの詳細については、「[QueryForecast](#)」を参照してください。

3. 完全な予測を Amazon S3 バケットにエクスポートします。提供する IAM ロールには、データを S3 バケットに書き込むことを許可する許可が必要です。IAM ロールの作成方法については、「[Amazon Forecast の IAM ロールを作成する \(AWS CLI\)](#)」を参照してください。

予測エクスポートジョブを作成します。

```
aws forecast create-forecast-export-job \
--forecast-export-job-name electricityforecast_exportjob \
--forecast-arn arn:aws:forecast:us-west-2:acct-id:forecast/electricityforecast \
--destination S3Config="{Path='s3://bucket',RoleArn='arn:aws:iam::acct-id:role/Role'}"
```

以下に、応答の例を示します。

```
{
```

```
"ForecastExportJobArn": "arn:aws:forecast::us-west-2:acct-id:forecast-export/64bbc087"
}
```

このオペレーションの詳細については、「[CreateForecastExportJob](#)」を参照してください。

#### 4. エクスポートジョブのステータスを取得します。

```
aws forecast describe-forecast-export-job \
--forecast-export-job-arn arn:aws:forecast:us-west-2:acct-id:forecast/
electricityforecast
```

以下に、応答の例を示します。

```
{
  "ForecastExportJobArn": "arn:aws:forecast::us-west-2:acct-id:forecast-export/64bbc087",
  "ForecastExportJobName": "electricityforecast_exportjob",
  "Status": "CREATE_IN_PROGRESS"
}
```

ステータスが ACTIVE の場合は、指定した S3 バケットの予測ファイルを確認できます。

## 開始方法 (Python ノートブック)

### Note

Python ノートブックを使用したチュートリアルの詳細なリストについては、Amazon Forecast の「[Github Samples](#)」(Github サンプル) のページを参照してください。

Python ノートブックで Amazon Forecast API の使用を開始するには、「[開始方法のチュートリアル](#)」を参照してください。チュートリアルでは、Forecast のコアステップを最初から最後までガイドします。

特定のプロセスに関する基本的なチュートリアルについては、次の Python ノートブックを参照してください。

1. [データの準備](#) - データセットの準備、データセットグループの作成、スキーマの定義、およびデータセットグループのインポートを行います。
2. [予測子の構築](#) - Forecast データセットにインポートしたデータに基づいて予測子をトレーニングします。
3. [予測子の評価](#) - 予測を取得し、予測を視覚化し、結果を比較します。
4. [予測子の再トレーニング](#) - 更新されたデータを使用して既存の予測子を再トレーニングします。
5. [AutoPredictor へのアップグレード](#) - レガシー予測子を AutoPredictor にアップグレードします。
6. [クリーンアップ](#) - チュートリアル中に作成されたデータセットグループ、予測子、予測を削除します。

AutoML の開始方法のチュートリアルを繰り返すには、「[AutoML の開始方法](#)」を参照してください。

## 高度なチュートリアル

より高度なチュートリアルについては、次の Python ノートブックを参照してください。

- [アイテムレベルの Explainability](#) - データセット属性が特定の時系列および時間ポイントの予測にどのように影響するかを理解します。
- [複数のモデルの比較](#) - Prophet、ETS、および DeepAR+ を使用して予測子を作成し、結果を可視化してそれらのパフォーマンスを比較します。
- [コールドスタート予測](#) - アイテムメタデータと DeepAR+ アルゴリズムを使用して、コールドスタートシナリオを予測します (履歴データがほとんどないか、まったくない場合)。
- [関連する時系列データセットの組み込み](#) - 関連する時系列データセットを使用して、モデルの精度を向上させます。
- [アイテムメタデータの組み込み](#) - アイテムメタデータを使用して、モデルの精度を向上させます。
- [Weather Index の使用](#) - Weather Index を使用して、予測子をトレーニングするときに、履歴気象情報と予測された気象情報を組み込みます。
- [What-if 分析の実行](#) - ささまざまな価格シナリオを調査し、それが需要にどのように影響するかを評価します。
- [アイテムレベルの精度を評価する](#) - バックテストのメトリクスと予測をエクスポートし、予測子のアイテムレベルのパフォーマンスを評価します。

## リソースのクリーンアップ

不要な料金が発生しないようにするには、使用開始実習の終了後に作成したリソースを削除します。リソースを削除するには、Amazon Forecast コンソールまたは SDK または AWS Command Line Interface (AWS CLI) の Delete API を使用します。たとえば、[DeleteDataset](#) API を使用してデータセットを削除します。

リソースを削除するには、ステータスが ACTIVE、CREATE\_FAILED、または UPDATE\_FAILED である必要があります。ステータスを確認するには、Describe API (例: [DescribeDataset](#)) を使用します。

次の表に示すように、一部のリソースは他のものよりも前に削除する必要があります。このプロセスには時間がかかる場合があります。

アップロードしたトレーニングデータ ( `electricityusagedata.csv` ) を削除するには、「[S3 バケットからオブジェクトを削除する方法](#)」を参照してください。

削除するリソース	最初にこちらを削除します	メモ
ForecastExportJob		
Forecast		エクスポート中に予測を削除することはできません。予測を削除すると、予測のクエリを実行できなくなります。
Predictor	関連するすべての予測。	
DatasetImportJob		削除できません。
Dataset		データセットを対象にするすべての DatasetImportJob も削除されます。  予想子によって使用されている Dataset は削除できません。

削除するリソース	最初にこちらを削除します	メモ
DatasetSchema	スキーマを参照するすべてのデータセット。	
DatasetGroup	関連するすべての予想子 関連するすべての予測。  データセットグループのすべてのデータセット。	予想子で使用されている Dataset を含む Dataset Group は削除できません。



# チュートリアル

次のチュートリアルでは、Amazon Forecast で一般的なタスクを実行し、一般的なユースケース向けの既製のソリューションを提供する方法を示します。Python ノートブックを使用したチュートリアルの詳細なリストについては、Amazon Forecast の「[GitHub Samples](#)」(GitHub サンプル) のページを参照してください。

## チュートリアル

- [CloudFormation を使用した Forecast の自動化](#) - AWS CloudFormation スタックを使用して、データセットを S3 バケットに自動的にデプロイし、Forecast パイプラインをトリガーします。

## AWS CloudFormation による自動化

このチュートリアルでは、AWS CloudFormation オートメーションスタックを使用して Amazon Forecast パイプラインを起動し、デモンストレーションデータセットを使用して予測を生成します。

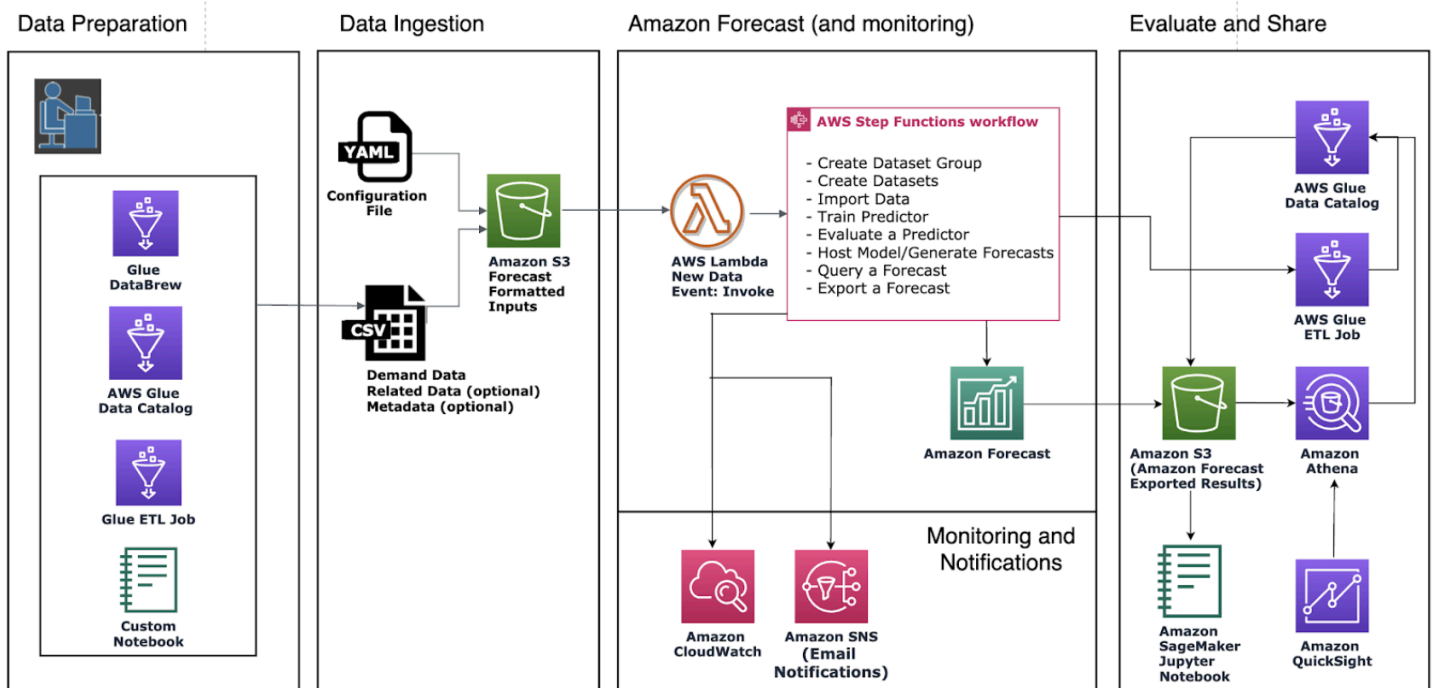
AWS Forecast AWS CloudFormation スタック:

- [Improving Forecast Accuracy with Machine Learning Solution](#) AWS CloudFormation テンプレートをデプロイします。
- [NYC Taxi Datasets](#) を Forecast Data Amazon S3 バケットにデプロイします。
- Forecast でデモの NYC タクシー予測パイプラインを自動的に開始します。

AWS CloudFormation テンプレートには、ターゲットの時系列、関連する時系列、およびアイテムメタデータのデモンストレーションデータセットがプリロードされています。コンソールの関連フィールドには、それぞれの S3 のロケーションが事前に入力されています。

デモンストレーションデータセットを使用してこのチュートリアルを完了した後、同じオートメーションスタックを使用して、独自のデータセットで予測を生成できます。

次の図は、このチュートリアルで使用されるコンポーネントを示しています。



## 前提条件

チュートリアルを開始する前に、AWS アカウントにログインし、AWS CloudFormation テンプレートをインストールしていることを確認してください。

1. AWS アカウント にログインします。まだ持っていなければ [AWS アカウントを作成します](#)。
2. AWS CloudFormation テンプレートをインストールします。自分に最も近いリージョンを選択してください:
  - 東京: [ap-northeast-1](#)
  - ソウル: [ap-northeast-2](#)
  - ムンバイ: [ap-south-1](#)
  - シンガポール: [ap-southeast-1](#)
  - シドニー: [ap-southeast-2](#)
  - フランクフルト: [eu-central-1](#)
  - アイルランド: [eu-west-1](#)
  - バージニア北部: [us-east-1](#)
  - オハイオ: [us-east-2](#)
  - オレゴン: [us-west-2](#)

これにより、[NYC Taxi Dataset](#) を使用してデモンストレーションスタックがデプロイされます。

# Forecast のオートメーションのための AWS CloudFormation テンプレートのデプロイ

NYC Taxi Dataset を使用して CloudFormation テンプレートをデプロイするには

ステップ 1: デフォルトを承諾して、[Next] (次へ) を選択します。

The screenshot shows the 'Create stack' wizard in the AWS CloudFormation console, specifically Step 1: Specify template. The interface is divided into a left-hand navigation pane and a main content area. The navigation pane lists four steps: Step 1 (Specify template), Step 2 (Specify stack details), Step 3 (Configure stack options), and Step 4 (Review). The main content area is titled 'Create stack' and contains two sections. The first section, 'Prerequisite - Prepare template', explains that every stack is based on a template and provides three radio button options: 'Template is ready' (selected), 'Use a sample template', and 'Create template in Designer'. The second section, 'Specify template', explains that a template is a JSON or YAML file and provides two radio button options: 'Amazon S3 URL' (selected) and 'Upload a template file'. Below these options, there is a text input field for the 'Amazon S3 URL' containing the URL: `https://s3.amazonaws.com/solutions-reference/improving-forecast-accuracy-with-machine-learning/latest/improving-forecast-accuracy-with-mach`. A 'View In Designer' button is located to the right of the URL field. At the bottom right of the wizard, there are 'Cancel' and 'Next' buttons.

ステップ 2: 通知用のメールアドレスを入力して、[Next] (次へ) を選択します。

**Datasets Configuration**

Target Time Series URL  
URL (S3, HTTP or HTTPS) for target time series data

s3://amazon-forecast-samples/automation\_solution/demo-nyctaxi/nyctaxi\_weather\_auto.csv

Related Time Series URL (or blank)  
URL (S3, HTTP or HTTPS) for related time series data

s3://amazon-forecast-samples/automation\_solution/demo-nyctaxi/nyctaxi\_weather\_auto.related.csv

Item Metadata URL (or blank)  
URL (S3, HTTP or HTTPS) for item metadata

s3://amazon-forecast-samples/automation\_solution/demo-nyctaxi/nyctaxi\_weather\_auto.metadata.csv

**Forecast Stack (Optional)**  
If provided, use an existing Improving Forecast Accuracy with Machine Learning stack  
Existing forecast stack name

**Improving Forecast Accuracy with Machine Learning Configuration**

Email  
Email to notify with forecast results

youremail@sample.com

**Deployment Configuration**

CloudWatch Log Level  
Change the verbosity of the logs output to CloudWatch

WARNING

Cancel Previous Next

ステップ 3: デフォルトを受け入れて、[Next] (次へ) を選択します。

ステップ 4: [Capabilities] (機能) で、両方のチェックボックスをオンにして、AWS CloudFormation が AWS Identity and Access Management (IAM) リソースとネストされたスタックを作成できるようにします。[スタックの作成] を選択します。

► Quick-create link

**Capabilities**

**ⓘ The following resource(s) require capabilities: [AWS::IAM::Role, AWS::CloudFormation::Stack]**

This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. [Learn more](#)

For this template, AWS CloudFormation might require an unrecognized capability: CAPABILITY\_AUTO\_EXPAND. Check the capabilities of these resources. [Learn more](#)

I acknowledge that AWS CloudFormation might create IAM resources with custom names.

I acknowledge that AWS CloudFormation might require the following capability:  
CAPABILITY\_AUTO\_EXPAND

Cancel Previous Create change set Create stack

Forecast に AWS CloudFormation テンプレートをデプロイしました。

## クリーンアップ

この AWS CloudFormation テンプレートをデプロイした後、新しく作成されたリソースをクリーンアップし、独自のデータセットを使用して AWS CloudFormation スタックをデプロイし、他のデプロイオプションを詳しく確認できます。

- **クリーンアップ:** デモスタックを削除すると、「Improving Forecast Accuracy with Machine Learning」スタックが保持されます。「Improving Forecast Accuracy with Machine Learning」スタックを削除すると、S3、Athena、QuickSight、および Forecast のすべてのデータが保持されません。
- **独自のデータセットの使用:** この AWS CloudFormation テンプレートを独自の時系列データでデプロイするには、ステップ 2 の「データセット設定」のセクションでデータセットの S3 の場所を入力します。
- **その他のデプロイオプション:** その他のデプロイオプションについては、「[Automated Deployment](#)」(自動デプロイ) を参照してください。データが既に利用可能な場合は、デモデータなしでスタックをデプロイできます。

# データセットのインポート

データセットには、[予測子](#)のトレーニングに使用するデータが含まれます。1 つ以上の Amazon Forecast データセットを作成し、トレーニングデータをインポートします。データセットグループは、一連の変化するパラメータを一定の期間にわたって詳細に説明する補完的なデータセットのコレクションです。データセットグループを作成したら、それを使用して予測子をトレーニングします。

各データセットグループには、[データセット](#)タイプ (ターゲット時系列、関連する時系列、項目メタデータ) ごとに 1 つ、最大 3 つのデータセットを含めることができます。

Forecast データセットとデータセットグループを作成および管理するには、Forecast コンソール、AWS Command Line Interface ( AWS CLI )、または AWS SDK を使用できます。

Forecast データセットの例については、[「Amazon Forecast サンプル GitHubリポジトリ」](#)を参照してください。

## トピック

- [データセット](#)
- [データセットグループ](#)
- [データ収集頻度の競合の解決](#)
- [関連する時系列データセットの使用](#)
- [項目メタデータデータセットの使用](#)
- [事前定義済みのデータセットドメインとデータセットタイプ](#)
- [データの更新](#)
- [欠落した値の処理](#)
- [Forecast のデータセットのガイドライン](#)

## データセット

Forecast データセットを作成および管理するために、[CreateDataset](#) および [DescribeDataset](#) オペレーションを含む Forecast API を使用できます。Forecast API の完全なリストについては、「[API リファレンス](#)」を参照してください。

データセットを作成するときは、次のような情報を入力します。

- データを記録した頻度/間隔。たとえば、週ごとに小売商品の売上を集計して記録するとします。[使用開始](#) 演習では、1時間あたりの平均電力使用量のデータを使用します。
- 予測形式 (ドメイン) とデータセットタイプ (ドメイン内)。データセットドメインは、実行する予測のタイプを指定しますが、データセットタイプは、トレーニングデータを Forecast フレンドリーなカテゴリに整理するのに役立ちます。
- データセットスキーマ。スキーマは、データセットの列ヘッダーをマップします。たとえば、需要をモニタリングする場合、1時間ごとの商品販売データを複数の店舗で収集する場合があります。この場合、スキーマは、左から右への順序を定義し、その順序でトレーニングデータファイルにタイムスタンプ、場所、および1時間あたりの売上が表示されます。スキーマは、string または integer など、各列のデータタイプも定義します。
- geolocation およびタイムゾーン情報。geolocation 属性は、スキーマ内において属性タイプ geolocation で定義されています。タイムゾーン情報は、[CreateDatasetImportJob](#) オペレーションで定義されます。[Weather Index](#) を有効にするには、geolocation とタイムゾーンの両方のデータを含める必要があります。

Forecast データセット内の各列は、予測ディメンションまたは機能を表します。予測ディメンションは、location や store など、時間の経過とともに変化しないデータの側面を表します。予測機能には、price や promotion など、時間の経過とともに変化するデータ内のパラメータが含まれます。timestamp や itemId のような一部のディメンションは、ターゲット時系列および関連する時系列データセットで必要です。

## データセットドメインとデータセットタイプ

Forecast データセットを作成するときは、ドメインとデータセットタイプを選択します。Forecast は、小売需要やウェブトラフィックの予測など、さまざまなユースケース向けにドメインを提供します。カスタムドメインを作成することもできます。Forecast ドメインの詳細なリストについては、「[事前定義済みのデータセットドメインとデータセットタイプ](#)」を参照してください。

各ドメイン内で、Forecast ユーザーは次のタイプのデータセットを指定できます。

- ターゲット時系列データセット (必須) - トレーニングデータが時系列、かつ、予想を生成するフィールドが含まれる場合は、このデータセットタイプを使用します。このフィールドは、ターゲットフィールドと呼ばれます。
- 関連する時系列データセット (オプション) - トレーニングデータは時系列だが、ターゲットフィールドが含まれていない場合は、このデータセットタイプを選択します。たとえば、項目の需要を予測している場合、関連する時系列データセットにはフィールドとして price が含まれますが、demand は含まれません。

- 項目メタデータデータセット (オプション) - トレーニングデータが時系列データではなく、ターゲット時系列または関連する時系列のデータセットのアイテムに関するメタデータ情報を含む場合は、このデータセットタイプを選択します。例えば、アイテムの需要を予想している場合、アイテムのメタデータデータセットはディメンションとして color または brand を持つ場合があります。

Forecast は、[CNN-QR](#) または [DeepAR+](#) アルゴリズムを使用する場合に、アイテムメタデータのデータセットタイプによって提供されるデータのみを考慮します。

項目メタデータは、予測を行うための直接的な履歴データはほとんどないが、類似したメタデータ属性を持つ項目の履歴データがある、コールドスタート予測シナリオで特に便利です。項目のメタデータを含めると、Forecast は類似の時系列に基づいてコールドスタート予想を作成するため、より正確な予想を作成できます。

トレーニングデータの情報と予測する内容によっては、複数のデータセットを作成する場合があります。

たとえば、靴や靴下などの小売商品の需要の予測を生成するとします。RETAIL ドメインに次のデータセットを作成します。

- ターゲット時系列データセット - 小売商品の履歴時系列需要データが含まれます (例: item\_id、timestamp、およびターゲットフィールド demand)。予測するターゲットフィールドを指定するため、データセットグループには少なくとも 1 つのターゲット時系列データセットが必要です。

ターゲット時系列データセットに最大 10 個のディメンションを追加することもできます。データセットグループにターゲット時系列データセットのみを含める場合、詳細度のみの項目レベルまたは予測ディメンションレベルで予測を作成できます。詳細については、「[CreatePredictor](#)」を参照してください。

- 関連する時系列データセット - price または revenue など、ターゲットフィールド以外の履歴時系列データが含まれます。関連する時系列データはターゲット時系列データにマップできる必要があるため、関連する時系列データセットごとに同じ識別フィールドが含まれている必要があります。RETAIL ドメインでは、item\_id や timestamp になります。

関連する時系列データセットには、ターゲットの時系列データセットから作成された予測を微調整するデータが含まれている場合があります。たとえば、予測を生成する将来の日付の price データを関連する時系列データセットに含めることができます。これにより、Forecast は、コンテキ



ストの追加ディメンションを使用して予測を行うことができます。詳細については、「[関連する時系列データセットの使用](#)」を参照してください。

- 項目メタデータデータセット – 小売商品のメタデータが含まれます。たとえば、メタデータには、brand、category、color、genre などがあります。

### 予測ディメンションを持つデータセットの例

前述の例の続きとして、店舗の過去の売上に基づき、靴と靴下の需要を予測するとします。次のターゲット時系列データセットで、store は時系列予測ディメンション、demand はターゲットフィールドです。靴下は 2 つの店舗 (NYC と SFO) で販売されており、靴は ORD でのみ販売されています。

このテーブルの最初の 3 行には、NYC、SFO、および ORD ストアで有効な最初の販売データが含まれています。最後の 3 行には、各店舗の最後に記録された販売データが含まれています。... 行は、最初のエントリと最後のエントリの間に記録されたすべての商品の販売データを表します。

timestamp	item_id	store	demand
2019-01-01	socks	NYC	25
2019-01-05	socks	SFO	45
2019-02-01	shoes	ORD	10
...			
2019-06-01	socks	NYC	100
2019-06-05	socks	SFO	5
2019-07-01	shoes	ORD	50

## データセットスキーマ

各データセットには、トレーニングデータのフィールドのユーザー指定の JSON マッピングであるスキーマが必要です。データセットに含める必要のあるディメンションとオプションのディメンションおよび機能がいずれも一覧表示されています。

データセットに geolocation 属性が含まれている場合は、スキーマ内で属性タイプ geolocation を使用して属性を定義します。詳細については、「[Adding Geolocation information](#)」(geolocation の追加) を参照してください。[Weather Index](#) を適用するには、ターゲットの時系列および関連する時系列データセットに geolocation 属性を含める必要があります。

一部のドメインには、含めることをお勧めするオプションのディメンションがあります。オプションのディメンションは、このガイドの後半にある各ドメインの説明に記載されています。例については、[RETAIL ドメイン](#)を参照してください。オプションのすべてのディメンションは、string のデータ型になります。

スキーマはすべてのデータセットに必要です。上記のターゲット時系列データセットの例に付随するスキーマを次に示します。

```
{
  "attributes": [
    {
      "AttributeName": "timestamp",
      "AttributeType": "timestamp"
    },
    {
      "AttributeName": "item_id",
      "AttributeType": "string"
    },
    {
      "AttributeName": "store",
      "AttributeType": "string"
    },
    {
      "AttributeName": "demand",
      "AttributeType": "float"
    }
  ]
}
```

このスキーマを使用するデータセットにトレーニングデータをアップロードすると、Forecast は timestamp フィールドがデータの列 1、item\_id フィールドが列 2、store フィールドが列 3、そしてターゲットフィールドである demand フィールドが列 4 であるとみなします。

関連する時系列データセットタイプの場合、すべての関連機能に属性タイプ float または integer が必要です。項目メタデータデータセットタイプの場合は、すべての機能に文字列の属性タイプが必要です。詳細については、「[SchemaAttribute](#)」を参照してください。

### Note

データセットのすべての列に `attributeName` と `attributeType` のペアが必要です。Forecast では、多くの名前が予約されており、これらをスキーマ属性の名前として使用することはできません。予約名のリストについては、「[予約フィールド名](#)」を参照してください。

## データセットグループ

データセットグループは、1 つから 3 つの認定データセットのコレクションで、データセットタイプごとに 1 つです。データセットをデータセットグループにインポートし、データセットグループを使用して予測子をトレーニングします。

Forecast には、データセットグループを作成し、データセットをそのグループに追加するための以下のオペレーションが含まれています:

- [CreateDatasetGroup](#)
- [UpdateDatasetGroup](#)

## データ収集頻度の競合の解決

Forecast では、[CreateDataset](#) オペレーションで指定したデータ頻度と一致しないデータを使用して予測子をトレーニングできます。例えば、一部のデータが時間の先頭にタイムスタンプが付いていないデータ (02:20、02:45) をインポートできます。Forecast では、指定したデータ頻度を使用してデータについて学習します。次に、Forecast は、予測子のトレーニング中にデータを集計します。詳細については、[さまざまな予想頻度のデータ集計](#)を参照してください。

## 関連する時系列データセットの使用

関連する時系列データセットには、ターゲット時系列データセットに含まれていない時系列データが含まれており、予測子の精度が向上する場合があります。

例えば、需要予測ドメインで、ターゲット時系列データセットには、`timestamp` や `item_id` デイメンションが含まれているのに対し、補完的な関連する時系列には、`item price`、`promotion`、`weather` などの追加データ機能も含まれます。

関連する時系列データセットには、最大 10 個の予測ディメンション (ターゲット時系列データセット内のものと同じ) と最大 13 個の関連する時系列機能を含めることができます。

### Python ノートブック

関連する時系列データセットの使用に関する step-by-step ガイドについては、[「関連する時系列の組み込み」](#)を参照してください。

## トピック

- [履歴およびフォワードルックの関連する時系列](#)
- [関連する時系列データセットの検証](#)
- [例: フォワードルックの関連する時系列ファイル](#)
- [例: 詳細度の予測](#)
- [レガシー予測子および関連する時系列](#)

## 履歴およびフォワードルックの関連する時系列

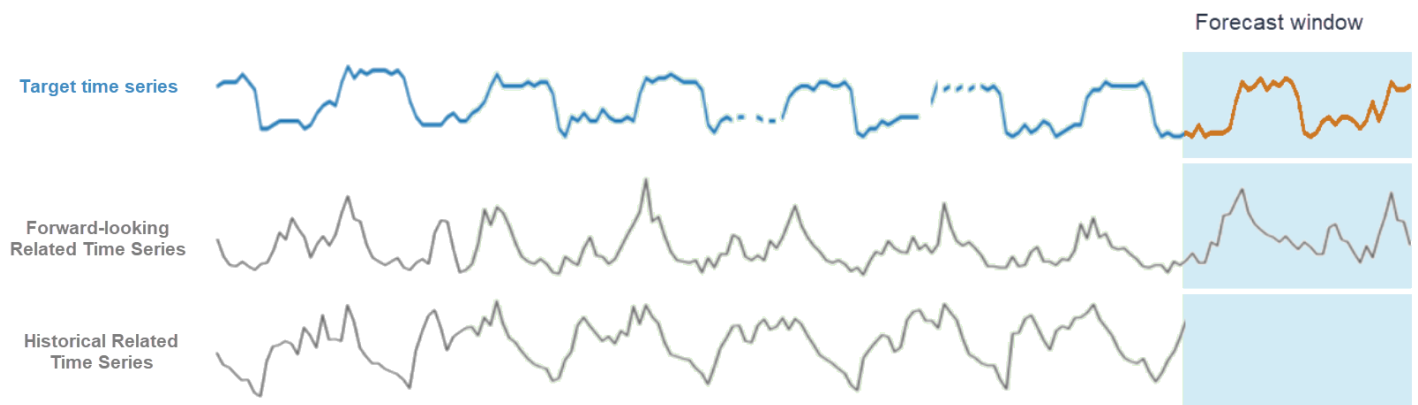
### Note

予測期間内の値を含む関連する時系列は、フォワードルックの時系列として扱われます。

関連する時系列には、次の 2 つの形式があります。

- 履歴時系列: 予測期間内にデータポイントがない時系列。
- フォワードルックの時系列: 予測期間内のデータポイントを持つ時系列。

履歴の関連する時系列には、予測期間までのデータポイントが含まれ、予測期間内のデータポイントは含まれません。フォワードルックの関連する時系列には、予測期間までおよび予測期間内のデータポイントが含まれます。



## 関連する時系列データセットの検証

関連する時系列データセットには次の制限があります。

- ターゲット時系列のターゲット値を含めることはできません。
- これには、`item_id` および `timestamp` デイメンションと、少なくとも 1 つの関連機能 (`price` など) が含まれている必要があります。
- 関連する時系列機能データは、`int` または `float` データ型である必要があります。
- ターゲット時系列全体を使用するには、ターゲット時系列データセットのすべてのアイテムも関連する時系列データセットに含まれている必要があります。関連する時系列にターゲット時系列のアイテムのサブセットのみが含まれている場合、モデルの作成と予測の生成は、その特定のアイテムのサブセットに限定されます。

例えば、ターゲット時系列に 1000 アイテムが含まれ、関連する時系列データセットに 100 アイテムしか含まれていない場合、モデルと予測はそれらの 100 アイテムのみに基づきます。

- 関連する時系列データセットにデータを記録する頻度は、予測を生成する間隔 (予測の精度) と一致する必要があります。

たとえば、週単位の精度で予測を生成する場合、ターゲットの時系列にデータを記録する頻度が毎日であっても、関連する時系列でデータを記録する頻度も週単位である必要があります。

- 関連する時系列データセット内の各項目のデータは、ターゲット時系列データセット内の対応する `item_id` の先頭の `timestamp` またはそれより前に開始する必要があります。

たとえば、`socks` のターゲット時系列データが 2019-01-01 に開始し、`shoes` のターゲット時系列データが 2019-02-01 に開始する場合、`socks` の関連する時系列データは 2019-01-01 以前、`shoes` のデータは 2019-02-01 以前に開始する必要があります。

- フォワードルックの関連する時系列データセットについては、すべての項目の最後のタイムスタンプは、ユーザーが指定した予測ウィンドウ (予測期間と呼ばれます) の最後のタイムスタンプ上にある必要があります。

以下の関連する時系列ファイルの例では、靴下と靴の両方の timestamp データは 2019-07-01 (最後に記録されたタイムスタンプ) かつ予測期間の後に終了する必要があります。ターゲット時系列のデータ頻度が毎日、予測期間が 10 日である場合、2019-07-11 までは、フォワードルックの関連する時系列ファイルで毎日のデータポイントを指定する必要があります。

- 履歴の関連する時系列データセットについては、すべてのアイテムの最後のタイムスタンプは、ターゲットの時系列の最後のタイムスタンプと一致する必要があります。

以下の関連する時系列ファイルの例では、靴下と靴の両方の timestamp データは 2019-07-01 (最後に記録されたタイムスタンプ) に終了する必要があります。

- 関連する時系列データセットで指定される Forecast デイメンションは、ターゲット時系列データセットで指定されたデイメンションと等しいか、サブセットである必要があります。
- 関連する時系列に欠落値を含めることはできません。関連する時系列データセットの欠落した値については、「[欠落した値の処理](#)」を参照してください。

## 例: フォワードルックの関連する時系列ファイル

次の表は、正しく設定された関連する時系列データセットファイルを示しています。この例では、次のことを前提としています。

- 最後のデータポイントは、2019-07-01 にターゲット時系列データセットに記録されました。
- 予測期間は 10 日です。
- 予測の精度は毎日です (D)。

「...」行は、前の行と後続の行の間にあるすべてのデータポイントを示します。

timestamp	item_id	store	price
2019-01-01	socks	NYC	10
2019-01-02	socks	NYC	10
2019-01-03	socks	NYC	15

timestamp	item_id	store	price
...			
2019-06-01	socks	NYC	10
...			
2019-07-01	socks	NYC	10
...			
2019-07-11	socks	NYC	20
2019-01-05	socks	SFO	45
...			
2019-06-05	socks	SFO	10
...			
2019-07-01	socks	SFO	10
...			
2019-07-11	socks	SFO	30
2019-02-01	shoes	ORD	50
...			
2019-07-01	shoes	ORD	75
...			
2019-07-11	shoes	ORD	60

## 例: 詳細度の予測

次の表は、ターゲット時系列および関連する時系列について週単位の精度で予測する、互換性のあるデータ記録頻度を示します。関連する時系列データセットのデータは集計できないため、Forecast は、選択した詳細度と同じ関連する時系列のデータ頻度のみを受け入れます。

ターゲット入力データの頻度	関連する時系列の頻度	詳細度の予測	Forecast でのサポート
1 日 1 回	毎週	毎週	はい
毎週	毎週	毎週	はい
該当なし	毎週	毎週	はい
1 日 1 回	1 日 1 回	毎週	いいえ

## レガシー予測子および関連する時系列













### Note

既存の予測子を にアップグレードするには AutoPredictor、「」を参照してください。 [the section called “へのアップグレード AutoPredictor”](#)

従来の予測子を使用する場合、[CNN-QR](#)、[DeepAR +](#)、および [Prophet](#) のアルゴリズムを使用して予測子をトレーニングするときに、関連する時系列データセットを使用できます。[NPTS](#)、[ARIMA](#)、および [ETS](#) は、関連する時系列データを受け入れません。

次の表は、Amazon Forecast の各アルゴリズムが受け入れる、関連する時系列のタイプを示しています。



	CNN-QR	DeepAR +	Prophet	NPTS	ARIMA	ETS
履歴の関連する時系列						
フォワードルックの時系列						

AutoML を使用する場合、履歴の関連する時系列データとフォワードルックの関連する時系列データの両方を提供でき、Forecast は該当する場合にのみそれらの時系列を使用します。

フォワードルックの関連する時系列データを提供する場合、Forecast は、CNN-QR、DeepAR+、および Prophet で関連データを使用し、NPTS、ARIMA、および ETS で関連データを使用しません。履歴の関連する時系列データが提供された場合、Forecast は、CNN-QR で関連データを使用し、DeepAR+、Prophet、NPTS、ARIMA、および ETS で関連データを使用しません。

## 項目メタデータデータセットの使用

項目メタデータデータセットには、ターゲットの時系列データセットの項目に有益なコンテキストを提供するカテゴリデータが含まれています。関連する時系列データセットとは異なり、項目メタデータデータセットは静的な情報を提供します。つまり、項目の色やブランドのように、データ値は時間が経過しても一定に保たれます。アイテムメタデータデータセットは、データセットグループへのオプションの追加です。項目メタデータを使用できるのは、ターゲットの時系列データセット内のすべての項目が、対応する項目メタデータデータセットに存在する場合だけです。

項目メタデータには、特定の項目のブランド、色、モデル、カテゴリ、製造場所、その他の補足機能が含まれる場合があります。たとえば、項目メタデータデータセットは、32 GB のストレージを持つ黒の Amazon E-reader の売り上げを表す、ターゲットの時系列データセットで見つかった需要データの一部のコンテキストを提供することがあります。これらの特性は day-to-day または から変更されないため hour-to-hour、アイテムメタデータデータセットに属します。

項目メタデータは、時系列データの説明パターンを検出して追跡するのに便利です。データセットグループに項目メタデータデータセットを含めると、Forecast は、項目間の類似性に基づいてより正

確な予測を行うようにモデルをトレーニングできます。たとえば、Amazon 製の仮想アシスタント製品は、他の会社が製造した製品よりも売り切れる可能性が高く、サプライチェーンをそれに応じて計画することができます。

項目メタデータは、予測を行うための直接的な履歴データがないが、類似したメタデータ属性を持つ項目の履歴データがある、コールドスタート予想シナリオで特に便利です。項目のメタデータにより、Forecast はコールドスタート項目と同様の項目を活用して予想を作成できます。

項目のメタデータを含めると、Forecast は類似の時系列に基づいてコールドスタート予想を作成するため、より正確な予想を作成できます。コールドスタート予想は、項目メタデータデータセットにはあるが直近の時系列にはない項目について生成されます。まず、Forecast は、コールドスタート以外の項目 (後続時系列の履歴データを含む項目) の予想を生成します。次に、各コールドスタート項目について、項目メタデータデータセットを使用して最近傍項目を検索します。次に、これらの最近傍を使用してコールドスタート予想を作成します。

項目メタデータデータセットの各行には、最大 10 個のメタデータフィールドを含めることができます。そのうちの 1 つは、メタデータをターゲットの時系列の項目に一致させる識別フィールドである必要があります。すべてのデータセットタイプと同様に、各フィールドの値はデータセットスキーマによって指定されます。

### Python ノートブック

アイテムメタデータの使用に関する step-by-step ガイドについては、[「アイテムメタデータの組み込み」](#)を参照してください。

## トピック

- [例: 項目メタデータファイルとスキーマ](#)
- [レガシー予測子およびアイテムメタデータ](#)
- [その他の参照資料](#)

## 例: 項目メタデータファイルとスキーマ

次の表は、Amazon E-reader を説明する、正しく設定された項目メタデータデータセットファイルのセクションを示しています。この例では、ヘッダー行がデータセットのスキーマを表し、リストされた各項目が、対応するターゲットの時系列データセットにあると仮定します。

item_id	brand	model	color	waterproof
1	Amazon	paperwhite	black	はい
2	Amazon	paperwhite	blue	はい
3	Amazon	base_model	black	なし
4	Amazon	base_model	white	なし
...				

以下は、CSV 形式で表された同じ情報です。

```
1,amazon,paperwhite,black,yes
2,amazon,paperwhite,blue,yes
3,amazon,base_model,black,no
4,amazon,base_model,white,no
...
```

このサンプルデータセットのスキーマを次に示します。

```
{
  "attributes": [
    {
      "AttributeName": "item_id",
      "AttributeType": "string"
    },
    {
      "AttributeName": "brand",
      "AttributeType": "string"
    },
    {
      "AttributeName": "model",
      "AttributeType": "string"
    },
    {
      "AttributeName": "color",
      "AttributeType": "string"
    },
    {
```

```
        "AttributeName": "waterproof",
        "AttributeType": "string"
    }
]
}
```

## レガシー予測子およびアイテムメタデータ

### Note

既存の予測子を にアップグレードするには AutoPredictor、「」を参照してください。 [the section called “へのアップグレード AutoPredictor”](#)

レガシー予測子を使用する場合、[CNN-QR](#) または [DeepAR+](#) アルゴリズムを使用して予測子をトレーニングするときに、アイテムメタデータを使用できます。AutoML を使用する場合、アイテムメタデータを提供でき、Forecast は該当する場合にのみそれらの時系列を使用します

## その他の参照資料

アイテムメタデータデータセットの使用に関する詳細なチュートリアルについては、「[Amazon Forecast サンプル GitHub リポジトリ](#)」の「[アイテムメタデータデータセットを予測子に組み込む](#)」を参照してください。

## 事前定義済みのデータセットドメインとデータセットタイプ

予測子をトレーニングするには、1つまたは複数のデータセットを作成し、データセットグループに追加して、トレーニング用のデータセットグループを提供します。

作成したデータセットごとに、データセットドメインとデータセットタイプを関連付けます。データセットドメインは、一般的ユースケースのために事前定義されたデータセットスキーマを指定し、モデルアルゴリズムやハイパーパラメータには影響しません。

Amazon Forecast では次のデータセットドメインがサポートされています。

- [RETAIL ドメイン](#) – 小売の需要予測
- [INVENTORY\\_PLANNING ドメイン](#) – サプライチェーンとインベントリの計画
- [EC2 CAPACITY ドメイン](#) – Amazon Elastic Compute Cloud (Amazon EC2) の容量を予測するため

- [WORK\\_FORCE ドメイン](#) – 従業員の計画
- [WEB\\_TRAFFIC ドメイン](#) – 今後のウェブトラフィックの見積もり
- [METRICS ドメイン](#) – 収益およびキャッシュフローなどの予測メトリクス
- [CUSTOM ドメイン](#) – その他すべての時系列予測のタイプ

ドメインごとに、1~3つのデータセットタイプがあります。ドメイン用に作成するデータセットタイプは、所有しているデータのタイプとトレーニングに含めるものに基づいています。

各ドメインにはターゲット時系列データセットが必要です。オプションで、関連する時系列および項目メタデータデータセットタイプをサポートします。

データセットタイプは次のとおりです。

- **ターゲット時系列** – 唯一必要なデータセットタイプ。このタイプは、予測を生成するターゲットフィールドを定義します。たとえば、一連の製品の売上を予測する場合は、予測する製品ごとに履歴の時系列データのデータセットを作成する必要があります。同様に、予測する収益、キャッシュフロー、売上などのメトリクス用のターゲット時系列データセットを作成できます。
- **関連する時系列** – ターゲットの時系列データに関連する時系列データ。たとえば、価格は製品の売上データに関連しているため、関連する時系列として指定できます。
- **項目メタデータ** – ターゲットの時系列データに適用可能なメタデータ。例えば、特定の商品の売上を予測している場合 ブランド、色、ジャンルなどの商品の属性は項目メタデータの一部になります。EC2 インスタンスの EC2 キャパシティを予測するとき、メタデータにはインスタンスタイプの CPU とメモリが含まれる場合があります。

データセットのタイプごとに、入力データに特定の必須フィールドが含まれている必要があります。Amazon Forecast が提案したオプションのフィールドを含めることもできます。

次の例は、データセットドメインとそれに対応するデータセットタイプを選択する方法を示しています。

#### Example 例 1 : RETAIL ドメインのデータセットタイプ

商品の需要予測に関心のある小売業者であれば、RETAIL ドメインに次のデータセットを作成できます。

- **ターゲット時系列**は、各アイテム (小売業者が販売する各製品) の履歴時系列需要 (売上) データの必須データセットです。RETAIL ドメインでは、このデータセットタイプでは、データセッ

トに `item_id`、`timestamp`、および `demand` の各フィールドが含まれている必要があります。`demand` フィールドは予測ターゲットであり、通常は特定の週または日に小売業者によって販売された商品の数です。

- オプションで、関連する時系列タイプのデータセット。RETAIL ドメインでは、このタイプには `price`、`inventory_onhand`、`webpage_hits` などのオプションの、ただし推奨される時系列情報を含めることができます。
- オプションで、項目メタデータタイプのデータセット。RETAIL ドメインでは、Amazon Forecast は `brand`、`color`、`category`、および `genre` など、ターゲット時系列で提供した項目に関連するメタデータ情報を提供することをお勧めします。

### Example 例 2: METRICS ドメインのデータセットタイプ

収益、売上、キャッシュフローなど、組織の主要なメトリクスを予測する場合は、Amazon Forecast に次のデータセットを提供できます。

- 予測するメトリクスの履歴時系列データを提供するターゲット時系列データセット。組織内のすべてのビジネスユニットの収益を予測する場合は、`metric`、`business unit`、および `metric_value` の各フィールドを使用して `target time series` データセットを作成できます。
- `category` や `location` など、必要とされない各メトリクスのメタデータがある場合は、関連する時系列および項目メタデータタイプのデータセットを提供します。

ターゲットメトリクスの予測を生成するには、少なくとも Forecast のターゲット時系列データセットを提供する必要があります。

### Example 例 3: CUSTOM ドメインのデータセットタイプ

予測アプリケーションのトレーニングデータは、どの Amazon Forecast ドメインにも適合しない可能性があります。そのような場合は、CUSTOM ドメインを選択してください。ターゲット時系列データセットを提供する必要がありますが、独自のカスタムフィールドを追加することができます。

**使用開始** 演習では、クライアントの電力使用量を予測します。電力使用量トレーニングデータはどのデータセットドメインにも適合しないため、CUSTOM ドメインを使用しました。この演習では、ターゲット時系列タイプという 1 つのデータセットタイプのみを使用します。データフィールドをデータセットタイプに必要な最小フィールドにマップします。

## RETAIL ドメイン

RETAIL ドメインは、次のデータセットタイプをサポートしています。データセットのタイプごとに、必須フィールドとオプションフィールドをリストします。トレーニングデータの列にフィールドをマッピングする方法については、「[データセットドメインとデータセットタイプ](#)」を参照してください。

### トピック

- [ターゲット時系列データセットタイプ](#)
- [関連する時系列データセットタイプ](#)
- [項目メタデータデータセットタイプ](#)

### ターゲット時系列データセットタイプ

ターゲット時系列は、小売組織によって販売された各品目または製品の履歴時系列データです。以下のフィールドが必要です。

- `item_id` (文字列) – 需要を予測する商品または製品の固有の ID。
- `timestamp` (タイムスタンプ)
- `demand` (浮動) – タイムスタンプ時の商品の販売数。これは、Amazon Forecast が予測を生成するターゲットフィールドでもあります。

次のディメンションはオプションです。予測の精度を変更するために使用できます。

- `location` (文字列) – 商品が販売されたストアの場所。これは、複数のストア/場所がある場合のみ使用してください。

これらの必須フィールド、およびオプションのディメンションのみが含まれている状態が理想です。その他の時系列情報は関連する時系列データセットに含めます。

### 関連する時系列データセットタイプ

特定の日に項目が受け取った価格やウェブヒット数など、関連する時系列データセットを Amazon Forecast に提供できます。情報が多いほど、予測はより正確になります。以下のフィールドが必要です。

- `item_id` (文字列)

- timestamp (タイムスタンプ)

次のフィールドはオプションです。予測結果の改善に役立ちます。

- price (浮動) – タイムスタンプ時の商品の価格。
- promotion\_applied (integer; 1=true, 0=false) – タイムスタンプ時にその商品のマーケティングプロモーションがあったかどうかを指定するフラグ。

必須および推奨されるオプションのフィールドに加えて、トレーニングデータに他のフィールドを含めることができます。データセットに他のフィールドを含めるには、データセットを作成するときにスキーマにフィールドを指定します。

## 項目メタデータデータセットタイプ

このデータセットは、需要が予測されている商品のメタデータ (属性) に関する情報を Amazon Forecast に提供します。以下のフィールドが必要です。

- item\_id (文字列)

次のフィールドはオプションです。予測結果の改善に役立ちます。

- category (文字列)
- brand (文字列)
- color (文字列)
- genre (文字列)

必須および推奨されるオプションのフィールドに加えて、トレーニングデータに他のフィールドを含めることができます。データセットに他のフィールドを含めるには、データセットを作成するときにスキーマにフィールドを指定します。

## CUSTOM ドメイン

CUSTOM ドメインは、次のデータセットタイプをサポートしています。データセットのタイプごとに、必須フィールドとオプションフィールドをリストします。トレーニングデータの列にフィールドをマッピングする方法については、「[データセットドメインとデータセットタイプ](#)」を参照してください。



## トピック

- [ターゲット時系列データセットタイプ](#)
- [関連する時系列データセットタイプ](#)
- [項目メタデータデータセットタイプ](#)

## ターゲット時系列データセットタイプ

以下のフィールドが必要です。

- `item_id` (文字列)
- `timestamp` (タイムスタンプ)
- `target_value` (浮動小数点整数) - これは、Amazon Forecast が予測を生成する `target` フィールドです。

これらの必須フィールドのみ含まれている状態が理想です。その他の時系列情報は関連する時系列データセットに含めます。

## 関連する時系列データセットタイプ

以下のフィールドが必要です。

- `item_id` (文字列)
- `timestamp` (タイムスタンプ)

必須フィールドに加えて、トレーニングデータに他のフィールドを含めることができます。データセットに他のフィールドを含めるには、データセットを作成するときにスキーマにフィールドを指定します。

## 項目メタデータデータセットタイプ

以下のフィールドが必要です。

- `item_id` (文字列)

次のフィールドはオプションです。予測結果の改善に役立ちます。

- `category` (文字列)

必須および推奨されるオプションのフィールドに加えて、トレーニングデータに他のフィールドを含めることができます。データセットに他のフィールドを含めるには、データセットを作成するときにスキーマにフィールドを指定します。

## INVENTORY\_PLANNING ドメイン

INVENTORY\_PLANNING ドメインを使用して、原材料の需要を予測し、特定の商品の在庫をどれだけ在庫するかを決定します。以下のデータセットタイプがサポートされています。データセットのタイプごとに、必須フィールドとオプションフィールドをリストします。トレーニングデータの列にフィールドをマッピングする方法については、「[データセットドメインとデータセットタイプ](#)」を参照してください。

### トピック

- [ターゲット時系列データセットタイプ](#)
- [関連する時系列データセットタイプ](#)
- [項目メタデータデータセットタイプ](#)

### ターゲット時系列データセットタイプ

以下のフィールドが必要です。

- item\_id (文字列)
- timestamp (タイムスタンプ)
- demand (浮動) - これは、Amazon Forecast が予測を生成するための target フィールドです。

次のディメンションはオプションです。予測の精度を変更するために使用できます。

- location (文字列) - 商品が在庫されている流通センターの場所。これは、複数のストア/場所がある場合にのみ使用してください。

これらの必須フィールド、およびオプションのディメンションのみが含まれている状態が理想です。その他の時系列情報は関連する時系列データセットに含めます。

### 関連する時系列データセットタイプ

以下のフィールドが必要です。

- item\_id (文字列)
- timestamp (タイムスタンプ)

次のフィールドはオプションです。予測結果の改善に役立ちます。

- price (浮動) – 商品の価格

必須および推奨されるオプションのフィールドに加えて、トレーニングデータに他のフィールドを含めることができます。データセットに他のフィールドを含めるには、データセットを作成するときにスキーマにフィールドを指定します。

## 項目メタデータデータセットタイプ

以下のフィールドが必要です。

- item\_id (文字列)

次のフィールドはオプションです。予測結果の改善に役立ちます。

- category (文字列) – 商品のカテゴリ。
- brand (文字列) – 商品のブランド。
- lead\_time (文字列) – 商品を製造するためのリードタイム (日数)。
- order\_cycle (文字列) – 注文サイクルは、作業が開始されたときに開始され、商品の出荷準備が整ったときに終了します。
- safety\_stock (文字列) – その商品を保持するための最小在庫量。

必須および推奨されるオプションのフィールドに加えて、トレーニングデータに他のフィールドを含めることができます。データセットに他のフィールドを含めるには、データセットを作成するときにスキーマにフィールドを指定します。

## EC2 CAPACITY ドメイン

Amazon EC2 の容量を予測するには、EC2 CAPACITY ドメインを使用します。以下のデータセットタイプがサポートされています。データセットのタイプごとに、必須フィールドとオプションフィールドをリストします。トレーニングデータの列にフィールドをマッピングする方法については、「[データセットドメインとデータセットタイプ](#)」を参照してください。

## ターゲット時系列データセットタイプ

以下のフィールドが必要です。

- `instance_type` (文字列) – インスタンスのタイプ (例: `c5.xlarge`)。
- `timestamp` (タイムスタンプ)
- `number_of_instances` (整数) – タイムスタンプで消費された特定のインスタンスタイプのインスタンスの数。これは、Amazon Forecast が予測を生成するための `target` フィールドです。

次のディメンションはオプションです。予測の精度を変更するために使用できます。

- `location` (文字列) – `us-west-2` や `us-east-1` AWS リージョンなどの を指定できます。これは、複数のリージョンをモデリングしている場合にのみ使用してください。

これらの必須および提案されたオプションのフィールドのみ含まれている状態が理想です。その他の時系列情報は関連する時系列データセットに含めます。

## 関連する時系列データセットタイプ

以下のフィールドが必要です。

- `instance_type` (文字列)
- `timestamp` (タイムスタンプ)

必須フィールドに加えて、トレーニングデータに他のフィールドを含めることができます。データセットに他のフィールドを含めるには、データセットを作成するときにスキーマにフィールドを指定します。

## WORK\_FORCE ドメイン

WORK\_FORCE ドメインを使用して労働力需要を予測します。以下のデータセットタイプがサポートされています。データセットのタイプごとに、必須フィールドとオプションフィールドをリストします。トレーニングデータの列にフィールドをマッピングする方法については、「[データセットドメインとデータセットタイプ](#)」を参照してください。

トピック

- [ターゲット時系列データセットタイプ](#)

- [関連する時系列データセットタイプ](#)
- [項目メタデータデータセットタイプ](#)

## ターゲット時系列データセットタイプ

以下のフィールドが必要です。

- `workforce_type` (文字列) – 予測されているタイプの労働力。たとえば、コールセンターの需要やフルフィルメントセンターの労働需要などです。
- `timestamp` (タイムスタンプ)
- `workforce_demand` (浮動小数点整数) - これは、Amazon Forecast が予測を生成する `target` フィールドです。

次のディメンションはオプションです。予測の精度を変更するために使用できます。

- `location` (文字列) – 労働力資源が求められている場所。これは、複数のストア/場所がある場合に使用してください。

これらの必須フィールド、およびオプションのディメンションのみが含まれている状態が理想です。その他の時系列情報は関連する時系列データセットに含めます。

## 関連する時系列データセットタイプ

以下のフィールドが必要です。

- `workforce_type` (文字列)
- `timestamp` (タイムスタンプ)

必須フィールドに加えて、トレーニングデータに他のフィールドを含めることができます。データセットに他のフィールドを含めるには、データセットを作成するときにスキーマにフィールドを指定します。

## 項目メタデータデータセットタイプ

以下のフィールドが必要です。

- `workforce_type` (文字列)

次のフィールドはオプションです。予測結果の改善に役立ちます。

- wages (浮動) – その特定の労働力タイプの平均賃金。
- shift\_length (文字列) – シフトの長さ。
- location (文字列) – 労働力の場所。

必須および推奨されるオプションのフィールドに加えて、トレーニングデータに他のフィールドを含めることができます。データセットに他のフィールドを含めるには、データセットを作成するときにスキーマにフィールドを指定します。

## WEB\_TRAFFIC ドメイン

ウェブプロパティまたは一連のウェブプロパティへのウェブトラフィックを予測するには、WEB\_TRAFFIC ドメインを使用します。以下のデータセットタイプがサポートされています。関連トピックでは、データセットタイプがサポートする必須フィールドとオプションのフィールドについて説明しています。これらのフィールドをトレーニングデータの列にマッピングする方法については、「[データセットドメインとデータセットタイプ](#)」を参照してください。

トピック

- [ターゲット時系列データセットタイプ](#)
- [関連する時系列データセットタイプ](#)

### ターゲット時系列データセットタイプ

以下のフィールドが必要です。

- item\_id (文字列) – 予測されている各ウェブプロパティの一意の識別子。
- timestamp (タイムスタンプ)
- value (浮動) - これは、Amazon Forecast が予測を生成するための target フィールドです。

これらの必須フィールドのみ含まれている状態が理想です。その他の時系列情報は関連する時系列データセットに含めます。

### 関連する時系列データセットタイプ

以下のフィールドが必要です。

- `item_id` (文字列)
- `timestamp` (タイムスタンプ)

必須フィールドに加えて、トレーニングデータに他のフィールドを含めることができます。データセットに他のフィールドを含めるには、データセットを作成するときにスキーマにフィールドを指定します。

### 項目メタデータデータセットタイプ

以下のフィールドが必要です。

- `item_id` (文字列)

次のフィールドはオプションです。予測結果の改善に役立ちます。

- `category` (文字列)

必須および推奨されるオプションのフィールドに加えて、トレーニングデータに他のフィールドを含めることができます。データセットに他のフィールドを含めるには、データセットを作成するときにスキーマにフィールドを指定します。

## METRICS ドメイン

収益、売上、キャッシュフローなどの予測メトリクスには METRICS ドメインを使用します。以下のデータセットタイプがサポートされています。データセットのタイプごとに、必須フィールドとオプションフィールドをリストします。トレーニングデータの列にフィールドをマッピングする方法については、「[データセットドメインとデータセットタイプ](#)」を参照してください。

### トピック

- [ターゲット時系列データセットタイプ](#)
- [関連する時系列データセットタイプ](#)
- [項目メタデータデータセットタイプ](#)

### ターゲット時系列データセットタイプ

以下のフィールドが必要です。

- `metric_name` (文字列)
- `timestamp` (タイムスタンプ)
- `metric_value` (浮動小数点整数) - これは、Amazon Forecast が予測を生成する `target` フィールドです (例えば、特定の日に生成される収益額)。

これらの必須フィールドのみ含まれている状態が理想です。その他の時系列情報は関連する時系列データセットに含めます。

## 関連する時系列データセットタイプ

以下のフィールドが必要です。

- `metric_name` (文字列)
- `timestamp` (タイムスタンプ)

必須フィールドに加えて、トレーニングデータに他のフィールドを含めることができます。データセットに他のフィールドを含めるには、データセットを作成するときにスキーマにフィールドを指定します。

## 項目メタデータデータセットタイプ

以下のフィールドが必要です。

- `metric_name` (文字列)

次のフィールドはオプションです。予測結果の改善に役立ちます。

- `category` (文字列)

必須および推奨されるオプションのフィールドに加えて、トレーニングデータに他のフィールドを含めることができます。データセットに他のフィールドを含めるには、データセットを作成するときにスキーマにフィールドを指定します。

## データの更新

新しいデータを収集するときに、そのデータを Forecast にインポートすることをお勧めします。そのためには、代替および増分更新の 2 つのオプションがあります。代替データセットのインポート



ジョブは、既存のデータをすべて新しくインポートされたデータに置き換えます。増分更新では、新しくインポートされたデータがデータセットに追加されます。

新しいデータをインポートしたら、既存の予測子を使用してそのデータの予想を生成できます。

## トピック

- [インポートモード](#)
- [既存のデータセットの更新](#)
- [予想の更新](#)

## インポートモード

Amazon Forecast が既存のデータセットに新しいデータを追加する方法を設定するには、データセットのインポートジョブのインポートモードを指定します。デフォルトインポートモードは FULL です。インポートモードは Amazon Forecast API を使用してのみ設定できます。

- データセット内の既存のデータをすべて上書きするには、[CreateDatasetImportJob](#) API オペレーションの FULL を指定します。
- データセット内の既存のデータにレコードを追加するには、[CreateDatasetImportJob](#) API オペレーションの INCREMENTAL を指定します。既存のレコードとインポートされたレコードの時系列 ID (項目 ID、ディメンション、タイムスタンプ) が同じ場合、既存のレコードは新しくインポートされたレコードに置き換えられます。Amazon Forecast では、常に最新のタイムスタンプのレコードが使用されます。

データセットをインポートしていない場合、増分オプションは使用できません。デフォルトのインポートモードは完全な置き換えモードです。

## 増分インポートモードのガイドライン

データセットの増分インポートを実行する場合、タイムスタンプ形式、データ形式、または位置情報データを変更することはできません。これらの項目のいずれかを変更するには、データのデータセットの完全インポートを実行する必要があります。

## 既存のデータセットの更新

### ⚠ Important

デフォルトで、データセットのインポートジョブは、インポートしたデータセット内の既存のデータを置き換えます。データセットのインポートジョブの [インポートモード](#) を指定することでこれを変更できます。

データセットを更新するには、そのデータセットのデータセットインポートジョブを作成し、インポートモードを指定します。

### CLI

データセットを更新するには、`create-dataset-import-job` コマンドを使用します。`import-mode` には、`FULL` で既存のデータを置き換えるか、`INCREMENTAL` で追加するかを指定します。詳細については、「[インポートモード](#)」を参照してください。

次のコードは、新しいデータをデータセットに段階的にインポートするデータセットインポートジョブの作成方法を示しています。

```
aws forecast create-dataset-import-job \  
    --dataset-import-job-name dataset import job name \  
    --dataset-arn dataset arn \  
    --data-source "S3Config":{"KMSKeyArn":"string",  
"Path":"string", "RoleArn":"string"} \  
    --import-mode INCREMENTAL
```

### Python

データセットを更新するには、`create_dataset_import_job` メソッドを使用します。`import-mode` には、`FULL` で既存のデータを置き換えるか、`INCREMENTAL` で追加するかを指定します。詳細については、「[インポートモード](#)」を参照してください。

```
import boto3  
  
forecast = boto3.client('forecast')  
  
response = forecast.create_dataset_import_job(  
    datasetImportJobName = 'YourImportJob',  
    datasetArn = 'dataset_arn',
```

```
dataSource = {"S3Config":{"KMSKeyArn":"string", "Path":"string",  
"RoleArn":"string"}},  
importMode = 'INCREMENTAL'  
)
```

## 予想の更新

新しいデータを収集するときに、それを使用して新しい予想を生成することをお勧めします。Forecast は、更新されたデータセットをインポートするときに予測子を自動的に再トレーニングすることはありませんが、予測子を再トレーニングして、更新されたデータで新しい予想を生成できます。例えば、毎日の売上データを収集し、予想に新しいデータポイントを含める場合、更新されたデータをインポートし、それを使用して予想を生成できます。新しい予測子をトレーニングする必要はありません。新しくインポートしたデータが予想に影響を与えるようにするには、予測子を再トレーニングする必要があります。

新しいデータから予想を生成する手順は、次のとおりです。

1. 新しいデータを Amazon S3 バケットにアップロードします。新しいデータには、前回のデータセットのインポート以降に追加されたデータのみが含まれている必要があります。
2. 新しいデータを使用して増分データセットインポートジョブを作成します。新しいデータは既存のデータに追加され、予想は更新されたデータから生成されます。新しいデータファイルに以前にインポートしたデータと新しいデータの両方が含まれている場合は、完全データセットインポートジョブを作成します。
3. 既存の予測子を使用して新しい予測を作成します。
4. 通常どおり予測を取得します。

## 欠落した値の処理

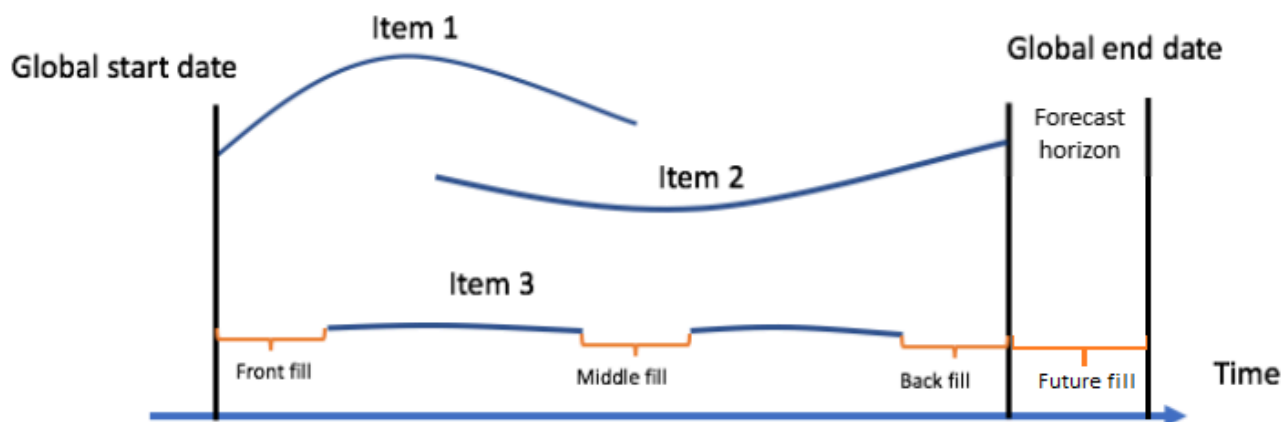
時系列予測データの一般的な問題は、欠落した値の存在です。データには、測定の失敗、フォーマットの問題、人為的ミス、記録する情報の不足など、さまざまな理由で欠落した値が含まれている可能性があります。たとえば、小売店の製品需要を予測しているときに、商品が売り切れたり使用不能になったりすると、その品目が在庫切れになっている間に記録する売上データは存在しないことになります。十分に優勢な場合、欠落した値がモデルの精度に影響を与える可能性があります。

Amazon Forecast には、ターゲット時系列および関連する時系列データセットの欠落した値を処理するための多数の filling メソッドが用意されています。フィルは、データセット内の欠落しているエントリに標準化された値を追加するプロセスです。

Forecast では、以下の filling メソッドがサポートされています。

- [Middle filling] (中間 filling) - データセットの項目の開始日と項目の終了日の間の欠落した値を埋めます。
- [Back filling] (後方 filling) - データセットの最後に記録されたデータポイントとデータセットのグローバル終了日との間の欠落した値を埋めます。
- [Future filling (related time series only)] (将来のフィル (関連する時系列のみ)) - グローバル終了日から予測期間の終了までの間に欠落した値を埋めます。

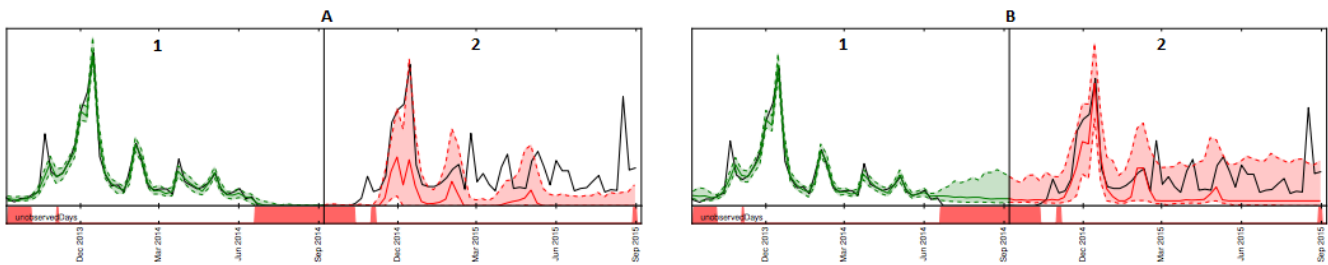
次の図は、さまざまなフィル方法を視覚的に示しています。



## フィルロジックの選択

フィルロジックを選択するときは、モデルによってロジックがどのように解釈されるかを考慮する必要があります。たとえば、小売のシナリオでは、利用可能な品目の販売数 0 を記録することは、利用できない品目の販売数 0 を記録することとは異なります。後者の場合、品目に対するお客様の関心がないことを意味するものではありません。このため、ターゲット時系列での 0 filling では、予測で予測子の偏りが不足することがありますが、NaN filling では、販売されている利用可能な品目 0 個の実際の出現が無視され、予測子の偏りが過剰になる可能性があります。

次の時系列グラフは、誤ったフィル値を選択するとモデルの精度に大きく影響することを示しています。グラフ A と B は、部分的にである項目の需要をプロットし out-of-stock、黒い線は実際の売上データを表します。A1 の欠落した値は 0 で埋められ、A2 では比較的偏りのない予測 (点線で表される) につながります。同様に、B1 の欠落した値は NaN で埋められ、B2 のより正確な予測につながります。



サポートされているフィルロジックのリストについては、次のセクションを参照してください。

## ターゲット時系列および関連する時系列フィルロジック

ターゲット時系列データセットと関連する時系列データセットの両方に対してフィルを実行できます。データセットタイプごとに異なる入力ガイドラインと制限があります。

### フィルガイドライン

データセットタイプ	デフォルトでフィルしますか?	サポートされているフィル方法	デフォルトのフィルロジック	受け入れられた filling ロジック
ターゲット時系列	はい	中間フィルと後方フィル	0	<ul style="list-style-type: none"> <li>• zero - 0 filling。</li> <li>• value - 整数または浮動小数。</li> <li>• nan - 数字ではありません。</li> <li>• mean - データ系列の平均値。</li> <li>• median - データ系列の中央値。</li> <li>• min - データ系列の最小値。</li> <li>• max - データ系列の最大値。</li> </ul>

データセットタイプ	デフォルトでフィルしますか?	サポートされているフィル方法	デフォルトのフィルロジック	受け入れられた filling ロジック
関連する時系列	いいえ	中間、後方、および将来のフィル	デフォルトなし	<ul style="list-style-type: none"> <li>• zero - 0 filling。</li> <li>• value - 整数または浮動小数点値。</li> <li>• mean - データ系列の平均値。</li> <li>• median - データ系列の中央値。</li> <li>• min - データ系列の最小値。</li> <li>• max - データ系列の最大値。</li> </ul>

### ⚠ Important

ターゲットおよび関連する時系列データセットの両方について、mean、median、min、および max は、欠落した値より前の最新の 64 個のデータエントリのローリングウィンドウに基づいて計算されます。

## 欠落した値の構文

欠損値の filling を実行するには、[CreatePredictor](#) オペレーションを呼び出すときに実装する filling のタイプを指定します。塗りつぶしロジックは [FeaturizationMethod](#) オブジェクトで指定されます。

次の抜粋は、ターゲット時系列属性および関連する時系列属性 (target\_value および price) に対して正しくフォーマットされた FeaturizationMethod オブジェクトを示しています。

filling メソッドを特定の値に設定するには、fill パラメータを value に設定し、対応する \_value パラメータで値を定義します。以下に示すように、関連する時系列の backfilling は、"backfill": "value" および "backfill\_value": "2" とともに値 2 に設定されます。

```
[
  {
    "AttributeName": "target_value",
    "FeaturizationPipeline": [
      {
        "FeaturizationMethodName": "filling",
        "FeaturizationMethodParameters": {
          "aggregation": "sum",
          "middlefill": "zero",
          "backfill": "zero"
        }
      }
    ]
  },
  {
    "AttributeName": "price",
    "FeaturizationPipeline": [
      {
        "FeaturizationMethodName": "filling",
        "FeaturizationMethodParameters": {
          "middlefill": "median",
          "backfill": "value",
          "backfill_value": "2",
          "futurefill": "max"
        }
      }
    ]
  }
]
```

## Forecast のデータセットのガイドライン

Amazon Forecast がデータセットのインポートに失敗する場合、またはデータセットが期待どおりに機能しない場合は、次のガイドラインを参照してください。

## タイムスタンプ形式

年 (Y)、月 (M)、週 (W)、および日 (D) の収集頻度の場合、Forecast は yyyy-MM-dd タイムスタンプ形式 (例: 2019-08-21) と、オプションで HH:mm:ss 形式 (例: 2019-08-21 15:00:00) をサポートします。

時 (H) と分 (M) の頻度では、Forecast は yyyy-MM-dd HH:mm:ss 形式 (例: 2019-08-21 15:00:00) のみをサポートしています。

ガイドライン: データセットの収集頻度のタイムスタンプ形式をサポートされている形式に変更します。

## Amazon S3 ファイルまたはバケット

データセットをインポートするとき、データを含む Amazon Simple Storage Service (Amazon S3) バケットの CSV または Parquet ファイルへのパス、またはデータを含む S3 バケットの名前を指定できます。CSV ファイルまたは Parquet ファイルを指定すると、Forecast はそのファイルだけをインポートします。S3 バケットを指定すると、Forecast はバケット内のすべての CSV または Parquet ファイルを最大 10,000 ファイルをインポートします。バケット名を指定して複数のファイルをインポートする場合は、すべての CSV または Parquet ファイルが、指定されたスキーマに準拠する必要があります。

ガイドライン: 次の構文を使用して、特定のファイルまたは S3 バケットを指定します。

```
s3://bucket-name/example-object.csv
```

```
s3://bucket-name/example-object.parquet
```

```
s3://bucket-name/prefix/
```

```
s3://bucket-name
```

Parquet ファイルには、.parquet、.parq、.pqt という拡張子を付けることもできますが、拡張子をまったく付けなくてもかまいません。

## データセットの完全更新

最初のデータセットのインポートは常に完全インポートで、それ以降のインポートは完全更新でも増分更新でもかまいません。Forecast API を使用してインポートモードを指定する必要があります。

完全更新では、既存のデータはすべて新しくインポートされたデータに置き換えられます。データセットの完全インポートジョブは集計されないため、最新のデータセットインポートは、予測子のトレーニング時や予想の生成時に使用されるものです。



ガイドライン: データセットを段階的に更新して、新しいデータを既存のデータに追加します。そうしない場合、最新のデータセットインポートに、前回のインポート以降に収集された新しいデータだけでなく、モデル化するすべてのデータが含まれていることを確認します。

### データセットの増分更新

タイムスタンプ、データ形式、位置情報などのフィールドは、現在アクティブなデータセットから読み取られます。この情報をデータセットの増分インポートに含める必要はありません。これらが含まれる場合は、元々提供されていた値と一致する必要があります。

ガイドライン: これらの値を変更する場合は、データセット全体をインポートしてください。

### 属性の順序

スキーマ定義で指定された属性の順序は、インポートする CSV または Parquet ファイルの列の順序と一致する必要があります。たとえば、最初の属性として timestamp を定義した場合、timestamp は入力ファイルの最初の列でもある必要があります。

ガイドライン: 入力ファイルの列が、作成したスキーマ属性と同じ順序であることを確認してください。

### Weather Index

Weather Index を適用するには、ターゲットの時系列および関連する時系列データセットに [geolocation 属性](#) を含める必要があります。また、ターゲットの時系列タイムスタンプの [タイムゾーン](#) を指定する必要があります。

ガイドライン: データセットに geolocation 属性が含まれており、タイムスタンプにタイムゾーンが割り当てられているようにしてください。詳細については、Weather Index の「[Conditions and Restrictions](#)」(条件と制約事項) を参照してください。

### データセットヘッダー

入力 CSV のデータセットヘッダーにより、検証エラーが発生する場合があります。CSV ファイルではヘッダーを省略することをお勧めします。

ガイドライン: データセットヘッダーを削除し、インポートを再試行します。

Parquet ファイルにはデータセットヘッダーが必要です。

### データセットのステータス

[the section called “CreateDatasetImportJob”](#) オペレーションでトレーニングデータをインポートするには、データセットの Status が ACTIVE である必要があります。

ガイドライン: [DescribeDataset](#) オペレーションを使用して、データセットのステータスを取得します。データセットの作成または更新に失敗した場合は、データセットファイルの形式を確認して、もう一度作成してください。

## デフォルトファイル形式

デフォルトのファイル形式は CSV です。

## ファイル形式と区切り記号

Forecast では、カンマ区切り値 (CSV) ファイル形式と Parquet 形式のみがサポートされています。タブ、スペース、コロン、またはその他の文字を使用して値を区切ることはできません。

ガイドライン: データセットを CSV 形式 (区切り文字としてカンマのみを使用) または Parquet 形式に変換し、ファイルのインポートを再試行します。

## ファイル名

ファイル名には、アルファベットを 1 文字以上含める必要があります。名前が数値のみのファイルはインポートできません。

ガイドライン: 少なくともアルファベットが 1 文字以上含まれるように入力データファイルの名前を変更し、ファイルのインポートを再試行します。

## パーティショニングされた Parquet データ

Forecast は、パーティショニングされた Parquet ファイルを読み取りません。

## What-If 分析のデータセット要件

What-If 分析には CSV データセットが必要です。[CreateWhatIfAnalysis](#) アクションの `TimeSeriesSelector` オペレーションと `TimeSeriesReplacementDataSource` オペレーションは、Parquet ファイルを受け入れ [CreateWhatIfForecast](#) しません。

# 予測子のトレーニング

予測子は、ターゲットの時系列、関連する時系列、アイテムメタデータ、および含める追加のデータセットを使用してトレーニングされた Amazon Forecast のモデルです。予測子を使用して、時系列データに基づいて予測を生成できます。

デフォルトでは、Amazon Forecast は を作成し AutoPredictor、Forecast はデータセット内の各時系列にアルゴリズムの最適な組み合わせを適用します。

## トピック

- [予測子の作成](#)
- [へのアップグレード AutoPredictor](#)
- [さまざまな予想頻度のデータ集計](#)
- [追加のデータセットの使用](#)
- [レガシー予測子の使用](#)
- [予測精度の評価](#)
- [予測子の再トレーニング](#)
- [Weather Index](#)
- [Holidays の特徴量化](#)
- [Predictor Explainability](#)
- [予測子のモニタリング](#)
- [Amazon Forecast アルゴリズム](#)

## 予測子の作成

Amazon Forecast は、予測子をトレーニングするために次の入力が必要とします。

- データセットグループ – ターゲット時系列データセットを含める必要があるデータセットグループ。ターゲット時系列データセットには、ターゲット属性 (item\_id) とタイムスタンプ属性、および任意のディメンションが含まれます。関連する時系列とアイテムメタデータはオプションです。詳細については、「[データセットのインポート](#)」を参照してください。
- 予測頻度 – 予測の粒度 (毎時、毎日、毎週など)。Amazon Forecast では、頻度の単位と値を指定することで、予想の正確な粒度を判断できます。整数値のみ指定できます。

頻度の単位	許可される値
分	1-59
毎時	1-23
1日1回	1-6
毎週	1-4
月別	1-11
年次	1

例えば、隔週の予想が必要な場合、頻度の単位は週に、値は 2 です。または、四半期ごとの予想が必要な場合は、頻度の単位は 1 か月で、値は 3 です。

予想頻度よりも高い頻度でデータが収集されると、そのデータは予想頻度に集約されます。これには、直近の時系列と関連する時系列データが含まれます。集計詳細については、「[さまざまな予想頻度のデータ集計](#)」を参照してください。

- 予測期間 – 予測されるタイムステップの数。

次のオプション入力の値を設定することもできます。

- 時間調整境界 — Forecast がデータを集約し、指定した予想頻度に合わせた予想を生成するために使用する時間境界です。集計詳細については、「[さまざまな予想頻度のデータ集計](#)」を参照してください。時間境界の指定について詳しくは、「[時間境界](#)」を参照してください。
- 予測ディメンション – ディメンションは、ターゲット時系列データセットのオプションの属性であり、ターゲット値 (item\_id) と組み合わせて使用して個別の時系列を作成できます。
- 予測タイプ – 予測子の評価に使用される分位数。
- 最適化メトリクス – 予測子を最適化するために使用される精度メトリクス。
- 追加のデータセット – Weather Index や Holidays などのビルトインの Amazon Forecast データセット。

ソフトウェア開発キット (SDK) または Amazon Forecast コンソールを使用して、予測子を作成できます。

## Console

予測子を作成するには

1. AWS Management Console にサインインして、Amazon Forecast コンソール (<https://console.aws.amazon.com/forecast/>) を開きます。
2. [Dataset groups] (データセットグループ) から、データセットグループを選択します。
3. ナビゲーションペインで、[Predictors] (予測子) を選択します。
4. [Train new predictor] (新しい予測子のトレーニング) を選択します。
5. 以下の必須フィールドに値を入力します。
  - 名前 - 一意の予測子名。
  - 予測頻度 - 予測の粒度。
  - [Forecast horizon] (予測期間) - 予測するタイムステップの数。
6. [開始] を選択します。

追加のデータセットについては、[the section called “Weather Index”](#) および [the section called “Holidays の特徴量化”](#) を参照してください。予測タイプと最適化メトリクスのカスタマイズの詳細については、「[the section called “予測子のメトリクス”](#)」を参照してください。

## AWS CLI

AWS CLI を使用して auto 予測子を作成するには、`create-predictor` コマンドを使用します。次のコードでは、今後 14 日の予測を行う自動予測子を作成します。

予測子の名前と、トレーニングデータを含むデータセットグループの Amazon リソースネーム (ARN) を指定します。必要に応じて、予測期間と予想頻度を変更します。必要に応じて、予測子に任意のタグを追加します。詳細については、[Amazon Forecast のリソースのタグ付け](#)を参照してください。

必須とオプションのパラメータの詳細については、「[CreateAutoPredictor](#)」を参照してください。

```
aws forecast create-predictor \  
--predictor-name predictor_name \  

```

```
--data-config DatasetGroupArn="arn:aws:forecast:region:account:dataset-  
group/datasetGroupName" \  
--forecast-horizon 14 \  
--forecast-frequency D \  
--tags Key=key1,Value=value1 Key=key2,Value=value2
```

予想タイプと最適化メトリクスのカスタマイズの詳細については、「[the section called “予測子のメトリクス”](#)」を参照してください。DataConfig データ型内で Weather Index および Holidays の追加のデータセットが定義されています。追加のデータセットについては、[the section called “Weather Index”](#) および [the section called “Holidays の特徴量化”](#) を参照してください。

## Python

SDK for Python (Boto3) を使用して自動予測子を作成するには、`create_auto_predictor` メソッドを使用します。次のコードでは、今後 14 日の予測を行う自動予測子を作成します。

予測子の名前と、トレーニングデータを含むデータセットグループの Amazon リソースネーム (ARN) を指定します。必要に応じて、予測期間と予想頻度を変更します。必要に応じて、予測子に任意のタグを追加します。詳細については、[Amazon Forecast のリソースのタグ付け](#)を参照してください。

必須とオプションのパラメータの詳細については、「[CreateAutoPredictor](#)」を参照してください。

```
import boto3  
  
forecast = boto3.client('forecast')  
  
create_predictor_response = forecast.create_auto_predictor(  
    PredictorName = 'predictor_name',  
    ForecastHorizon = 14,  
    ForecastFrequency = 'D',  
    DataConfig = {  
        "DatasetGroupArn": "arn:aws:forecast:region:account:dataset-  
group/datasetGroupName"  
    },  
    Tags = [  
        {  
            "Key": "key1",  
            "Value": "value1"  
        },  
        {
```

```
        "Key": "key2",
        "Value": "value2"
    }
]
)
print(create_predictor_response['PredictorArn'])
```

予想タイプと最適化メトリクスのカスタマイズの詳細については、「[the section called “予測子のメトリクス”](#)」を参照してください。DataConfig データ型内で Weather Index および Holidays の追加のデータセットが定義されています。追加のデータセットについては、[the section called “Weather Index”](#) および [the section called “Holidays の特徴量化”](#) を参照してください。

## へのアップグレード AutoPredictor

### Python ノートブック

予測子を にアップグレードする step-by-step ガイドについては AutoPredictor、「[予測子を にアップグレードする AutoPredictor](#)」を参照してください。

AutoML または手動選択 (CreatePredictor) で作成された予測子は、 にアップグレードできます AutoPredictor。既存の を にアップグレードすると、関連するすべての予測子構成設定が転送 AutoPredictor されます。

にアップグレードした後 AutoPredictor、元の予測子はアクティブなままになり、アップグレードされた予測子には個別の予測子 ARN があります。これにより、2 つの予測子間の精度メトリクスを比較でき、元の予測子を使用して予測を生成できます。

ソフトウェア開発キット (SDK) または Amazon Forecast コンソールを使用して、予測子をアップグレードできます。

### Console

予測子をアップグレードするには

1. AWS Management Console にサインインして、Amazon Forecast コンソール (<https://console.aws.amazon.com/forecast/>) を開きます。
2. ナビゲーションペインで、[Predictors] (予測子) を選択します。

3. アップグレードする予測子を選択し、[Upgrade] (アップグレード) を選択します。
4. アップグレードされた予測子の一意の名前を設定します。
5. にアップグレード AutoPredictorを選択します。

## CLI

AWS CLI を使用して予測子をアップグレードするには、`create-predictor` メソッドを使用します。ただし、予測子の名前と `reference-predictor-arn` の値 (アップグレードする予測子の ARN) のみを指定します。

```
aws forecast create-predictor \  
--predictor-name predictor_name \  
--reference-predictor-arn arn:aws:forecast:region:account:predictor/predictorName
```

## Python

SDK for Python (Boto3) を使用して予測子をアップグレードするには、`create_auto_predictor` メソッドを使用します。ただし、予測子の名前と `ReferencePredictorArn` の値 (アップグレードする予測子の ARN) のみを指定します。

```
import boto3  
  
forecast = boto3.client('forecast')  
  
create_predictor_response = forecast.create_auto_predictor(  
    PredictorName = 'predictor_name',  
    ReferencePredictorArn =  
    'arn:aws:forecast:region:account:predictor/predictorName'  
)  
print(create_predictor_response['PredictorArn'])
```

## さまざまな予想頻度のデータ集計

予測子を作成するときは、予想頻度を指定する必要があります。予想頻度によって、予想における予測頻度が決まります。例えば、月次売上予想などです。Amazon Forecast の予測子は、指定した予想頻度よりも高いデータ頻度の予想を生成できます。例えば、データが毎日記録されている場合でも、週ごとの予想を生成できます。トレーニング中、Forecast は日次データを集計し、週ごとの予想頻度で予想を生成します。



## トピック

- [集計の仕組み](#)
- [時間境界](#)
- [データ集計の想定](#)

## 集計の仕組み

トレーニング中、Amazon Forecast は、指定した予想頻度と一致しないデータをすべて集計します。例えば、日次データはあるが、週次の予想頻度を指定しているとします。Forecast は、そのデータが属する週に基づいて日次データを調整します。次に、Forecast はそれを週ごとに 1 つのレコードにまとめます。Forecast は、時間境界との関係に基づいて、データがどの週 (または月、日など) に属するかを判断します。時間境界は、1 日の開始時間や週の開始日など、時間単位の開始点を指定します。

時間単位や分単位の予想、または時間境界が不明な場合、Forecast は頻度の時間単位に基づくデフォルトの時間境界を使用します。日次、週次、月次、または年次の予想頻度となる自動予測子の場合、カスタムの時間境界を指定できます。時間境界の詳細については、「[時間境界](#)」を参照してください

集計時のデフォルトの変換方法は、データを合計することです。予測子の作成時に変換を設定できます。これは、Forecast コンソールの [予測子を作成] ページの [入力データ設定] セクションで行います。または、CreateAutoPredictor オペレーションの [AttributeConfig](#) の Transformations パラメータで変換メソッドを設定することもできます。

次の表は、デフォルトの時間境界を使用した 1 時間ごとの予測頻度の集計例を示しています。各時間は正時から始まります。

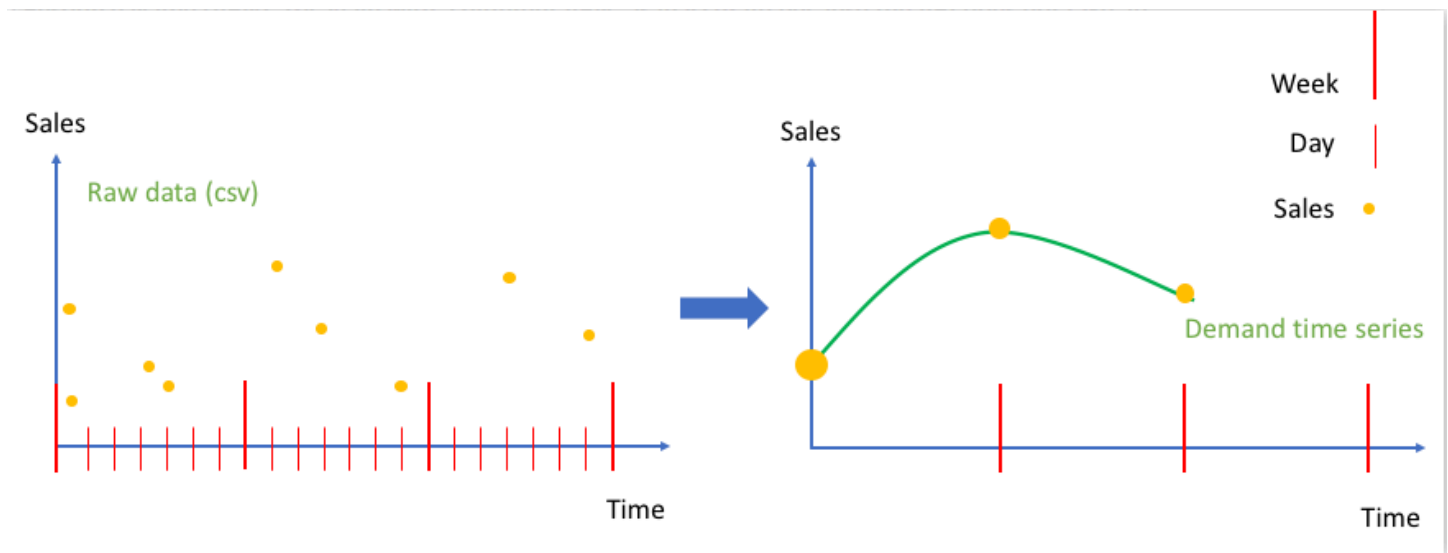
### 変換前

時間	データ	時間の冒頭
2018-03-03 01:00:00	100	Yes
2018-03-03 02:20:00	50	No
2018-03-03 02:45:00	20	いいえ
2018-03-03 04:00:00	120	Yes

## 変換後

時間	データ	メモ
2018-03-03 01:00:00	100	
2018-03-03 02:00:00	70	02:00:00 ~ 02:59:59 (50 + 20) の間の値の合計
2018-03-03 03:00:00	空	03:00:00 ~ 03:59:59 の間に値はありません
2018-03-03 04:00:00	120	

次の図は、Forecast がデフォルトの毎週の時間境界に合うようにデータを変換する方法を示しています。



## 時間境界

時間境界は、週の開始日など、時間単位の開始点を指定します。データを集計する前に、Amazon Forecast は予想頻度の時間単位に基づいてデータを調整します。これは、データと時間的境界との関係に基づいて行われます。

例えば、日単位の予想頻度を指定し、独自の時間境界は指定しない場合、Forecast は各時間ごとのレコードをその日付に基づいて調整します。1日の開始は0時間です。1日の開始(0時間)の定義が時間境界です。次に、Forecast は時間単位のレコードをその日の1つのレコードに集約します。

Forecast は、予想頻度の時間単位に基づくデフォルトの時間境界を使用します。自動予測子を作成する場合、カスタムの時間境界を指定できます。

カスタム時間境界とカスタム予想頻度の両方を指定すると、Forecast は予想頻度内のデータを集計し、カスタム時間境界に合わせます。予想頻度によってデータが集計される頻度が決まり、カスタム時間境界によって調整が行われる場所が決まります。例えば、データを毎日収集していて、Amazon Forecast でその月の 15 日に 1 年間の四半期予想を生成させたいとします。そのためには、予想頻度を 3 か月ごとに、カスタム期間を 15 に設定します。次の AWS Command Line Interface の例を参照してください。

```
aws forecast create-predictor \
--predictor-name predictor_name \
--data-config DatasetGroupArn="arn:aws:forecast:region:account:dataset-
group/datasetGroupName" \
--forecast-horizon 4 \
--forecast-frequency 3M \
--time-alignment-boundary DayOfMonth=15
```

この例では、すべての日次データが 3 か月ごとの 15 日に合計されます (デフォルトの集計)。

この集計では日次データは必要なく、データが毎月またはそれ以上の頻度で収集されることだけであることに注意してください。

## トピック

- [デフォルトの時間境界](#)
- [時間境界の指定](#)

## デフォルトの時間境界

次の表は、Forecast がデータを集計するときに使用するデフォルトの時間調整の境界を示しています。

頻度	境界
分	分の末尾 (45:00、06:00)
時間	時間の末尾 (09:00:00、13:00:00)
日	1 日の最初の 1 時間 (0 時間)

頻度	境界
週	直近の月曜日
月	毎月 1 日
年	年始 (1 月 1 日)

## 時間境界の指定

### Note

自動予測子には時間境界のみを指定できます。

日次、週次、月次、または年次の予測頻度で自動予測子を作成する場合、Forecast がデータを集計するために使用する時間境界を指定できます。業務カレンダーがデフォルトの時間境界と一致しない場合は、時間境界を指定できます。例えば、各月にその 3 日目から始まる月次予想を生成したい場合があります。時間境界を指定しない場合、Forecast は [デフォルトの時間境界](#) のセットを使用します。

指定する時間境界単位は、予想頻度より 1 単位小さくなければなりません。次の表は、指定できる時間境界の単位と値を予想頻度別にまとめたものです。

境界値が 28 以下の Monthly 時間境界のみを指定できます。

予想頻度単位	境界の単位	境界の値
1 日 1 回	時間	0 ~ 23
毎週	曜日	月曜日から日曜日
月別	日	1 から 28
年次	月	1 月から 12 月

予測子を作成するときは、次のように時間調整の境界を指定します。プログラムで指定できるさまざまな時間境界単位と境界値については、「[TimeAlignmentBoundary](#)」を参照してください。

## Console

予測子の時間調整境界を指定するには

1. AWS Management Console にサインインして、Amazon Forecast コンソール (<https://console.aws.amazon.com/forecast/>) を開きます。
2. [Dataset groups] (データセットグループ) から、データセットグループを選択します。
3. ナビゲーションペインで、[Predictors] (予測子) を選択します。
4. [Train new predictor] (新しい予測子のトレーニング) を選択します。
5. 必須の [名前]、[予想頻度]、および [予想] フィールドに値を入力します。
6. [時間調整境界] には、予測子がデータを集計するときに使用する時間境界を指定します。このリストの値は、選択した予想頻度によって異なります。
7. [Start] (開始) を選択します。Forecast は、予測子の作成時に指定した時間調整境界を使用してデータを集計します。

## AWS CLI

AWS CLI で予測子の時間調整境界を指定するには、`create-predictor` コマンドを使用します。`time-alignment-boundary` パラメータには、時間単位と境界値を指定します。次のコードは、毎週火曜日から始まる先の 5 週間を予測する自動予測子を作成します。

`DayOfWeek` と `DayOfMonth` 値はすべて大文字でなければなりません。指定できるさまざまな時間境界単位と境界値については、「[TimeAlignmentBoundary](#)」を参照してください。必須とオプションのパラメータの詳細については、「[CreateAutoPredictor](#)」を参照してください。

```
aws forecast create-predictor \  
--predictor-name predictor_name \  
--data-config DatasetGroupArn="arn:aws:forecast:region:account:dataset-  
group/datasetGroupName" \  
--forecast-horizon 5 \  
--forecast-frequency W \  
--time-alignment-boundary DayOfWeek=TUESDAY
```

## Python

SDK for Python (Boto3) で予測子のタイムアライメント境界を指定するには、`create_auto_predictor` メソッドを使用します。`TimeAlignmentBoundary` パラメー

タには、時間単位をキー、境界値を値とするディクショナリを指定します。次のコードは、毎週火曜日から始まる先の 5 週間を予測する自動予測子を作成します。

DayOfWeek と DayOfMonth 値はすべて大文字でなければなりません。指定できるさまざまな時間境界単位と境界値については、「[TimeAlignmentBoundary](#)」を参照してください。必須とオプションのパラメータの詳細については、「[CreateAutoPredictor](#)」を参照してください。

```
import boto3

forecast = boto3.client('forecast')

create_predictor_response = forecast.create_auto_predictor(
    PredictorName = 'predictor_name',
    ForecastHorizon = 5,
    ForecastFrequency = 'W',
    DataConfig = {
        "DatasetGroupArn": "arn:aws:forecast:region:account:dataset-
group/datasetGroupName"
    },
    TimeAlignmentBoundary = {
        "DayOfWeek": "TUESDAY"
    }
)
print(create_predictor_response['PredictorArn'])
```

## データ集計の想定

Forecast は、データが特定のタイムゾーンのものであるとは想定していません。ただし、時系列データを集計するときは、次の前提になります。

- すべてのデータは同じタイムゾーンからのものです。
- すべての予想は、データセットのデータと同じタイムゾーンにあります。
- [the section called “CreatePredictor”](#) オペレーションの [the section called “InputDataConfig”](#) パラメータで [the section called “SupplementaryFeature”](#) 休日機能を指定した場合、入力データは同じ国のものです。

## 追加のデータセットの使用

Amazon Forecast には、予測子を作成するときに Weather Index と Holidays を含めることができます。Weather Index は気象情報をモデルに組み込み、Holidays は祝祭日に関する情報を組み込みます。

Weather Index には、ターゲット時系列データセットの「geolocation」属性と、タイムスタンプのタイムゾーンに関する情報が必要です。詳細については、「[the section called “Weather Index”](#)」を参照してください。

Holidays には、250 か国を超える祝祭日情報が含まれています。詳細については、「[the section called “Holidays の特徴量化”](#)」を参照してください。

## レガシー予測子の使用

### Note

既存の予測子を にアップグレードするには AutoPredictor、「」を参照してください。[the section called “へのアップグレード AutoPredictor”](#)

AutoPredictor は、Amazon Forecast を使用して予測子を作成するデフォルトの推奨方法です。は、データセット内の各時系列に最適なアルゴリズムの組み合わせを適用することで予測子 AutoPredictor を作成します。

で作成された予測子 AutoPredictor は、AutoML または手動選択で作成された予測子よりも一般的に正確です。Forecast Explainability および予測子の再トレーニング機能は、で作成された予測子でのみ使用できます AutoPredictor。

Amazon Forecast は、次の方法でレガシー予測子を作成することもできます。

1. AutoML - Forecast は、最もパフォーマンスの高いアルゴリズムを見つけて、それをデータセット全体に適用します。
2. 手動選択 - データセット全体に適用される単一のアルゴリズムを手動で選択します。

Software Development Kit (SDK) を使用して、レガシー予測子を作成できる場合があります。

## SDK

AutoML を使用するには

[CreatePredictor](#) オペレーションを使用して、PerformAutoML の値を "true" に設定します。

```
{
  ...
  "PerformAutoML": "true",
}
```

AutoML を使用する場合は、CreatePredictorAlgorithmArn、 のパラメータに値を設定することはできませんHP0ConfigTrainingParameters。

## 予測精度の評価

Amazon Forecast は、予測子を評価し、予測の生成に使用する項目を選択するのに役立つ精度メトリクスを生成します。Amazon Forecast は、二乗平均平方根誤差 (RMSE)、重み付き分位損失 (wQL)、平均絶対パーセント誤差 (MAPE)、平均絶対スケーリング誤差 (MASE)、および重み付き絶対誤差率 (WAPE) メトリクスを提供します。

Amazon Forecast は、バックテストを使用してパラメータをチューニングし、精度メトリクスを生成します。バックテスト中、Forecast は、時系列データをトレーニングセットとテストセットといたった 2 つのセットに自動的に分割します。トレーニングセットは、モデルをトレーニングし、テストセット内のデータポイントの予想を生成するために使用されます。Forecast は、予測値をテストセットの観測値と比較することにより、モデルの精度を評価します。

Forecast を使用すると、さまざまな予測タイプを使用して予測子を評価できます。これは、分位数予測と平均予測のセットである場合があります。平均予測は点推定を提供しますが、分位数予測は通常、考えられる結果の範囲を提供します。

### Python ノートブック

予測子メトリクスの評価に関するステップバイステップガイドについては、「[Computing Metrics Using Item-level Backtests](#)」(項目レベルのバックテストを使用したメトリクスのコンピューティング) を参照してください。



## トピック

- [精度メトリクスの解釈](#)
- [重み付き分位損失 \(wQL\)](#)
- [重み付き絶対誤差率 \(WAPE\)](#)
- [二乗平均平方根誤差 \(RMSE\)](#)
- [平均絶対パーセント誤差 \(MAPE\)](#)
- [平均絶対スケーリング誤差 \(MASE\)](#)
- [精度メトリクスのエクスポート](#)
- [Forecast タイプの選択](#)
- [レガシー予測子の使用](#)

## 精度メトリクスの解釈

Amazon Forecast は、予測子を評価するために、二乗平均平方根誤差 (RMSE)、重み付き分位損失 (wQL)、平均重み付き分位損失 (平均 wQL)、平均絶対スケーリング誤差 (MASE)、平均絶対パーセント誤差 (MAPE)、および重み付き絶対誤差率 (WAPE) メトリクスを提供します。全体的な予測子のメトリクスとともに、Forecast は、各バックテストウィンドウのメトリクスを計算します。

Amazon Forecast のソフトウェア開発キット (SDK) および Amazon Forecast コンソールを使用して、予測子の精度メトリクスを表示できます。

### Forecast SDK

[GetAccuracyMetrics](#) オペレーションを使用して、各バックテストの RMSE、MASE、MAPE、WAPE、平均 wQL、および wQL のメトリクスを表示するように `PredictorArn` を指定します。

```
{
  "PredictorArn": "arn:aws:forecast:region:acct-id:predictor/example-id"
}
```

### Forecast Console

[Predictors] (予測子) のページで予測子を選択します。予測子の精度メトリクスは、[Predictor metrics] (予測子メトリクス) のセクションに表示されます。

**Note**

平均 wQL、wQL、RMSE、MASE、MAPE、および WAPE メトリクスの場合、値が小さいほど優れたモデルであることを示します。

## トピック

- [重み付き分位損失 \(wQL\)](#)
- [重み付き絶対誤差率 \(WAPE\)](#)
- [二乗平均平方根誤差 \(RMSE\)](#)
- [平均絶対パーセント誤差 \(MAPE\)](#)
- [平均絶対スケーリング誤差 \(MASE\)](#)
- [精度メトリクスのエクスポート](#)
- [Forecast タイプの選択](#)
- [レガシー予測子の使用](#)

## 重み付き分位損失 (wQL)

重み付き分位損失 (wQL) メトリクスは、指定された分位数でのモデルの精度を測定します。これは、過小予測と過大予測のコストが異なる場合に特に役立ちます。wQL 関数の重み ( $\tau$ ) を設定することにより、過小予測と過大予測に対して異なるペナルティを自動的に組み込むことができます。

損失関数は以下のように計算されます。

$$\text{wQL}[\tau] = 2 \frac{\sum_{i,t} [\tau \max(y_{i,t} - q_{i,t}^{(\tau)}, 0) + (1 - \tau) \max(q_{i,t}^{(\tau)} - y_{i,t}, 0)]}{\sum_{i,t} |y_{i,t}|}$$

実行する条件は以下のとおりです。

$\tau$  - セット {0.01, 0.02, ..., 0.99} 内の分位数

$q_{i,t}^{(\tau)}$  - モデルが予測する  $\tau$  分位数。

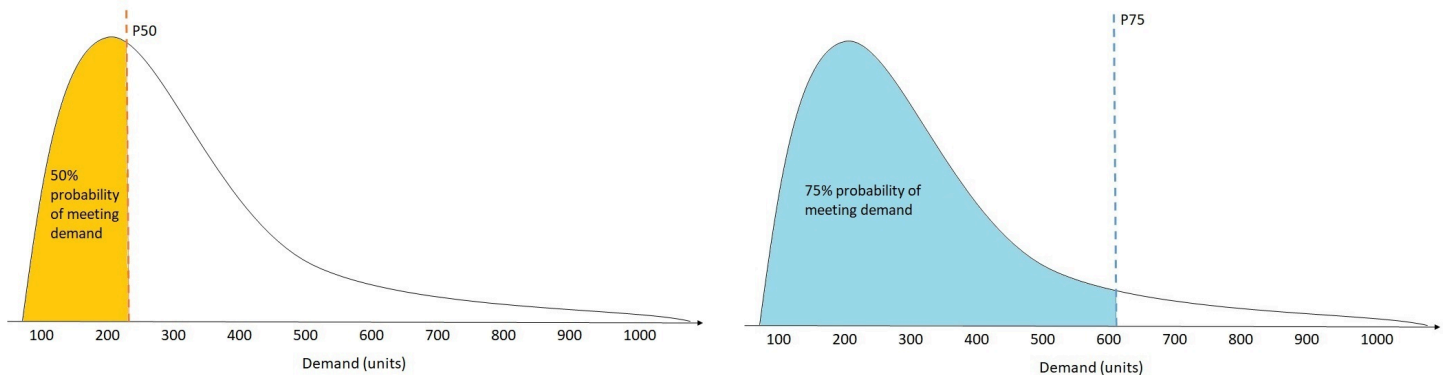
$y_{i,t}$  - ポイント (i,t) での観測値

wQL の分位数 ( $\tau$ ) は、0.01 (P1) から 0.99 (P99) の範囲です。wQL メトリクスは、平均予測を求めるために計算することはできません。

デフォルトでは、Forecast は、0.1 (P10)、0.5 (P50)、および 0.9 (P90) で wQL を計算します。

- P10 (0.1) - true の値は 10% の確率で予測値より低くなることが予想されます。
- P50 (0.5) - true の値は 50% の確率で予測値より低くなることが予想されます。これは、予測の中央値とも呼ばれます。
- P90 (0.9) - true の値は 90% の確率で予測値より低くなることが予想されます。

小売業では、在庫不足のコストは在庫過剰のコストよりも高いことが多いため、P75 ( $\tau = 0.75$ ) での予測は、分位数の中央値 (P50) での予測よりも有益な場合があります。これらの場合、wQL[0.75] は、より大きなペナルティの重みを過小予測 (0.75) に割り当て、より小さなペナルティの重みを過大予測 (0.25) に割り当てます。



上の図は、wQL[0.50] と wQL[0.75] でのさまざまな需要予測を示しています。P75 の予測値は P50 の予測値よりも大幅に高くなります。これは、P75 の予測が 75% の確率で需要を満たすと予想されるのに対し、P50 の予測は 50% の確率で需要を満たすと予想されるためです。

指定のバックテスト期間では、すべての項目と時間ポイントの観察された値の合計がほぼゼロの場合、重み付き分位損失の式は未定義になります。これらの場合、Forecast は重み付けされていない分位損失を出力します。これは、wQL 式の分子です。

Forecast は、指定されたすべての分位数にわたる重み付き分位損失の平均値である平均 wQL も計算します。デフォルトでは、これは wQL[0.10]、wQL[0.50]、および wQL[0.90] の平均になります。

## 重み付き絶対誤差率 (WAPE)

重み付き絶対誤差率 (WAPE) は、観測値からの予測値の全体的な偏差を測定します。WAPE は、観測値の合計と予測値の合計を取り、これら 2 つの値の間の誤差を計算することによって計算されます。値が小さいほど、モデルの精度が高くなります。

指定のバックテスト期間では、すべての時間ポイントとすべての項目の観察された値の合計がほぼゼロの場合、重み付き絶対パーセント誤差の式は未定義になります。これらの場合、Forecast は重み付けされていない絶対誤差の合計を出力します。これは、WAPE 式の分子です。

$$\text{WAPE} = \frac{\sum_{i,t} |y_{i,t} - \hat{y}_{i,t}|}{\sum_{i,t} |y_{i,t}|}$$

実行する条件は以下のとおりです。

$y_{i,t}$  - ポイント (i,t) での観測値

$\hat{y}_{i,t}$  - ポイント (i,t) での予測値

Forecast は、平均予測を予測値  $\hat{y}_{i,t}$  として使用します。

WAPE は、二乗誤差の代わりに絶対誤差を使用するため、二乗平均平方根誤差 (RMSE) よりも外れ値に対してロバストです。

Amazon Forecast は、これまで、WAPE メトリクスを平均絶対パーセント誤差 (MAPE) と呼び、予測の中央値 (P50) を予測値として使用していました。Forecast は、平均予測を使用して WAPE を計算するようになりました。以下に示すように、wQL[0.5] メトリクスは WAPE[median] メトリクスと同等です。

$$\text{wQL}[0.5] = 2 \frac{\sum_{i,t} 0.5 [\max(y_{i,t} - q_{i,t}^{(0.5)}, 0) + \max(q_{i,t}^{(0.5)} - y_{i,t}, 0)]}{\sum_{i,t} |y_{i,t}|} = \frac{\sum_{i,t} |y_{i,t} - q_{i,t}^{(0.5)}|}{\sum_{i,t} |y_{i,t}|}$$

## 二乗平均平方根誤差 (RMSE)

二乗平均平方根誤差 (RMSE) は、二乗誤差の平均の平方根であるため、他の精度メトリクスよりも外れ値の影響を受けやすくなります。値が小さいほど、モデルの精度が高くなります。

$$\text{RMSE} = \sqrt{\frac{1}{nT} \sum_{i,t} (\hat{y}_{i,t} - y_{i,t})^2},$$
$$i = 1, \dots, n$$
$$t = 1, \dots, T$$

実行する条件は以下のとおりです。

$y_{i,t}$  - ポイント (i,t) での観測値

$\hat{y}_{i,t}$  - ポイント (i,t) での予測値

$nT$  - テストセット内のデータポイントの数

Forecast は、平均予測を予測値  $\hat{y}_{i,t}$  として使用します。予測子メトリクスを計算する場合、 $nT$  はバックテストウィンドウ内のデータポイントの数です。

RMSE は、外れ値の影響を増幅する残差の 2 乗値を使用します。非常に大きなコストを発生させる可能性があるのがいくつかの大きな誤予測だけであるユースケースでは、RMSE がより適切なメトリクスです。

2020 年 11 月 11 日より前に作成された予測子は、デフォルトで 0.5 分位数 (P50) を使用して RMSE を計算していました。Forecast は、平均予測を使用するようになりました。

## 平均絶対パーセント誤差 (MAPE)

平均絶対パーセント誤差 (MAPE) は、各時間単位の観測値と予測値の間のパーセント誤差の絶対値を取得し、それらの値を平均します。値が小さいほど、モデルの精度が高くなります。

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right|$$

実行する条件は以下のとおりです。

$A_t$  - ポイント  $t$  での観測値

$F_t$  - ポイント  $t$  での予測値

$n$  - 時系列のデータポイントの数

Forecast は、平均予測を予測値  $F_t$  として使用します。

MAPE は、値が時間ポイント間で大幅に異なり、外れ値が大きな影響を及ぼす場合に役立ちます。

## 平均絶対スケーリング誤差 (MASE)

平均絶対スケーリング誤差 (MASE) は、平均誤差をスケーリング係数で除することによって計算されます。このスケーリング係数は、予測頻度に基づいて選択された季節性の値  $m$  に依拠します。値が小さいほど、モデルの精度が高くなります。

$$MASE = \text{mean} \left( \frac{|e_j|}{\frac{1}{T-m} \sum_{t=m+1}^T |Y_t - Y_{t-m}|} \right) = \frac{\frac{1}{j} \sum_j |e_j|}{\frac{1}{T-m} \sum_{t=m+1}^T |Y_t - Y_{t-m}|}$$

実行する条件は以下のとおりです。

$Y_t$  - ポイント  $t$  での観測値

$Y_{t-m}$  - ポイント  $t-m$  での観測値

$e_j$  - 点  $j$  での誤差 (観測値 - 予測値)

$m$  - 季節性の値

Forecast は、平均予測を予測値として使用します。

MASE は、本質的に循環的であるか、季節的な特性を持つデータセットに最適です。例えば、夏季には需要が高く、冬季には需要が少ない商品の予測は、季節的な影響を考慮に入れることでより正確なものとなり得ます。

## 精度メトリクスのエクスポート

### Note

エクスポートファイルは、[Dataset Import] (データセットのインポート) から直接情報を返すことができます。これにより、インポートされたデータに式またはコマンドが含まれている場合、ファイルは CSV インジェクションに対して脆弱になります。このため、エクスポートされたファイルはセキュリティ警告を表示する可能性があります。悪意のあるアクティビティを回避するには、エクスポートされたファイルを読み取るときにリンクとマクロを無効にします。

Forecast を使用すると、バックテスト中に生成された予測値と精度メトリクスをエクスポートできます。

これらのエクスポートを使用して、特定の時間ポイントおよび分位数で特定の項目を評価し、予測子をより良く理解できます。バックテストエクスポートは、指定された S3 の場所に送信され、2 つのフォルダを含みます。

- `forecasted-values`: 各バックテストの各予想タイプでの予想値を含む CSV または Parquet ファイルが含まれています。
- `accuracy-metrics-values`: すべてのバックテストの平均とともに、各バックテストのメトリクスを含む CSV または Parquet ファイルが含まれています。これらのメトリクスには、各分位数の wQL、平均 wQL、RMSE、MASE、MAPE、および WAPE が含まれます。

`forecasted-values` フォルダには、各バックテストウィンドウの各予測タイプの予測値が含まれています。また、項目 ID、ディメンション、タイムスタンプ、ターゲット値、およびバックテストウィンドウの開始時間と終了時間に関する情報も含まれています。

`accuracy-metrics-values` フォルダには、各バックテストウィンドウの精度メトリクスと、すべてのバックテストウィンドウの平均メトリクスが含まれています。これには、指定された各分位数の wQL メトリクスと、平均 wQL、RMSE、MASE、MAPE、および WAPE メトリクスが含まれます。

両方のフォルダ内のファイルは、命名規則に従います:

<ExportJobName>\_<ExportTimestamp>\_<PartNumber>.csv。

Amazon Forecast のソフトウェア開発キット (SDK) および Amazon Forecast コンソールを使用して、精度メトリクスをエクスポートできます。

## Forecast SDK

[CreatePredictorBacktestExportJob](#) オペレーションを使用して、[DataDestination](#) オブジェクト内の S3 の場所と IAM ロールを、PredictorArn および PredictorBacktestExportJobName とあわせて指定します。

例:

```
{
  "Destination": {
    "S3Config": {
      "Path": "s3://bucket/example-path/",
      "RoleArn": "arn:aws:iam::000000000000:role/ExampleRole"
    }
  },
  "Format": "PARQUET";
  "PredictorArn": "arn:aws:forecast:region:predictor/example",
  "PredictorBacktestExportJobName": "backtest-export-name",
}
```

## Forecast Console

[Predictors] (予測子) のページで予測子を選択します。[Predictor metrics] (予測子メトリクス) のセクションで、[Export backtest results] (バックテストの結果をエクスポート) を選択します。

[Create predictor backtest export] (予測子バックテストのエクスポートを作成) のステージで、[Export name] (エクスポート名)、[IAM Role] (IAM ロール)、および [S3 predictor backtest export location] (S3 予測子バックテストのエクスポート場所) のフィールドを設定します。



## Create predictor backtest export [Info](#)

Export backtest data and metrics to an S3 location.

### Export details

#### Export name

The name can help you distinguish this export job from your other exports.

The export name must have 1 to 63 characters. Valid characters: a-z, A-Z, 0-9, and \_

#### IAM Role [Info](#)

Amazon forecast requires permissions to store the exported predictor in S3. Choose or create a role that has permissions to write to S3. If you created an IAM role when you imported a dataset and specified it in the Any S3 bucket field, choose that IAM role.

#### KMS Key ARN - optional

The ARN of the IAM role that Amazon Forecast uses to access the AWS KMS key.

The KMS key must have 1 to 256 characters. Valid characters: a-z, A-Z, 0-9, -, ., /, and :

#### S3 predictor backtest export location [Info](#)

This is the path to the S3 bucket or folder in the bucket where you want to store your exported predictor.

[Browse S3](#)

Your predictor export will be one or more CSV files.

#### ▼ Tags - optional [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.

[Cancel](#)[Create predictor backtest export](#)

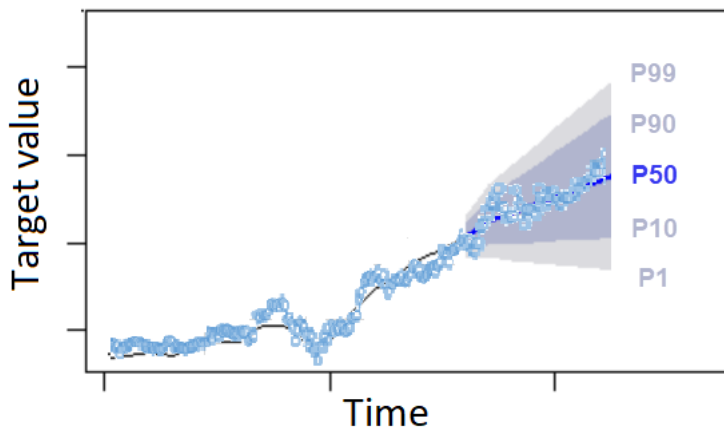
## Forecast タイプの選択

Amazon Forecast は、予測タイプを使用して予測を作成し、予測子を評価します。Forecast のタイプには、次の 2 つの形式があります。

- 平均予測タイプ - 平均を期待値として使用する予測。通常、特定の時間ポイントのポイント予測として使用されます。
- 分位数の予測タイプ - 指定された分位数での予測。通常、予測間隔を提供するために使用されます。これは、予測の不確実性を説明するための可能な値の範囲です。例えば、0.65 分位数での予測では、65% の確率で観測値よりも低い値が推定されます。

デフォルトでは、Forecast は、予測子の予測タイプに、0.1 (P10)、0.5 (P50)、および 0.9 (P90) といった値を使用します。mean と、0.01 (P1) から 0.99 (P99) までの範囲の分位数を含む、最大 5 つのカスタム予測タイプを選択できます。

分位数は、予測の上限と下限を提供できます。例えば、予測タイプ 0.1 (P10) および 0.9 (P90) を使用すると、80% 信頼区間と呼ばれる値の範囲が提供されます。観測値は 10% の確率で P10 の値よりも低く、P90 の値は 90% の確率で観測値よりも高いことが予想されます。p10 と P90 で予測を生成することにより、true の値は、80% の確率でこれらの範囲内に収まることが想定されます。この値の範囲は、以下の図の P10 と P90 の間の影付きの領域で示されています。



過小予測のコストが過大予測のコストと異なる場合は、分位数予測をポイント予測として使用することもできます。例えば、小売業のいくつかのケースにおいては、在庫不足のコストは、過剰に在庫を抱えている場合のコストよりも高くなります。これらの場合、0.65 (P65) での予測は、中央値 (P50) または平均予測よりも有益です。

予測子をトレーニングする場合、Amazon Forecast のソフトウェア開発キット (SDK) と Amazon Forecast コンソールを使用してカスタム予測タイプを選択できます。

## Forecast SDK

[CreateAutoPredictor](#) オペレーションを使用して、ForecastTypes パラメータでカスタム予想タイプを指定します。パラメータを文字列の配列としてフォーマットします。

例えば、0.01、mean、0.65、および 0.99 予測タイプで予測子を作成するには、次のコードを使用します。

```
{
  "ForecastTypes": [ "0.01", "mean", "0.65", "0.99" ],
},
```

## Forecast Console

[Train Predictor] (Predictor をトレーニング) のステージで、[Forecast types] (予測のタイプ) のフィールドでカスタム予測タイプを指定します。[Add new forecast type] (新規予測タイプを追加) を選択して、予測タイプ値を入力します。

例えば、0.01、mean、0.65、および 0.99 予測タイプを使用して予測子を作成するには、以下に示す [Forecast types] (予測タイプ) のフィールドに次の値を入力します。

### Forecast types - optional [Info](#)

Enter up to 5 quantile values between .01 and .99. The word 'mean' may also be entered if you wish to include the mean value.

Forecast type	Value	
<input type="text" value="Forecast type 1"/>	<input type="text" value=".01"/>	<input type="button" value="Remove"/>
<input type="text" value="Forecast type 2"/>	<input type="text" value="mean"/>	<input type="button" value="Remove"/>
<input type="text" value="Forecast type 3"/>	<input type="text" value=".65"/>	<input type="button" value="Remove"/>
<input type="text" value="Forecast type 4"/>	<input type="text" value=".99"/>	<input type="button" value="Remove"/>
<input type="button" value="Add new forecast type"/>		

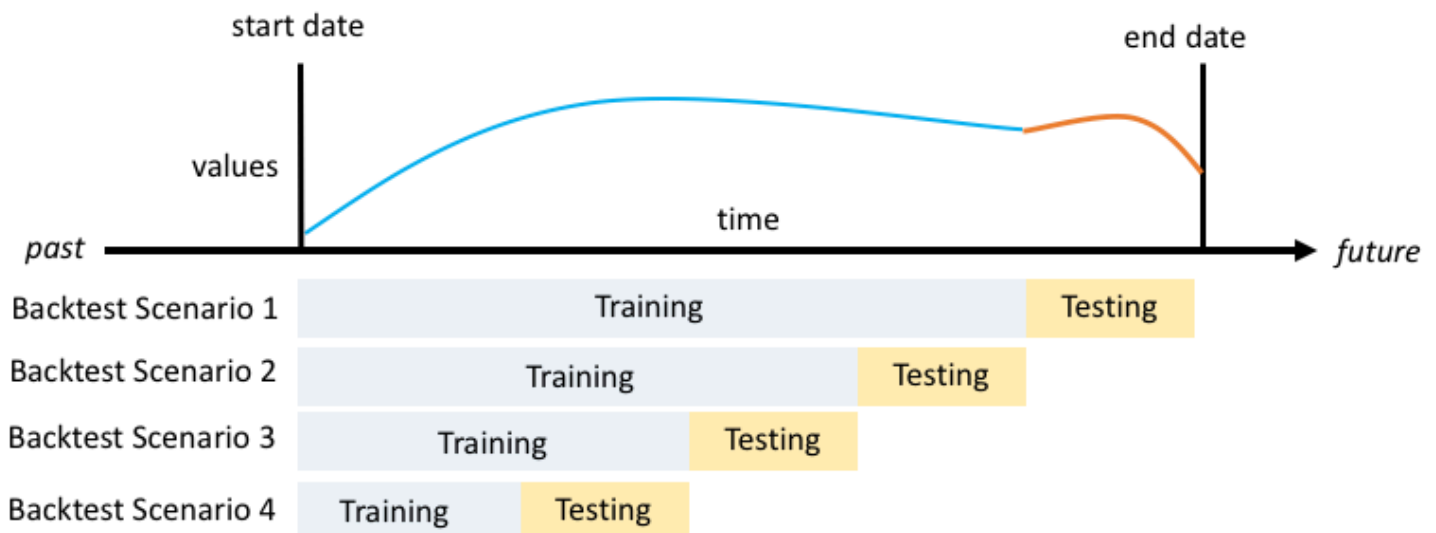
## レガシー予測子の使用

### バックテストパラメータの設定

Forecast は、バックテストを使用して精度メトリクスを計算します。複数のバックテストを実行する場合、Forecast は、すべてのバックテストウィンドウで各メトリクスを平均します。デフォルトでは、Forecast は、1 つのバックテストを計算し、バックテストウィンドウ (テストセット) のサイズは予測期間 (予測ウィンドウ) の長さに等しくなります。予測子をトレーニングするときに、バックテストウィンドウの長さ and バックテストシナリオの数の両方を設定できます。

Forecast は、バックテストプロセスから入力された値を除外し、特定のバックテストウィンドウ内で値が入力された項目は、そのバックテストから除外されます。これは、Forecast がバックテスト中に予測値と観測値を比較するだけであり、入力された値は観測値ではないためです。

バックテストウィンドウは、少なくとも予測期間と同じ大きさであり、ターゲット時系列データセット全体の長さの半分よりも小さくありません。1~5 個のバックテストから選択できます。



一般に、バックテストの数を増やすと、テスト中に時系列の大部分が使用され、Forecast はすべてのバックテストのメトリクスの平均をとることができるため、より信頼性の高い精度メトリクスが生成されます。

Amazon Forecast のソフトウェア開発キット (SDK) および Amazon Forecast コンソールを使用して、バックテストパラメータを設定できます。

## Forecast SDK

[CreatePredictor](#) オペレーションを使用して、[EvaluationParameters](#) データ型にバックテストパラメータを設定します。BackTestWindowOffset パラメータを使用してバックテスト中に設定されたテストの長さを指定し、NumberOfBacktestWindows パラメータを使用してバックテストウィンドウの数を指定します。

例えば、10 個の時間ポイントのテストセットで 2 つのバックテストを実行するには、次のコードを使用します。

```
"EvaluationParameters": {
  "BackTestWindowOffset": 10,
  "NumberOfBacktestWindows": 2
}
```

## Forecast Console

[Train Predictor] (Predictor をトレーニング) のステージでは、バックテスト中に設定されたテストの長さを [Backtest window offset] (バックテストウィンドウオフセット) のフィールドで設定

し、バックテストウィンドウの数を [Number of backtest windows] (バックテストウィンドウの数) のフィールドで設定します。

例えば、10 個の時間ポイントのテストセットで 2 つのバックテストを実行するには、次の値を設定します。

**Number of backtest windows - optional Info**

This is the number of times that the algorithm splits the input data for use in training and evaluation.

**Backtest window offset - optional Info**

This is the point in the dataset where you want to split the data for model training and evaluation.

## HPO および AutoML

デフォルトでは、Amazon Forecast は 0.1 (P10)、0.5 (P50)、および 0.9 (P90) 分位数を使用して、ハイパーパラメータ最適化 (HPO) 中のハイパーパラメータチューニングと AutoML 中のモデル選択を行います。予測子の作成時にカスタム予測タイプを指定すると、Forecast は、HPO および AutoML 中にそれらの予測タイプを使用します。

カスタム予測タイプが指定されている場合、Forecast は、HPO および AutoML 中に、それらの指定された予測タイプを使用して最適な結果を決定します。HPO 中、Forecast は、最初のバックテストウィンドウを使用して最適なハイパーパラメータ値を見つけます。AutoML 中、Forecast は、すべてのバックテストウィンドウの平均と HPO からの最適なハイパーパラメータ値を使用して、最適なアルゴリズムを見つけます。

AutoML と HPO の両方について、Forecast は、予測タイプ全体の平均損失を最小化するオプションを選択します。AutoML および HPO 中に、平均重み付き分位損失 (平均 wQL)、重み付き絶対誤差率 (WAPE)、二乗平均平方根誤差 (RMSE)、平均絶対パーセント誤差 (MAPE)、または平均絶対スケールリング誤差 (MASE) といった精度メトリクスのいずれかを使用して予測子を最適化することもできます。

Amazon Forecast の ソフトウェア開発キット (SDK) と Amazon Forecast コンソールを使用して、最適化メトリクスを選択できます。

### Forecast SDK

[CreatePredictor](#) オペレーションを使用して、ObjectiveMetric パラメータでカスタム予想タイプを指定します。

ObjectiveMetric パラメータには、次の値を指定できます。

- AverageWeightedQuantileLoss - 平均重み付き分位損失
- WAPE - 重み付き絶対誤差率
- RMSE - 二乗平均平方根誤差
- MAPE - 平均絶対パーセント誤差
- MASE - 平均絶対スケーリング誤差

例えば、AutoML を使用して予測子を作成し、平均絶対スケーリング誤差 (MASE) 精度メトリクスを使用して最適化するには、次のコードを使用します。

```
{
  ...
  "PerformAutoML": "true",
  ...
  "ObjectiveMetric": "MASE",
},
```

## Forecast Console

[Train Predictor] (Predictor をトレーニング) のステージ中、[Automatic (AutoML)] (自動 (AutoML)) を選択します。[目的のメトリクス] のセクションで、予測子の最適化に使用する精度メトリクスを選択します。

例えば、次の画像は、AutoML を使用して作成され、平均絶対スケーリング誤差 (MASE) 精度メトリクスを使用して最適化された予測子を示しています。

コンソールを使用する場合、AutoML を使用して予測子を作成するときのみ、Objective メトリクスを指定できます。アルゴリズムを手動で選択した場合、HPO のために Objective メトリクスを指定することはできません。

## 予測子の再トレーニング

### Note

再トレーニングは、AutoPredictor ([CreateAutoPredictor](#)) で作成された予測子でのみ使用できます。既存のレガシー予測子を AutoPredictor にアップグレードできます。「[the section called “へのアップグレード AutoPredictor”](#)」を参照してください。

予測子を最新の状態に保つために、更新されたデータセットで予測子を保持できます。予測子を再トレーニングする場合、Amazon Forecast は、同じ予測子の構成設定を維持します。再トレーニング後も、元の予測子はアクティブなままで、再トレーニングされた予測子には個別の Predictor ARN があります。

予測子を再トレーニングすると、次の 2 つの方法で予測精度を向上させることができます。

1. より最新のデータ: 再トレーニングされた予測子は、モデルをトレーニングするときに、より最新のデータを組み込みます。
2. 予測子の改善: 再トレーニングされた予測子には、Amazon Forecast アルゴリズムと追加のデータセットの更新と改善が組み込まれます。

予測子の再トレーニングでは、新しい予測子を最初から作成する場合と比較して、最大 50% の時間を短縮できます。Predictor のトレーニング時間はより短く、Forecast は既存の構成設定を自動的に使用します。

#### Python ノートブック

予測子の再トレーニングに関するステップバイステップガイドについては、「[Retraining a predictor](#)」(予測子の再トレーニング)を参照してください。

ソフトウェア開発キット (SDK) または Amazon Forecast コンソールを使用して、予測子を再トレーニングできます。

#### Console

予測子を再トレーニングするには

1. AWS Management Console にサインインして、Amazon Forecast コンソール (<https://console.aws.amazon.com/forecast/>) を開きます。
2. ナビゲーションペインで、[Predictors] (予測子) を選択します。
3. 再学習する予測子を選択します。
4. [Predictor actions] (Predictor アクション) のドロップダウンで、[Retrain] (再トレーニング) を選択します。
5. アップグレードされた予測子の一意の名前を設定します。
6. [Retrain predictor] (予測子の再トレーニング) を選択します。

## SDK

予測子を再トレーニングするには

[CreateAutoPredictor](#) オペレーションを使用して、予測子に一意的な名前を割り当て、再トレーニングする予測子に `ReferencePredictorArn` の値を設定します。

```
{
  "PredictorName": "RetrainedPredictor",
  "ReferencePredictorArn": "arn:aws:forecast:us-west-2:938097332257:predictor/OriginalPredictor"
}
```

予測子を再トレーニングするときは、`PredictorName` と `ReferencePredictorArn` パラメータのみに値を割り当てます。

## Weather Index

Amazon Forecast の Weather Index はビルトインの特徴量化であり、履歴気象情報および予測される気象情報をモデルに組み込みます。これは、気温と降水量が製品需要に大きな影響を与える可能性がある小売業のユースケースで特に役立ちます。

Weather Index が有効になっている場合、Forecast は、予測子のトレーニング中に精度の向上が見られる時系列にのみ気象の特徴量化を適用します。時系列に気象情報を追加してもバックテスト中の予測精度が向上しない場合、Forecast はその特定の時系列に Weather Index を適用しません。

Weather Index を適用するには、ターゲットの時系列データセットおよび関連する時系列データセットに [geolocation](#) 属性を含める必要があります。また、ターゲットの時系列タイムスタンプの [タイムゾーン](#) を指定する必要があります。データセット要件の詳細については、「[条件と制約事項](#)」を参照してください。

### Python ノートブック

Weather Index の使用に関するステップバイステップガイドについては、「[NY Taxi: Amazon Forecast with Weather Index](#)」(NY タクシー: Weather Index を使用した Amazon Forecast) を参照してください。

## トピック



- [Weather Index の有効化](#)
- [データセットへの geolocation の追加](#)
- [タイムゾーンの指定](#)
- [条件と制限](#)

## Weather Index の有効化

Weather Index は、予測子のトレーニング段階で有効になります。[CreateAutoPredictor](#) オペレーションを使用する場合、Weather Index は [AdditionalDataset](#) データ型に含まれます。

Weather Index を有効にする前に、ターゲットの時系列および関連する時系列データセットに geolocation 属性を含め、タイムスタンプのタイムゾーンを定義する必要があります。詳細については、「[Adding Geolocation Information](#)」(geolocation 情報の追加)と「[Specifying Time Zones](#)」(タイムゾーンの指定)を参照してください。

Forecast コンソールまたは Forecast ソフトウェア開発キット (SDK) を使用して、Weather Index を有効にできます。

### Console

Weather Index を有効にするには

1. AWS Management Console にサインインして、Amazon Forecast コンソール (<https://console.aws.amazon.com/forecast/>) を開きます。
2. [Dataset groups] (データセットグループ) から、データセットグループを選択します。
3. ナビゲーションペインで、[Predictors] (予測子) を選択します。
4. [Train new predictor] (新しい予測子のトレーニング) を選択します。
5. [Enable Weather Index] (Weather Index を有効化) を選択します。

### SDK

Weather Index を有効にするには

[CreateAutoPredictor](#) オペレーションを使用して、[AdditionalDataset](#) データ型に "Name": "weather" と "Value": "true" を追加して Weather Index を有効にします。

```
"DataConfig": {
```

```
...
  "AdditionalDatasets": [
    ...
    {
      "Name": "weather",
    }
  ]
},
```

## データセットへの geolocation の追加

Weather Index を使用するには、ターゲットの時系列データセットおよび関連する時系列データセットに各項目についての geolocation 属性を含める必要があります。属性は、データセットスキーマ内の geolocation 属性タイプで定義されます。

あるデータセット内のすべての geolocation の値は、単一のリージョン内にのみ存在する必要があります。リージョンは、米国 (ハワイ州とアラスカ州を除く)、カナダ、南米、中米、アジアパシフィック、欧州、アフリカおよび中東です。

次の 2 つの形式のいずれかで geolocation 属性を指定します。

- [Latitude & Longitude] (緯度と経度) (すべてのリージョン) - 緯度と経度を 10 進数のフォーマットで指定します (例: 47.61\_-122.33)
- 郵便番号 (米国のみ) - 国コード (US) の後に 5 桁の郵便番号を指定します (例: US\_98121)

[Latitude & Longitude (緯度および経度)] の形式は、すべてのリージョンでサポートされています。  
[Postal code (郵便番号)] の形式は、米国リージョンでのみサポートされています。

### トピック

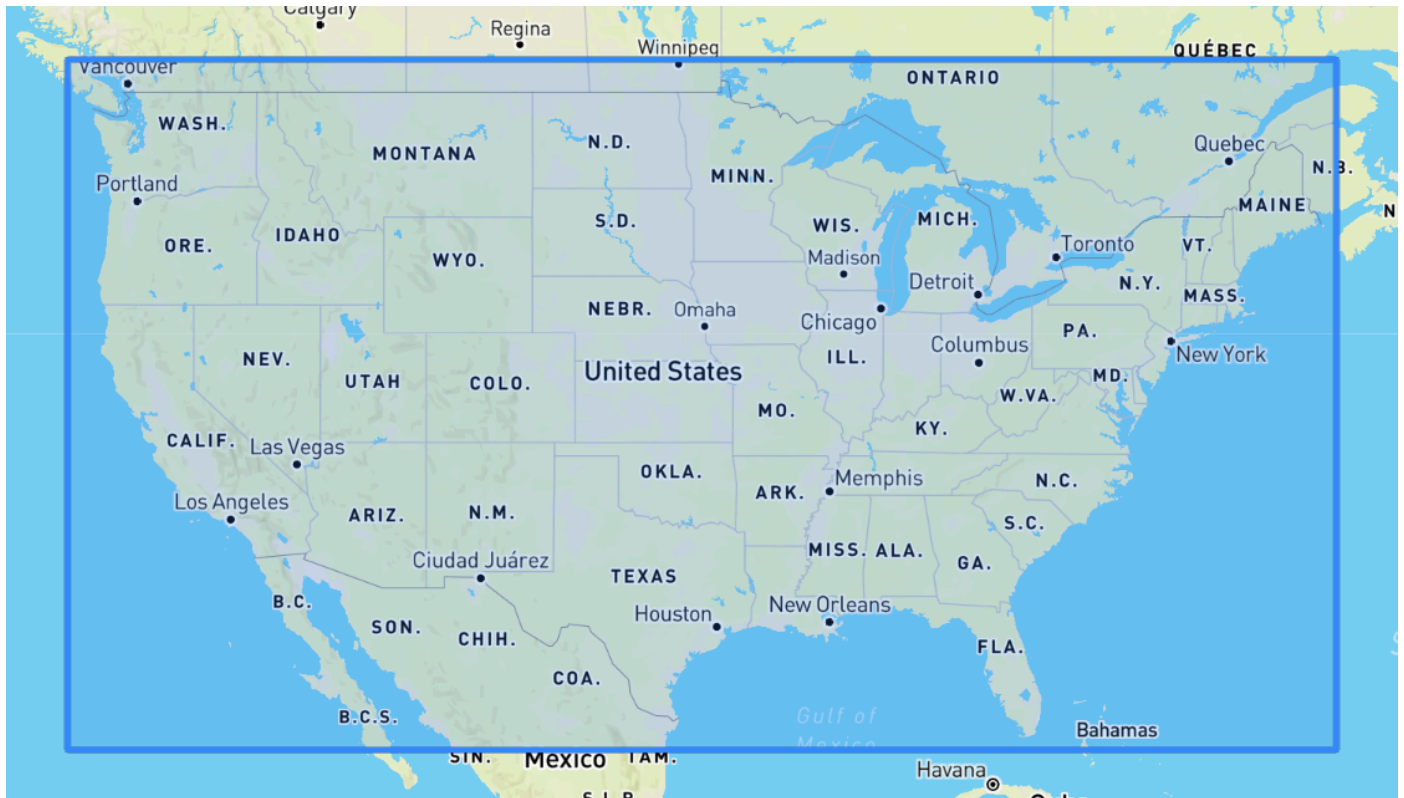
- [緯度と経度の境界](#)
- [データセットスキーマに geolocation を含める](#)
- [geolocation フォーマットの設定](#)

## 緯度と経度の境界

以下は、承認されたリージョンの緯度と経度の境界です。

## US Region

境界: 緯度 (24.6, 50.0)、経度 (-126.0, -66.4)。



## Canada Region

境界: 緯度 (41.0, 75.0)、経度 (-142.0, -52.0)。



## Europe Region

境界: 緯度 (34.8, 71.8)、経度 (-12.6, 44.8)。



## South America Region

境界: 緯度 (-56.6, 14.0)、経度 (-82.4, -33.00)。



## Asia Pacific Region

境界: 緯度 (-47.8, 55.0)、経度 (67.0, 180.60)。



## Central America Region

境界: 緯度 (6.80、33.20)、経度 (-118.80、-58.20)。



## Africa & Middle East Region

境界: 緯度 (-35.60, 43.40)、経度 (-18.80, -58.20)。





## データセットスキーマに geolocation を含める

コンソールまたは [CreateDataset](#) オペレーションを使用して、ターゲット時系列および関連する時系列の JSON スキーマ内で場所の属性タイプを「geolocation」として定義します。スキーマの属性は、データセットに表示されるとおりに順序付けする必要があります。

```
{  
  "Attributes": [  
    {
```

```
    "AttributeName": "timestamp",
    "AttributeType": "timestamp"
  },
  {
    "AttributeName": "target_value",
    "AttributeType": "float"
  },
  {
    "AttributeName": "item_id",
    "AttributeType": "string"
  },
  {
    "AttributeName": "location",
    "AttributeType": "geolocation"
  }
]
}
```

## geolocation フォーマットの設定

geolocation 属性の形式は、[Postal Code] (郵便番号) または [Latitude & Longitude] (緯度と経度) の形式にすることができます。Forecast コンソールまたは Forecast ソフトウェア開発キット (SDK) を使用して、geolocation の形式を設定できます。

### Console

時系列データセットに geolocation 属性を追加するには

1. AWS Management Console にサインインして、Amazon Forecast コンソール (<https://console.aws.amazon.com/forecast/>) を開きます。
2. [データセットグループの作成] を選択します。
3. [Schema builder] (スキーマビルダー) で、geolocation の [Attribute type] (属性タイプ) を geolocation に設定します。
4. [Geolocation フォーマット] のドロップダウンで、ロケーションフォーマットを選択します。

### Dataset details

**Dataset name**  
The name can help you distinguish this dataset from other datasets on your Datasets dashboard.

example\_dataset

The dataset name must have 1 to 63 characters. Valid characters: a-z, A-Z, 0-9, and \_

**Frequency of your data**  
This is the frequency at which entries are registered into your data file.

Your data entries have a time interval of 1 day

**Data schema [Info](#)**  
Use the data schema section to specify the attribute types for each column in your dataset. You can specify the schema in two ways:

**Schema builder**  
Specify your Attribute Name, Attribute Type, and attribute order in the text boxes provided.

**JSON schema**  
Specify AttributeName and AttributeType in the JSON format.

**Schema Builder [Info](#)**  
The attributes below are required for your chosen domain. You may add additional attributes. All attributes displayed must exist in your CSV file and must be ordered in the same order that they appear in your CSV file. To reorder the attributes, simply drag and drop each attribute to the correct position.

Column

1	Attribute Name item_id	Attribute Type string	
2	Attribute Name timestamp	Attribute Type timestamp	Timestamp Format <a href="#">Info</a> yyyy-MM-dd
3	Attribute Name target_value	Attribute Type float	
4	Attribute Name location	Attribute Type geolocation	Geolocation format <a href="#">Info</a> X <ul style="list-style-type: none"> <li>Lat/Long Decimal Degrees (US a... ▲</li> <li>Lat/Long Decimal Degrees (US and Europe) <b>#####_###.#####</b></li> <li>Postal Code (US only)</li> <li>CountryCode_PostalCode</li> </ul>

[Add attribute](#)

You can add up to 9 attributes.

また、JSON 形式で属性を定義し、[Geolocation format] (geolocation の形式) のドロップダウンから場所の形式を選択することもできます。

## SDK

時系列データセットに geolocation 属性を追加するには

[CreateDatasetImportJob](#) オペレーションを使用して、GeolocationFormat の値を次のいずれかに設定します。

- [Latitude & longitude] (緯度と経度) (すべてのリージョン): "LAT\_LONG"
- 郵便番号 (米国のみ): "CC\_POSTALCODE"

例えば、緯度と経度の形式を指定するには、CreateDatasetImportJob リクエストに次の情報を含めます。

```
{
  ...
  "GeolocationFormat": "LAT_LONG"
}
```

## タイムゾーンの指定

Amazon Forecast にタイムゾーン情報を geolocation 属性と自動的に同期させるか、データセット全体に単一のタイムゾーンを手動で割り当てることができます。

### トピック

- [タイムゾーンを geolocation と自動的に同期する](#)
- [単一のタイムゾーンを手動で選択する](#)

## タイムゾーンを geolocation と自動的に同期する

このオプションは、複数のタイムゾーンのタイムスタンプを含み、それらのタイムスタンプが現地時間で表されるデータセットに最適です。Forecast は、項目の geolocation 属性に基づいて、ターゲット時系列データセット内のすべての項目にタイムゾーンを割り当てます。

Forecast コンソールまたは Forecast SDK を使用して、タイムスタンプを geolocation 属性と自動的に同期できます。

### Console

タイムゾーンを geolocation 属性と同期するには

1. AWS Management Console にサインインして、Amazon Forecast コンソール (<https://console.aws.amazon.com/forecast/>) を開きます。
2. ナビゲーションペインで、[Create dataset group] (データセットグループを作成) を選択します。

3. [Dataset import details] (データセットのインポートの詳細) で、[Sync time zone with location] (タイムゾーンを場所と同期) を選択します。

### Dataset import details

**Dataset import name**  
The name can help you distinguish this dataset import from other imports on your dataset detail page.

The dataset import name must have 1 to 63 characters. Valid characters: a-z, A-Z, 0-9, and \_

**Time zone [Info](#)**  
Select a time zone option.

**Select time zone**  
Manually select a single time zone. Use this option if your timestamps are normalized to a single time zone.

**Sync time zone with location**  
Automatically derive multiple time zones from your geolocation attribute. Use this option if timestamps are listed in multiple time zones.

**Data location [Info](#)**  
The location is the path to the file in your S3 bucket that contains your data.

Your files must be in CSV format.

**IAM role [Info](#)**  
Dataset groups require permissions from IAM to read your dataset files in S3. Choose or create a role using this control.

Custom IAM role ARN

## SDK

タイムゾーンを geolocation 属性と同期するには

[CreateDatasetImportJob](#) オペレーションを使用して、"UseGeolocationForTimeZone" を "true" に設定します。

```
{
  ...
  "UseGeolocationForTimeZone": "true"
}
```

## 単一のタイムゾーンを手動で選択する

### Note

米国リージョン、カナダリージョン、南米リージョン、中米リージョン、アジアパシフィックリージョン、欧州リージョン、アフリカおよび中東リージョン以外のタイムゾーンを手動で選択できます。ただし、すべての geolocation の値は、これらのいずれかのリージョン内にある必要があります。

このオプションは、すべてのタイムスタンプが単一のタイムゾーン内にあるデータセット、またはすべてのタイムスタンプが単一のタイムゾーンに正規化されているデータセットに最適です。このオプションを使用すると、データセット内のすべての項目に同じタイムゾーンが適用されます。

Weather Index では、以下のタイムゾーンを使用できます。

### 米国リージョン

- America/Los\_Angeles
- America/Phoenix
- America/Denver
- America/Chicago
- America/New\_York

### カナダリージョン

- America/Vancouver
- America/Edmonton
- America/Regina
- America/Winnipeg
- America/Toronto
- America/Halifax
- America/St\_Johns

### 欧州リージョン

- Europe/London

- Europe/Paris
- Europe/Helsinki

#### 南米リージョン

- America/Buenos\_Aires
- America/Noronha
- America/Caracas

#### アジアパシフィックリージョン

- Asia/Kabul
- Asia/Karachi
- Asia/Kolkata
- Asia/Kathmandu
- Asia/Dhaka
- Asia/Rangoon
- Asia/Bangkok
- Asia/Singapore
- Asia/Seoul
- Australia/Adelaide
- Australia/Melbourne
- Australia/Lord\_Howe
- Australia/Eucla
- Pacific/Norfolk
- Pacific/Auckland

#### 中米

- America/Puerto\_Rico

#### アフリカおよび中東

- Africa/Nairobi
- Asia/Tehran

- Asia/Dubai

## その他

- Pacific/Midway
- Pacific/Honolulu
- Pacific/Marquesas
- America/Anchorage
- Atlantic/Cape\_Verde
- Asia/Anadyr
- Pacific/Chatham
- Pacific/Enderbury
- Pacific/Kiritimati

データセット内の項目が受け入れられたリージョン内にあるが、タイムスタンプがそのリージョン外のタイムゾーンに標準化されている場合は、[その他] のリストからタイムゾーンを選択します。

有効なタイムゾーン名の完全なリストについては、「[Joda-Time library](#)」(Joda-Time ライブラリ) を参照してください。

Forecast コンソールまたは Forecast SDK を使用して、データセットのタイムゾーンを手動で設定できます。

## Console

データセットの単一のタイムゾーンを選択するには

1. AWS Management Console にサインインして、Amazon Forecast コンソール (<https://console.aws.amazon.com/forecast/>) を開きます。
2. ナビゲーションペインで、[Create dataset group] (データセットグループを作成) を選択します。
3. [Dataset import details] (データセットのインポートの詳細) で、[Select time zone] (タイムゾーンを選択) を選択します。

例えば、以下を使用して、ロサンゼルス時間 (パシフィック標準時) をデータセットに適用します。



### Dataset import details

**Dataset import name**  
The name can help you distinguish this dataset import from other imports on your dataset detail page.

The dataset import name must have 1 to 63 characters. Valid characters: a-z, A-Z, 0-9, and \_

**Time zone** [Info](#)  
Select a time zone option.

**Select time zone**  
Manually select a single time zone. Use this option if your timestamps are normalized to a single time zone.

**Sync time zone with location**  
Automatically derive multiple time zones from your geolocation attribute. Use this option if timestamps are listed in multiple time zones.

**Select time zone** [Info](#)  
Select a time zone for your dataset.

**Data location** [Info](#)  
The location is the path to the file in your S3 bucket that contains your data.

Your files must be in CSV format.

**IAM role** [Info](#)  
Dataset groups require permissions from IAM to read your dataset files in S3. Choose or create a role using this control.

Custom IAM role ARN

## SDK

データセットの単一のタイムゾーンを選択するには

[CreateDatasetImportJob](#) オペレーションを使用して、"TimeZone" を有効なタイムゾーンに設定します。

例えば、以下を使用して、ロサンゼルス時間 (パシフィック標準時) をデータセットに適用します。

```
{
  ...
  "TimeZone": "America/Los_Angeles"
}
```

## 条件と制限

Weather Index を使用する場合は、次の条件と制約事項が適用されます。

- 使用可能なアルゴリズム: レガシー予測子を使用している場合、CNN-QR、DeepAR+、および Prophet アルゴリズムを使用して予測子をトレーニングすると、Weather Index を有効にできます。Weather Index は、ARIMA、ETS、および NPTS には適用されません。
- 予想頻度: 有効な予想頻度は、Minutely、Hourly、および Daily です。
- [Forecast horizon] (予測期間): 予測期間は、今後 14 日間を超えて設定することはできません。各予測頻度の予測期間の制限については、以下のリストを参照してください。
  - 1 minute - 500
  - 5 minutes - 500
  - 10 minutes - 500
  - 15 minutes - 500
  - Hourly - 330
  - Daily - 14
- 時系列の長さ: Weather Index を使用してモデルをトレーニングする場合、Forecast は、Forecast 気象データセットの特徴量化の開始日前のタイムスタンプを持つすべての時系列データセットを切り捨てます。Forecast weather データセットの機能には、次の開始日が含まれています。
  - 米国リージョン: 2018 年 7 月 2 日
  - 欧州リージョン: 2018 年 7 月 2 日
  - アジアパシフィックリージョン: 2018 年 7 月 2 日
  - カナダリージョン: 2019 年 7 月 2 日
  - 南米リージョン: 2020 年 1 月 2 日
  - 中米リージョン: 2020 年 9 月 2 日
  - アフリカおよび中東リージョン: 2021 年 3 月 25 日

Weather Index を有効にすると、開始日の前のタイムスタンプを持つデータポイントは、予測子のトレーニング中に使用されません。

- 場所の数: ターゲット時系列データセットは、2000 の一意の場所を超えることはできません。
- リージョンの境界: データセット内のすべてのアイテムは 1 つのリージョン内にある必要があります。
- [Minimum time series length] (最小時系列長): Weather Index をテストする際の追加のデータ要件により、時系列データセットの最小長は次のとおりです:

$$3 \times \text{ForecastHorizon} + (\text{BacktestWindows} + 1) \times \text{BacktestWindowOffset}$$

時系列データセットがこの要件を満たしていない場合は、以下を減らすことを検討してください。

- `ForecastHorizon` - 予測期間を短縮します。
- `BacktestWindowOffset` - バックテスト中にテストセットの長さを短くします。
- `BacktestWindows` - バックテストの数を減らします。

## Holidays の特徴量化

Holidays は、特徴エンジニアリングされた祝祭日情報データセットをモデルに組み込むビルトインの特徴量化です。250 か国を超える祝祭日カレンダーをネイティブにサポートします。Amazon Forecast には、ホリデーカレンダーを生成するための [Holiday API ライブラリ](#) と [Jollyday API](#) の両方が組み込まれています。

Holidays の特徴量化は、祝祭日が需要に大きな影響を与える可能性がある小売ドメインで特に役立ちます。

Holiday 特徴量化では、最小 5 分、最大 1 か月の予想頻度がサポートされます。

### トピック

- [Holidays の特徴量化の有効化](#)
- [国コード](#)
- [追加の祝祭日カレンダー](#)

## Holidays の特徴量化の有効化

Holidays の特徴量化は、[\[Additional Dataset\]](#) (追加のデータセット) として Amazon Forecast に含まれており、予測子をトレーニングする前に有効になります。履歴データには少なくとも 2 年分のデータを含めることをお勧めします。これにより、Forecast は特定の休日に関連する需要パターンを特定できます。国を選択すると、Holidays は、トレーニング中にその国の祝祭日カレンダーをデータセット内のすべてのアイテムに適用します。

Amazon Forecast コンソールまたは Forecast ソフトウェア開発キット (SDK) を使用して、Holidays を有効にできます。

### Forecast SDK

[CreateAutoPredictor](#) オペレーションを使用して、「"Name": "holiday"」を追加し、2 文字の国コードを "CountryCode" にマッピングするように "Configuration" を設定して、Holidays を有効にします。[the section called “国コード”](#) を参照してください。

例えば、米国の祝祭日カレンダーを含めるには、次のコードを使用します。

```
"DataConfig": {
  "AdditionalDatasets": [
    {
      "Name": "holiday",
      "Configuration": {
        "CountryCode" : ["US"]
      }
    },
  ]
},
```

## Forecast Console

[Train Predictor] (Predictor をトレーニング) のステージで、[Country for Holidays] (Holidays の国) のドロップダウンから国を選択します。

### Holidays | [Info](#)

Include holidays in predictor training to improve forecast accuracy.

Activate holidays

Select a country

Choose a country ▼

## 国コード

Amazon Forecast は、以下の国の祝祭日カレンダーのネイティブサポートを提供します。API で国を指定するときは、[Country Code] (国コード) を使用します。

サポートされている国

国	国コード
アフガニスタン	AF
オーランド諸島	AX

国	国コード
アルバニア	AL
アルジェリア	DZ
アメリカ領サモア	AS
アンドラ	AD
アンゴラ	AO
アンギラ	AI
南極	AQ
アンティグアバーブーダ	AG
アルゼンチン	AR
アルメニア	AM
アルバ	AW
オーストラリア	AU
オーストリア	AT
アゼルバイジャン	AZ
バハマ諸島	BS
バーレーン	BH
バングラデシュ	BD
バルバドス	BB
ベラルーシ	BY
ベルギー	BE

国	国コード
ベリーズ	BZ
ベニン	BJ
バミューダ	BM
ブータン	BT
ボリビア	BO
ボスニアヘルツェゴビナ	BA
ボツワナ	BW
ブーベ島	BV
ブラジル	BR
英領インド洋地域	IO
英領バージン諸島	VG
ブルネイ・ダルサラーム	BN
ブルガリア	BG
ブルキナファソ	BF
ブルンジ	BI
カンボジア	KH
カメルーン	CM
カナダ	CA
カーボベルデ	CV
オランダカリブ領域	BQ

国	国コード
ケイマン諸島	KY
中央アフリカ共和国	CF
チャド	TD
チリ	CL
中国	CN
クリスマス島	CX
ココス (キーリング) 諸島	CC
コロンビア	CO
コモロ	KM
クック諸島	CK
コスタリカ	CR
クロアチア	HR
キューバ	CU
キュラソー	CW
キプロス	CY
チェコ共和国	CZ
コンゴ民主共和国	CD
デンマーク	DK
ジブチ	DJ
ドミニカ	DM

国	国コード
ドミニカ共和国	DO
エクアドル	EC
エジプト	EG
エルサルバドル	SV
赤道ギニア	GQ
エリトリア	ER
エストニア	EE
エスワティニ	SZ
エチオピア	ET
フォークランド諸島	FK
フェロー諸島	FO
フィジー	FJ
フィンランド	FI
フランス	FR
フランス領ギアナ	GF
フランス領ポリネシア	PF
フランス領南方・南極地域	TF
ガボン	GA
ガンビア	GM
ジョージア	GE



国	国コード
ドイツ	DE
ガーナ	GH
ジブラルタル	GI
ギリシャ	GR
グリーンランド	GL
グレナダ	GD
グアドループ	GP
グアム	GU
グアテマラ	GT
ガーンジー代官管轄区	GG
ギニア	GN
ギニアビサウ	GW
ガイアナ	GY
ハイチ	HT
ハード諸島と McDonald 諸島	HM
ホンジュラス	HN
香港	HK
ハンガリー	HU
アイスランド	IS
インド	IN

国	国コード
インドネシア	ID
イラン	IR
イラク	IQ
アイルランド	IE
マン島	IM
イスラエル	IL
イタリア	IT
コートジボワール	CI
ジャマイカ	JM
日本	JP
ジャージー	JE
ヨルダン	JO
カザフスタン	KZ
ケニア	KE
キリバス	KI
コソボ共和国	XK
クウェート	KW
キルギスタン	KG
ラオス	LA
ラトビア	LV

国	国コード
レバノン	LB
レソト	LS
リベリア	LR
リビア	LY
リヒテンシュタイン	LI
リトアニア	LT
ルクセンブルグ	LU
マカオ	MO
マダガスカル	MG
マラウイ	MW
マレーシア	MY
モルジブ	MV
マリ	ML
マルタ	MT
マーシャル諸島共和国	MH
マルチニーク	MQ
モーリタニア	MR
モーリシャス	MU
マヨット	YT
メキシコ	MX

国	国コード
ミクロネシア	FM
モルドバ	MD
モナコ	MC
モンゴル	MN
モンテネグロ	ME
モントセラト	MS
モロッコ	MA
モザンビーク	MZ
ミャンマー	MM
ナミビア	NA
ナウル	NR
ネパール	NP
オランダ	NL
ニューカレドニア	NC
ニュージーランド	NZ
ニカラグア	NI
ニジェール	NE
ナイジェリア	NG
ニウエ	NU
ノーフォーク島	NF

国	国コード
北朝鮮	KP
北マケドニア	MK
北マリアナ諸島	MP
ノルウェー	NO
オマーン	OM
パキスタン	PK
パラオ	PW
パレスチナ	PS
パナマ	PA
パプアニューギニア	PG
パラグアイ	PY
ペルー	PE
フィリピン	PH
ピトケアン諸島	PN
ポーランド	PL
ポルトガル	PT
プエルトリコ	PR
カタール	QA
コンゴ共和国	CG
レユニオン	RE

国	国コード
ルーマニア	RO
ロシア連邦	RU
ルワンダ	RW
サン・バルテルミー	BL
セントヘレナ・アセンションおよびトリスタンダクーニャ	SH
セントクリストファーネイビス	KN
セントルシア	LC
サン・マルタン島	MF
サンピエール・ミクロン	PM
セントビンセントおよびグレナディーン諸島	VC
サモア	WS
サンマリノ	SM
サントメプリンシペ	ST
サウジアラビア	SA
セネガル	SN
セルビア	RS
セイシェル	SC
シエラレオネ	SL
シンガポール	SG
シントマーテルテン	SX

国	国コード
スロバキア	SK
スロベニア	SI
ソロモン諸島	SB
ソマリア	SO
南アフリカ	ZA
サウスジョージア・サウスサンドウィッチ諸島	GS
韓国	KR
南スーダン	SS
スペイン	ES
スリランカ	LK
スーダン	SD
スリナム	SR
スバルバル諸島およびヤンマイエン島	SJ
スウェーデン	SE
スイス	CH
シリア・アラブ共和国	SY
台湾	TW
タジキスタン	TJ
タンザニア	TZ
タイ	TH

国	国コード
東ティモール	TL
トーゴ	TG
トケラウ	TK
トンガ	TO
トリニダード・トバゴ	TT
チュニジア	TN
トルコ	TR
トルクメニスタン	TM
タークスおよびカイコス諸島	TC
ツバル	TV
ウガンダ	UG
ウクライナ	UA
アラブ首長国連邦	AE
英国	GB
国連	UN
アメリカ	US
合衆国領有小離島	UM
アメリカ領バージン諸島	VI
ウルグアイ	UY
ウズベキスタン	UZ



国	国コード
バヌアツ	VU
バチカン市国	VA
ベネズエラ	VE
ベトナム	VN
ウォリス・フツナ	WF
西サハラ	EH
イエメン	YE
ザンビア	ZM
ジンバブエ	ZW

## 追加の祝祭日カレンダー

Amazon Forecast は、インド、韓国、およびアラブ首長国連邦の祝祭日もサポートしています。それらの祝祭日を以下に示します。

### India - "IN"

- 1 月 26 日 - 共和国記念日
- 8 月 15 日 - 独立記念日
- 10 月 2 日 - マハトマ・ガンジー生誕日

### Korea - "KR"

- 1 月 1 日 - 新年
- 3 月 1 日 - 独立運動記念日
- 5 月 5 日 - こどもの日
- 6 月 6 日 - 戦没者慰霊日

8 月 15 日 - 解放記念日

10 月 3 日 - 建国記念日

10 月 9 日 - ハングルの日

12 月 25 日 - クリスマス

United Arab Emirates - "AE"

1 月 1 日 - 新年

12 月 1 日 - 記念日

12 月 2~3 日 - 建国記念日

ラマダン\*

断食明け大祭\*

犠牲祭\*

イスラーム暦新年\*

\*イスラームの祝祭日は月の周期によって決まります。

## Predictor Explainability

Predictor Explainability は、データセット内の属性がターゲット変数にどのように影響するかをよりよく理解するのに役立ちます。Forecast は、Impact スコアと呼ばれるメトリクスを使用して、各属性の相対的な影響を定量化し、それらが予測値を増加させるか減少させるかを決定します。

例えば、ターゲットが sales であり、price および color といった関連する 2 つの属性がある予測シナリオについて考えてみます。Forecast では、商品の価格が売上に大きく影響する (Impact スコアが高い) が、アイテムの色による影響はごくわずか (Impact スコアが低い) であることが判明する場合があります。

Predictor Explainability を有効にするには、関連する時系列、項目メタデータ、または祝祭日や Weather Index などの追加のデータセットの少なくとも 1 つを予測子に含める必要があります。詳細については、「[制約事項とベストプラクティス](#)」を参照してください。

特定の時系列および時間ポイントの Impact スコアを作成するには、Predictor Explainability ではなく Forecast Explainability を使用します。「[Forecast Explainability](#)」を参照してください。

## トピック

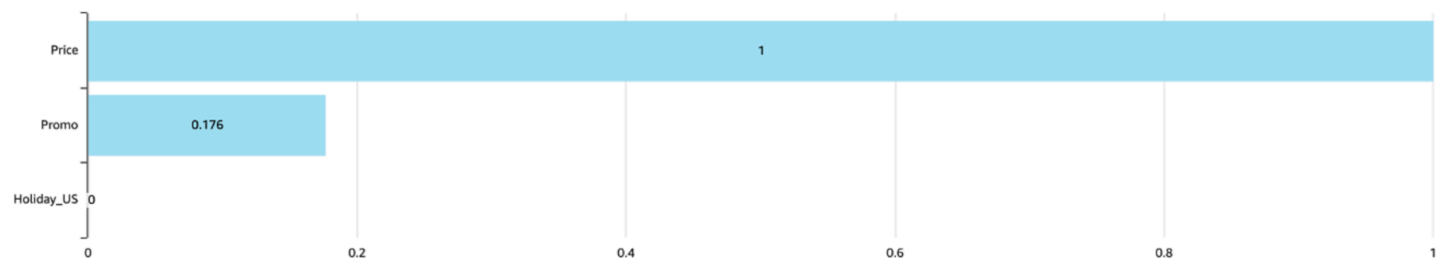
- [影響スコアの解釈](#)
- [Predictor Explainability の作成](#)
- [Predictor Explainability のエクスポート](#)
- [制約事項とベストプラクティス](#)

## 影響スコアの解釈

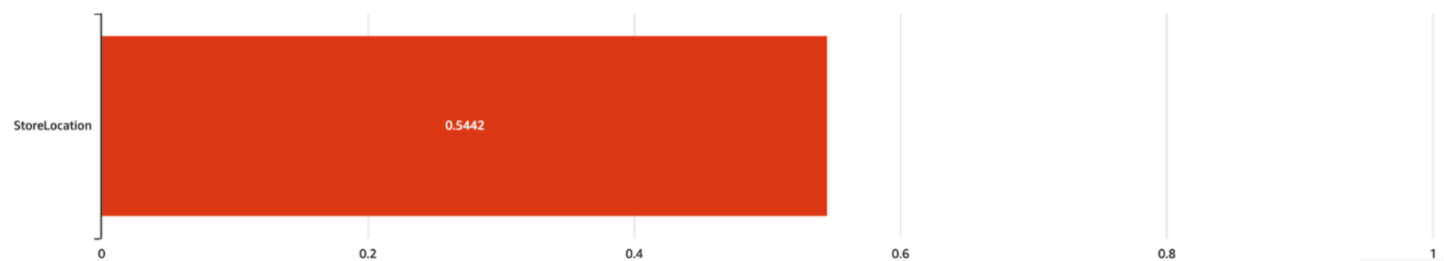
影響スコアは、予測値に対する相対的な影響属性を測定します。例えば、「price」属性の影響スコアが「store location」属性の 2 倍である場合、商品の料金は、予測値に対して、店舗の場所の 2 倍の影響を及ぼすと結論付けることができます。

影響スコアは、属性が予測値を増加させるか減少させるかに関する情報も提供します。コンソールでは、これは 2 つのグラフで示されます。青いバーのある属性は予測値を増やし、赤いバーのある属性は予測値を減らします。

Attributes increasing impact score



Attributes decreasing impact score



コンソールでは、Impact スコアの範囲は 0 から 1 です。ここで、スコア 0 は影響がないことを示し、スコアが 1 に近づくほど、その影響が重大であることを示します。SDK では、Impact スコアの範囲は -1 から 1 で、記号は影響の方向を示します。

Impact スコアは、絶対的な影響ではなく、属性の相対的な影響を測定することに注意することが重要です。したがって、Impact スコアで、特定の属性がモデルの精度が向上するかどうかを判断する

ことはできません。属性の Impact スコアが低い場合、それは必ずしも予測値への影響が少ないことを意味するわけではありません。これは、予測子が使用する他の属性よりも予測値への影響が少ないことを意味します。

## Predictor Explainability の作成

### Note

予測子ごとに最大 1 つの Predictor Explainability を作成できます

Predictor Explainability を有効にすると、Amazon Forecast は、データセット内のすべての属性の Impact スコアを計算します。Impact スコアは、影響属性が全体的な予測値に与える影響として解釈できます。予測子を作成するときに Predictor Explainability を有効にするか、予測子を作成した後に特徴を有効にすることができます。

## 新しい予測子向けの Predictor Explainability の有効化

新しい予測子を作成するときに Predictor Explainability を有効にすると、Predictor のリソースと Explainability のリソースの両方が作成されます。ソフトウェア開発キット (SDK) または Amazon Forecast コンソールを使用して、新しい予測子の Predictor Explainability を有効にできます。

### Console

Predictor Explainability を有効にするには

1. にサインイン AWS Management Console し、<https://console.aws.amazon.com/forecast/> で Amazon Forecast コンソールを開きます。
2. [Dataset groups] (データセットグループ) から、データセットグループを選択します。
3. ナビゲーションペインで、[Predictors] (予測子) を選択します。
4. [Train new predictor] (新しい予測子のトレーニング) を選択します。
5. [Predictor configuration] (予測子の設定) のセクションで、[Enable Explainability] (Explainability を有効化) を選択します。
6. 以下の必須フィールドに値を入力します。
  - 名前 - 一意の予測子名。
  - 予測頻度 - 予測の粒度。

- [Forecast horizon] (予測期間) - 予測するタイムステップの数。

## 7. [Start (開始)] を選択

### Python

SDK for Python (Boto3) で新しい予測子の説明可能性を有効にするには、`create_auto_predictor`メソッドを使用して `true` `ExplainPredictor` に設定します。

次のコードでは、今後 24 (`ForecastHorizon`) 日 (`ForecastFrequency`) の予測を行う自動予測子を作成し、`ExplainPredictor` が `true` に設定されています。必須とオプションのパラメータの詳細については、「[CreateAutoPredictor](#)」を参照してください。

```
import boto3

forecast = boto3.client('forecast')

create_predictor_response = forecast.create_auto_predictor(
    PredictorName = 'predictor_name',
    ForecastHorizon = 24,
    ForecastFrequency = 'D',
    DataConfig = {
        "DatasetGroupArn": "arn:aws:forecast:region:account:dataset-
group/datasetGroupName"
    },
    ExplainPredictor = True
)
```

## 既存の予測子向けの Predictor Explainability の有効化

既存の予測子向けに Predictor Explainability を有効にすると、そのリソース用に Explainability のリソースが作成されます。Explainability リソースがまだ含まれていない予測子についてのみ Explainability リソースを作成できます。更新されたデータセットの Impact スコアを表示するには、更新されたデータを使用して予測子を再トレーニングまたは再作成します。

ソフトウェア開発キット (SDK) または Amazon Forecast コンソールを使用して、新しい予測子の Predictor Explainability を有効にできます。

### Console

Predictor Explainability を有効にするには

1. にサインイン AWS Management Console し、<https://console.aws.amazon.com/forecast/> で Amazon Forecast コンソールを開きます。
2. [Dataset groups] (データセットグループ) から、データセットグループを選択します。
3. ナビゲーションペインで、[Predictors] (予測子) を選択します。
4. 予測子を選択します。
5. [Predictor Explainability] のセクションで、[Enable Explainability] (Explainability を有効化) を選択します。
6. Predictor Explainability の一意の名前を指定します。
7. [Start (開始)] を選択

## Python

SDK for Python (Boto3) で既存の予測子の Predictor Explainability を有効にするには、`create_explainability` メソッドを使用します。Explainability の名前と予測子の ARN を指定し、`ExplainabilityConfig` は `TimePointGranularity` と `TimeSeriesGranularity` の両方を ALL に設定します。コンソール内で表示できる Explainability ビジュアライゼーションを作成するには、`EnableVisualization` を True に設定します。

必須とオプションのパラメータの詳細については、「[CreateExplainability](#)」を参照してください。

```
import boto3

forecast = boto3.client('forecast')

create_explainability_response = forecast.create_explainability(
    ExplainabilityName = 'explainability_name',
    ResourceArn = 'arn:aws:forecast:region:accountNumber:predictor/predictorName',
    ExplainabilityConfig = {
        "TimePointGranularity": "ALL",
        "TimeSeriesGranularity": "ALL"
    },
    EnableVisualization = True
)
```

## Predictor Explainability のエクスポート

### Note

エクスポートファイルは、[Dataset Import] (データセットのインポート) から直接情報を返すことができます。これにより、インポートされたデータに式またはコマンドが含まれている場合、ファイルは CSV インジェクションに対して脆弱になります。このため、エクスポートされたファイルはセキュリティ警告を表示する可能性があります。悪意のあるアクティビティを回避するには、エクスポートされたファイルを読み取るときにリンクとマクロを無効にします。

Forecast を使用すると、Impact スコアの CSV または Parquet ファイルを S3 の場所にエクスポートできます。Impact スコアの範囲は -1 から 1 で、記号は影響の方向を示します。Amazon Forecast のソフトウェア開発キット (SDK) および Amazon Forecast コンソールを使用して、Impact スコアをエクスポートできます。

	A	B	C	D
1	Price-NormalizedImpactScore	Promotion-NormalizedImpactScore	WeatherIndex-NormalizedImpactScore	Holiday_US-NormalizedImpactScore
2	-0.97	0.1	0.87	0.23

### Console

Predictor Explainability をエクスポートするには

1. にサインイン AWS Management Console し、<https://console.aws.amazon.com/forecast/> で Amazon Forecast コンソールを開きます。
2. [Dataset groups] (データセットグループ) から、データセットグループを選択します。
3. ナビゲーションペインで、[Predictors] (予測子) を選択します。
4. 予測子を選択します。
5. [Predictor Explainability] のセクションで、[Export] (エクスポート) を選択します。
6. [Export name] (エクスポート名) のフィールドで、エクスポートの一意の名前を入力します。
7. [S3 explainability export location] (S3 の Explainability のエクスポート場所) のフィールドで、CSV ファイルをエクスポートするための S3 の場所を指定します。

- [IAM Role] (IAM ロール) のフィールドで、指定された S3 の場所へのアクセス権を持つロールを提供します。
- [Create Export] (エクスポートを作成) を選択します。

## Python

SDK for Python (Boto3) で Predictor Explainability をエクスポートするには、`create_explainability_export` メソッドを使用します。ジョブに名前を付け、Explainability の ARN を指定し、Destination オブジェクトに Amazon S3 の宛先ロケーションと IAM サービスロールを指定します。

必須とオプションのパラメータの詳細については、「[CreateExplainabilityExport](#)」を参照してください。

```
import boto3

forecast = boto3.client('forecast')

export_response = forecast.create_explainability_export(
    Destination = {
        "S3Config": {
            "Path": "s3://bucketName/filename.csv",
            "RoleArn": "arn:aws:iam::accountNumber:role/roleName"
        }
    },
    ExplainabilityArn =
'arn:aws:forecast:region:accountNumber:explainability/explainabilityName',
    ExplainabilityExportName = 'job_name'
)
```

## 制約事項とベストプラクティス

Predictor Explainability を使用する場合は、次の制約事項とベストプラクティスを考慮してください。

- Predictor Explainability は、で作成された一部の予測子でのみ使用できます AutoPredictor。AutoML または手動選択で作成されたレガシー予測子の Explainability を有効にすることはできません。「[へのアップグレード AutoPredictor](#)」を参照してください。



- Predictor Explainability はすべてのモデルで利用できるわけではありません。ARIMA (AutoRegressive 統合移動平均)、ETS (指数平滑化ステートスペースモデル)、および NPTS (非パラメトリック時系列) モデルには、外部時系列データが組み込まれていません。したがって、これらのモデルでは、追加のデータセットを含めても、説明可能性レポートは作成されません。
- Explainability には属性が必要 - 予測子には、関連する時系列、項目メタデータ、Holidays、または Weather Index の少なくとも 1 つを含める必要があります。
- 予測子は 1 つの Explainability リソースに制限されている - 1 つの予測子のために複数の Explainability リソースを作成することはできません。更新されたデータセットの Impact スコアに関心がある場合は、予測子を再トレーニングします。
- 影響スコアがゼロの場合は影響がないことが示唆されている - 属性の影響スコアが 0 の場合、その属性は予測値に大きな影響を与えません。
- 失敗した Predictor Explainability ジョブの再試行 - Forecast が Predictor を正常に作成しても、Predictor Explainability ジョブが失敗した場合、コンソールまたは CreateExplainability オペレーションで Predictor Explainability の作成を再試行できます。
- 特定の時間ポイントおよび時系列の Impact スコアの作成不可 - 特定の時間ポイントおよび時系列の Impact スコアを表示するには、「[Forecast Explainability](#)」を参照してください。
- 予測子の Explainability のビジュアライゼーションが作成後 90 日間利用可能 - 90 日後にビジュアライゼーションを表示するには、予測子を再トレーニングします。

## 予測子のモニタリング

### Note

予測子モニタリングを有効にすると、Amazon Forecast は、予想データを削除した後でも、予測子のパフォーマンス分析のために各予想のデータを保存します。このデータを削除するには、モニタリングリソースを削除します。

予測子モニタリングにより、予測子のパフォーマンスが時間の経過とともにどのように変化するかを確認できます。経済発展や顧客行動の変化など、さまざまな要因が業績の変化を引き起こす可能性があります。

例えば、ターゲットが sales であり、price および color といった関連する 2 つの属性がある予測シナリオについて考えてみます。最初の予測子を作成してから数か月以内に、特定の色が予想外に顧客の間で人気が高まる可能性があります。これにより、この属性を持つ商品の売り上げが伸びる可

能性があります。この新しいデータは、予測子のパフォーマンスと、予測子が生成する予想の精度に影響を与える可能性があります。

予測子モニタリングを有効にすると、予想を生成してより多くのデータをインポートするにつれて、予測子のパフォーマンスが Forecast によって分析されます。Forecast は、新しいデータを以前の予想と比較して、パフォーマンスの変化を検出します。Forecast コンソールでは、さまざまな精度メトリクスが時間の経過と共にどのように変化したかを示すグラフを表示できます。また、[ListMonitorEvaluations](#) オペレーションによってモニタリング結果を取得することもできます。

予測子モニタリングは、予測子を再トレーニングする時期かどうかを判断するのに役立ちます。パフォーマンスが低下している場合は、より新しいデータで予測子を再トレーニングしたほうがいいかもしれません。予測子を再トレーニングすることを選択した場合、新しい予測子には前の予測子のモニタリングデータが含まれます。また、予測子モニタリングを使用して、実稼働環境に関するコンテキストデータを収集したり、さまざまな実験の比較を行ったりすることもできます。

予測子モニタリングは AutoPredictors でのみ使用できます。既存のレガシー予測子を AutoPredictor にアップグレードできます。「[Upgrading to AutoPredictor](#)」(AutoPredictor へのアップグレード)を参照してください。

## トピック

- [予測子モニタリングワークフロー](#)
- [予測子モニタリングを有効にする](#)
- [モニタリング結果の表示](#)
- [制約事項とベストプラクティス](#)

## 予測子モニタリングワークフロー

予測子のモニタリング結果を得るには、まず予測子を使用して予測を生成し、さらにデータをインポートする必要があります。モニタリングワークフローは次のとおりです。

1. 自動予測子の予測子モニタリングを有効にします。
  - モニタリングを有効にして新しい予測子を作成します。「[新しい予測子の予測子モニタリングを有効にする](#)」を参照してください。
  - または、既存の予測子のモニタリングを有効にします。「[既存の予測子の予測子モニタリングを有効にする](#)」を参照してください。
2. 予測子を使用して 1 つ以上の予想を生成します。

- 追加のデータをインポートする。Forecast へのデータのインポートの詳細については、「[データセットのインポート](#)」を参照してください。
- 予測子のモニタリング結果を表示する：
  - 予測子の [モニタリング] タブで結果を表示できます。
  - また、[ListMonitorEvaluations](#) オペレーションによってモニタリング結果を取得することもできます。

詳細については、「[モニタリング結果の表示](#)」を参照してください。

## 予測子モニタリングを有効にする

予測子を作成するとき、または既存の予測子に対して予測子モニタリングを有効にすることができます。

### Note

予測子モニタリングは AutoPredictors でのみ使用できます。既存のレガシー予測子を AutoPredictor にアップグレードできます。「[Upgrading to AutoPredictor](#)」(AutoPredictor へのアップグレード)を参照してください。

### トピック

- [新しい予測子の予測子モニタリングを有効にする](#)
- [既存の予測子の予測子モニタリングを有効にする](#)

## 新しい予測子の予測子モニタリングを有効にする

コンソール、AWS CLI、AWS SDK、および [CreateAutoPredictor](#) 操作を使用して、新しい予測子の予測子モニタリングを有効にできます。

### Console

予測子モニタリングを有効にするには

- AWS Management Console にサインインして、Amazon Forecast コンソール (<https://console.aws.amazon.com/forecast/>) を開きます。
- [Dataset groups] (データセットグループ) から、データセットグループを選択します。

3. ナビゲーションペインで、[Predictors] (予測子) を選択します。
4. [Train new predictor] (新しい予測子のトレーニング) を選択します。
5. [予測子の設定] のセクションで、[モニタリングを有効化] を選択します。
6. 以下の必須フィールドに値を入力します。
  - 名前 - 一意の予測子名。
  - 予測頻度 - 予測の粒度。
  - [Forecast horizon] (予測期間) - 予測するタイムステップの数。
7. [開始] を選択して、モニタリングを有効にした自動予測子を作成します。予測子を使用して予測を生成し、さらにデータをインポートすると、モニタリング結果が表示されます。

## Python

SDK for Python (Boto3) で新しい予測子の予測子モニタリングを有効にするには、`create_auto_predictor` メソッドを使用し、`MonitoringConfig` にモニター名を指定します。

次のコードでは、今後 24 (`ForecastHorizon`) 日 (`ForecastFrequency`) の予測を行う自動予測子を作成し、`MyPredictorMonitor` が `MonitorName` として指定されています。予想を生成し、さらにデータをインポートすると、予測子モニタリング結果が表示されます。結果の取得の詳細については、「[モニタリング結果の表示](#)」を参照してください。

予測子作成の必須とオプションのパラメータの詳細については、「[CreateAutoPredictor](#)」を参照してください。

```
import boto3

forecast = boto3.client('forecast')

create_predictor_response = forecast.create_auto_predictor(
    PredictorName = 'predictor_name',
    ForecastHorizon = 24,
    ForecastFrequency = 'D',
    DataConfig = {
        "DatasetGroupArn": "arn:aws:forecast:region:account:dataset-
group/datasetGroupName"
    },
    MonitorConifg = {
        "MonitorName": "MyMonitorName"
    }
)
```

```
)
```

## 既存の予測子の予測子モニタリングを有効にする

コンソール、AWS CLI、AWS SDK を使用して、既存の予測子の予測子モニタリングを有効にできます。

### Console

予測子モニタリングを有効にするには

1. AWS Management Console にサインインして、Amazon Forecast コンソール (<https://console.aws.amazon.com/forecast/>) を開きます。
2. [Dataset groups] (データセットグループ) から、データセットグループを選択します。
3. ナビゲーションペインで、[Predictors] (予測子) を選択します。
4. 予測子を選択します。
5. [モニタリング] タブに移動します。
6. [モニタリングの詳細] セクションで、[モニタリングを開始] を選択します。

[モニタリングステータス] が [アクティブ] の場合、予測子モニタリングが有効です。予想を生成し、さらにデータをインポートすると、予測子モニタリング結果が表示されます。詳細については、「[モニタリング結果の表示](#)」を参照してください。

### Python

SDK for Python (Boto3) で既存の予測子の予測子モニタリングを有効にするには、`create_monitor` メソッドを使用します。モニタリングの名前を指定し、`ResourceArn` にモニタリングする予測子の Amazon リソースネーム (ARN) を指定します。`describe_monitor` メソッドを使用してモニター ARN を指定し、モニターのステータスを取得します。予想を生成し、さらにデータをインポートすると、予測子モニタリング結果が表示されます。詳細については、「[モニタリング結果の表示](#)」を参照してください。

必須とオプションのパラメータの詳細については、「[CreateMonitor](#)」と「[DescribeMonitor](#)」を参照してください。

```
import boto3
```

```
forecast = boto3.client('forecast')

create_monitor_response = forecast.create_monitor(
    MonitorName = 'monitor_name',
    ResourceArn = 'arn:aws:forecast:region:accountNumber:predictor/predictorName'
)

monitor_arn = create_monitor_response['MonitorArn']

describe_monitor_response = forecast.describe_monitor(
    MonitorArn = monitor_arn
)

print("Monitor status: " + describe_monitor_response['Status'])
```

## モニタリング結果の表示

予想を生成し、さらにデータをインポートすると、予測子モニタリング結果が表示されます。Forecast コンソールを使用して結果を可視化することも、[ListMonitorEvaluations](#) オペレーションを使用して結果をプログラマ的に取得することもできます。

Forecast コンソールには、各[予測子メトリクス](#)の結果のグラフが表示されます。グラフには、予測子と予測子のイベント (再トレーニングなど) の存続期間における各メトリクスの変化が含まれます。

[ListMonitorEvaluations](#) オペレーションでは、さまざまな期間のメトリクス結果と予測イベントが返されます。

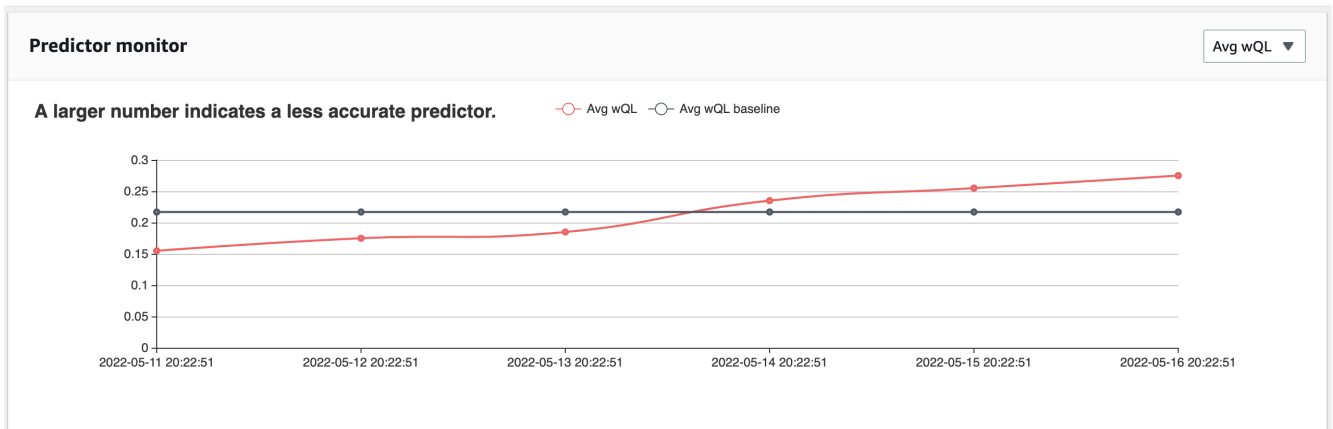
### Console

予測子のモニタリング結果を表示するには

1. AWS Management Console にサインインして、Amazon Forecast コンソール (<https://console.aws.amazon.com/forecast/>) を開きます。
  2. [Dataset groups] (データセットグループ) から、データセットグループを選択します。
  3. ナビゲーションペインで、[Predictors] (予測子) を選択します。
  4. 予測子を選び、[モニタリング] タブを選択します。
- [モニタリング結果] セクションには、さまざまな精度メトリクスが時間の経過とともにどのように変化したかが示されます。ドロップダウンリストを使用して、グラフが追跡する指標を変更します。

- [モニタリング履歴] セクションには、結果で追跡されたさまざまなイベントの詳細が一覧表示されます。

以下は、予測子の Avg wQL スコアが時間の経過とともにどのように変化したかを示すグラフの例です。このグラフでは、Avg wQL 値が時間とともに増加していることがわかります。この増加は、予測子の精度が低下していることを示しています。この情報を使用して、モデルを再検証して対策を講じる必要があるかどうかを判断します。



## SDK for Python (Boto3)

SDK for Python (Boto3) でモニタリング結果を取得するには、`list_monitor_evaluations` メソッドを使用します。モニターの Amazon リソースネーム (ARN) を指定し、オプションで取得する結果の最大数を `MaxResults` パラメータで指定します。オプションで `Filter` を指定して結果をフィルタリングします。SUCCESS または FAILURE のいずれかの `EvaluationState` で評価をフィルタリングできます。次のコードでは、モニタリング評価が最大 20 回成功します。

```
import boto3

forecast = boto3.client('forecast')

monitor_results = forecast.list_monitor_evaluations(
    MonitorArn = 'monitor_arn',
    MaxResults = 20,
    Filters = [
        {
            "Condition": "IS",
            "Key": "EvaluationState",
            "Value": "SUCCESS"
        }
    ]
)
```

```
]
)
print(monitor_results)
```

以下に、JSON レスポンスの例を示します。

```
{
  "NextToken": "string",
  "PredictorMonitorEvaluations": [
    {
      "MonitorArn": "MonitorARN",
      "ResourceArn": "PredictorARN",
      "EvaluationTime": "2020-01-02T00:00:00Z",
      "EvaluationState": "SUCCESS",
      "WindowStartDatetime": "2019-01-01T00:00:00Z",
      "WindowEndDatetime": "2019-01-03T00:00:00Z",
      "PredictorEvent": {
        "Detail": "Retrain",
        "Datetime": "2020-01-01T00:00:00Z"
      },
      "MonitorDataSource": {
        "DatasetImportJobArn": "arn:aws:forecast:region:accountNumber:dataset-import-job/*",
        "ForecastArn": "arn:aws:forecast:region:accountNumber:forecast/*",
        "PredictorArn": "arn:aws:forecast:region:accountNumber:predictor/*",
      },
      "MetricResults": [
        {
          "MetricName": "AverageWeightedQuantileLoss",
          "MetricValue": 0.17009070456599376
        },
        {
          "MetricName": "MAPE",
          "MetricValue": 0.250711322309796
        },
        {
          "MetricName": "MASE",
          "MetricValue": 1.6275608734888485
        },
        {
```



```
    "MetricName": "RMSE",
    "MetricValue": 3100.7125081405547
  },
  {
    "MetricName": "WAPE",
    "MetricValue": 0.17101159704738722}
  ]
}
]
```

## 制約事項とベストプラクティス

予測子モニタリングを使用する場合は、次の制約事項とベストプラクティスを考慮してください。

- 予測子モニタリングは自動予測子でのみ使用可能 - AutoML で、または手動選択を通じて作成されたレガシー予測子のモニタリングを有効にすることはできません。「[Upgrading to AutoPredictor](#)」(AutoPredictor へのアップグレード) を参照してください。
- 予測子モニタリングは自動予測子ごとに異なる - 自動予測子ごとに作成できるモニターは 1 つだけです。
- 予測子モニタリングには新しいデータと予測の生成が必要 — 新しい予測の生成に使用される新しいデータをインポートすると、予測子モニタリングの結果が利用可能になります。新しいデータをインポートしない場合や、新しくインポートしたデータが予測期間全体を網羅していない場合、モニタリング結果は表示されません。
- 予測子モニタリングには新しい予測が必要 — モニタリング結果を生成するには、新しい予測を継続的に生成する必要があります。新しい予測を生成しなければ、モニタリング結果は表示されません。
- Amazon Forecast は、予測子のパフォーマンス分析のために各予想のデータを保存 - 予測を削除しても、Forecast でこれらのデータは保存されます。これらのデータを削除するには、関連するモニターを削除します。
- [StopResource](#) オペレーションにより、現在の評価と将来の評価がすべて停止されます。
- avgWQL メトリクスは、平均以外の分位数の予想を生成する場合にのみ使用できます。
- 進行中のモニター評価は [ListMonitorEvaluations](#) オペレーションには表示されません。

# Amazon Forecast アルゴリズム

Amazon Forecast の予測子は、アルゴリズムを使用して、時系列データセットを使用してモデルをトレーニングします。その後、トレーニングされたモデルを使用して、メトリクスと予測を生成します。

モデルのトレーニングに使用するアルゴリズムがわからない場合は、予測子を作成するときに AutoML を選択し、Forecast がデータセットに最適なモデルをトレーニングするようにします。それ以外の場合は、Amazon Forecast のいずれかのアルゴリズムを手動で選択できます。

## Python ノートブック

AutoML の使用に関するステップバイステップガイドについては、「[Getting Started with AutoML](#)」(AutoML の開始方法)を参照してください。

## ビルトイン Forecast アルゴリズム

Amazon Forecast では、6 つのビルトインアルゴリズムが用意されており、それらの中から選択できます。これらは、自己回帰和分移動平均 (ARIMA) のような一般的に使用される統計アルゴリズムから、CNN-QR や DeepAR+ のような複雑なニューラルネットワークアルゴリズムにまで及びます。

### [CNN-QR](#)

```
arn:aws:forecast:::algorithm/CNN-QR
```

Amazon Forecast CNN-QR、畳み込みニューラルネットワーク - 分位点回帰は、因果畳み込みニューラルネットワーク (CNN) を使用して時系列を予測するための独自の機械学習アルゴリズムです。CNN-QR は、数百の時系列を含む大規模なデータセットで最適に機能します。項目のメタデータを受け入れ、将来の値なしで関連する時系列データを受け入れる唯一の Forecast のアルゴリズムです。

### [DeepAR+](#)

```
arn:aws:forecast:::algorithm/Deep_AR_Plus
```

Amazon Forecast DeepAR は、再帰型ニューラルネットワーク (RNN) を使用して時系列を予測するための、独占的な機会学習アルゴリズムです。DeepAR+ は、数百の特徴時系列を含む大規模なデー

タセットで最適に機能します。このアルゴリズムは、フォワードルックの関連する時系列と項目メタデータを受け入れます。

## [Prophet](#)

```
arn:aws:forecast:::algorithm/Prophet
```

Prophet は、非線形トレンドが年次、週次、および日次の季節性に適合する加法モデルに基づく時系列予測アルゴリズムです。これは、季節性の強い効果といくつかの季節の履歴データを持つ時系列で最適に機能します。

## [NPTS](#)

```
arn:aws:forecast:::algorithm/NPTS
```

Amazon Forecast ノンパラメトリック時系列 (NPTS) の独占的なアルゴリズムは、スケーラブルで確率的なベースライン予測機能です。NPTS は、スパースまたは断続的な時系列を使用する場合に特に役立ちます。Forecast は、Standard NPTS、Seasonal NPTS、Climatological Forecaster、および Seasonal Climatological Forecaster の 4 つのアルゴリズムバリエーションを提供します。

## [ARIMA](#)

```
arn:aws:forecast:::algorithm/ARIMA
```

自己回帰和分移動平均 (ARIMA) は、時系列予測に一般的に使用される統計アルゴリズムです。このアルゴリズムは、時系列が 100 未満の単純なデータセットに特に役立ちます。

## [ETS](#)

```
arn:aws:forecast:::algorithm/ETS
```

指数平滑法 (ETS) は、時系列予測に一般的に使用される統計アルゴリズムです。このアルゴリズムは、時系列が 100 未満の単純なデータセット、および季節性パターンのあるデータセットに特に役立ちます。ETS は、時系列データセット内のすべての観測値の加重平均を予測として計算し、時間の経過とともに重みを指数関数的に減少させます。

## Forecast アルゴリズムの比較

次の表を使用して、時系列データセットに最適なオプションを見つけます。

	ニューラルネットワーク		柔軟なローカルアルゴリズム	ベースラインアルゴリズム		
	CNN-QR	DeepAR+	Prophet	NPTS	ARIMA	ETS
計算負荷の高いトレーニングプロセス	高	高	中	低	低	低
履歴の関連する時系列を受け入れる *						
フォワードルックの関連する時系列を受け入れる *						
項目のメタデータ (製品の色、ブランドなど) を受け入れる						
Weather Index のビルトインの特徴量化を受け入れる						
スパースデータセットに適している						
ハイパーパラメータ最適化 (HPO) を実行する						
デフォルトのハイパーパラメータ値の上書きを許可する						

\* 関連する時系列の詳細については、「[Related Time Series](#)」(関連する時系列)を参照してください。

## 自己回帰和分移動平均 (ARIMA) のアルゴリズム

自己回帰和分移動平均 ([ARIMA](#)) は、時系列予測に一般的に使用される局所統計アルゴリズムです。ARIMA は、入力データセットの標準的時間構造 (時間のパターン化された編成) をキャプチャします。Amazon Forecast の ARIMA アルゴリズムは、Comprehensive R Archive Network (CRAN) の Package 'forecast' の [Arima 関数](#) を呼び出します。

### ARIMA の仕組み

ARIMA アルゴリズムは、定常時系列にマッピングできるデータセットに特に役立ちます。定常時系列の統計プロパティ (自己相関など) は、時間に依存しません。定常時系列を持つデータセットには通常、信号とノイズの組み合わせが含まれます。信号は正弦波振動のパターンを示すか、または季節的な構成要素を含む可能性があります。ARIMA は、信号をノイズから分離するフィルターのように動作し、今後の信号を推定して予測を立てます。

### ARIMA のハイパーパラメータおよびチューニング

ARIMA ハイパーパラメータとチューニングの詳細については、[CRAN](#) の「[Package 'forecast'](#)」(パッケージ「予測」) の Arima 関数のドキュメントを参照してください。

Amazon Forecast は、次の表を使用して、[CreateDataset](#) オペレーションで指定された DataFrequency パラメータを R [ts](#) 関数の frequency パラメータに変換します。

DataFrequency (文字列)	R ts frequency (整数)
Y	1
M	12
W	52
D	7
H	24
30 分	2

DataFrequency (文字列)	R ts frequency (整数)
15 分	4
10 分	6
5 分	12
1 min	60

24 未満の頻度または短い時系列の場合、ハイパーパラメータは [CRAN](#) の Package 'forecast' の `auto.arima` 関数を使用して設定されます。24 以上の頻度または長い時系列の場合は、「[Forecasting with long seasonal periods](#)」で説明するように、 $K = 4$  のフーリエ級数を使用します。

表にないサポートされているデータ頻度のデフォルト値は、ts 頻度 1 に設定されます。

## CNN-QR アルゴリズム

Amazon Forecast CNN-QR、畳み込みニューラルネットワーク - 分位点回帰は、因果畳み込みニューラルネットワーク (CNN) を使用してスカラー (1 次元) 時系列を予測するための独自の機械学習アルゴリズムです。この教師あり学習アルゴリズムは、時系列の大規模なコレクションから 1 つのグローバルモデルをトレーニングし、分位数デコーダーを使用して確率的予測を行います。

### トピック

- [CNN-QR の開始方法](#)
- [CNN-QR の仕組み](#)
- [CNN-QR での関連データの使用](#)
- [CNN-QR ハイパーパラメータ](#)
- [ヒントとベストプラクティス](#)

### CNN-QR の開始方法

CNN-QR を使用して予測子をトレーニングするには、次の 2 つの方法があります。

1. CNN-QR アルゴリズムを手動で選択します。
2. AutoML の選択 (CNN-QR は AutoML の一部です)。

使用するアルゴリズムがわからない場合は、AutoML を選択することをお勧めします。これにより、Forecast は、データに対して最も正確なアルゴリズムである場合に CNN-QR を選択します。CNN-QR が最も正確なモデルとして選択されたかどうかを確認するには、[DescribePredictor](#) API を使用するか、コンソールで予測子名を選択します。

CNN-QR の主なユースケースは次のとおりです。

- 大規模で複雑なデータセットを使用した予測 - CNN-QR は、大規模で複雑なデータセットを使用してトレーニングした場合に最適に機能します。ニューラルネットワークは多くのデータセットから学習できます。これは、関連する時系列と項目メタデータがある場合に役立ちます。
- 履歴の関連する時系列を使用した予測 - CNN-QR では、関連する時系列に予測期間内のデータポイントを含める必要はありません。この追加された柔軟性により、商品の価格、イベント、ウェブメトリクス、製品カテゴリなど、関連する時系列および項目メタデータを幅広く含めることができます。

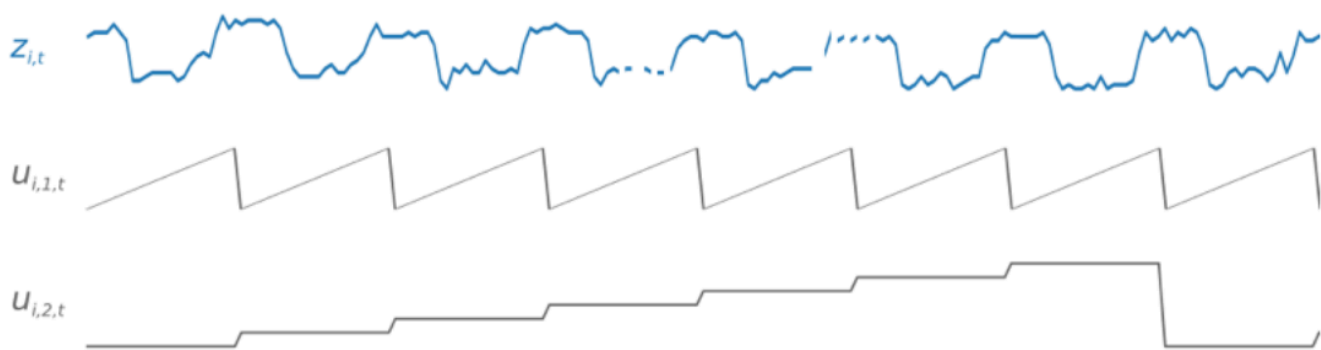
## CNN-QR の仕組み

CNN-QR は、確率的予測のためのシーケンス間 (Seq2Seq) モデルであり、エンコードシーケンスを条件として、予測がデコードシーケンスをどれだけ適切に再構築するかをテストします。

このアルゴリズムでは、エンコードシーケンスとデコードシーケンスでさまざまな特徴を使用できるため、エンコーダーで関連する時系列を使用し、デコーダーから除外することができます (その逆も可能です)。デフォルトでは、予測期間内のデータポイントに関連する時系列は、エンコーダーとデコーダーの両方に含まれます。予測期間にデータポイントがない関連する時系列は、エンコーダーのみ含まれます。

CNN-QR は、学習可能な特徴抽出機能として使用できる階層的因果 CNN を使用して分位点回帰を実行します。

週末の急増などの時間依存パターンの学習を容易にするために、CNN-QR は時系列の詳細度に基づいて特徴の時系列を自動的に作成します。例えば、CNN-QR は、週に 1 回の頻度で 2 つの特徴の時系列 (月初からの日付と年初からの日付) を作成します。アルゴリズムは、このように派生した特徴の時系列を使用するだけでなく、トレーニングと推論の間にユーザーが提供するカスタムの特徴の時系列も使用します。次の例は、ターゲット時系列である  $z_{i,t}$  と、派生した 2 つの時系列の特徴を示しています。 $u_{i,1,t}$  はその日の時間を表し、 $u_{i,2,t}$  はその曜日を表します。



CNN-QR は、データの頻度とトレーニングデータのサイズに基づいて、これらの特徴の時系列を自動的に組み込みます。次の表に、サポートされている基本的な頻度ごとに、派生させることのできる特徴を示します。

時系列の頻度	派生する特徴
分	分、時、曜日、月初からの日付、年初からの日付
時間	時、曜日、月初からの日付、年初からの日付
日	曜日、月初からの日付、年初からの日付
週	月初からの週、年初からの週
月	月

トレーニング中、トレーニングデータセットの各時系列は、固定された事前定義済みの長さを持つ、隣接するコンテキストウィンドウと予測ウィンドウのペアで構成されます。これを次の図に示します。コンテキストウィンドウは緑色で表され、予測ウィンドウは青色で表されています。

特定のトレーニングセットでトレーニングされたモデルを使用すると、トレーニングセット内の時系列の予測と他の時系列の予測を生成できます。トレーニングデータセットは、関連する時系列と項目メタデータのリストに関連付けられている可能性のあるターゲット時系列で構成されています。

以下の図は、インデックス  $i$  が付されたトレーニングデータセットの要素に対してこれがどのように機能するかを示しています。トレーニングデータセットは、ターゲット時系列  $z_{i,t}$ 、および 2 つの関連付けられた関連する時系列  $x_{i,1,t}$  と  $x_{i,2,t}$  から構成されます。最初の関連する時系列である  $x_{i,1,t}$  はフォワードルックの時系列であり、2 番目の時系列である  $x_{i,2,t}$  は履歴時系列です。





CNN-QR は、ターゲット時系列である  $z_{i,t}$  と、関連する時系列である  $x_{i,1,t}$  および  $x_{i,2,t}$  を学習して、予測ウィンドウで予測を生成します。これは、オレンジ色の線で表されます。

## CNN-QR での関連データの使用

CNNQR では、過去と将来の時系列データセットの両方がサポートされます。将来を見据えた関連時系列データセットを提供する場合、欠損値は[将来のフィル手法](#)を使用して埋められます。履歴およびフォワードルックの関連する時系列の詳細については、「[Using Related Time Series Datasets](#)」(関連する時系列データセットの使用)を参照してください。

CNN-QR で項目メタデータデータセットを使用することもできます。これらは、ターゲット時系列の項目に関する静的情報を含むデータセットです。項目メタデータは、履歴データがほとんどないか、まったくないコールドスタート予想シナリオで特に役立ちます。項目メタデータの詳細については、「[Item Metadata](#)」(項目メタデータ)を参照してください。

## CNN-QR ハイパーパラメータ

Amazon Forecast は、選択したハイパーパラメータで CNN-QR モデルを最適化します。CNN-QR を手動で選択する場合、これらのハイパーパラメータのトレーニングパラメータを渡すオプションがあります。次の表に、CNN-QR アルゴリズムの調整可能なハイパーパラメータを示します。

Parameter Name	値	説明
context_length	有効な値 正の整数	予測を生成する前にモデルが読み込む時間ポイントの数。通常、CNN-QR はそれ以上の履歴データを調べるためにラグを使用しないため、CNN-QR の context_length の値は DeepAR+ よりも大きくなります。

Parameter Name	値	説明
	有効範囲 1 ~ 500 一般的な値 $2 * \text{ForecastHorizon}$ から $12 * \text{ForecastHorizon}$ 調整可能な HPO Yes	<code>context_length</code> の値が事前定義された範囲外の場合、CNN-QR は、デフォルトの <code>context_length</code> を適切な値に自動的に設定します。
<code>use_related_data</code>	有効な値 ALL NONE HISTORICAL FORWARD_LOOKING デフォルト値 ALL 調整可能な HPO Yes	モデルに含める関連する時系列データの種類を決定します。 次の 4 つのオプションのいずれかを選択します。 <ul style="list-style-type: none"> <li>ALL: 提供されたすべての関連する時系列を含めます。</li> <li>NONE: 提供されているすべての関連する時系列を除外します。</li> <li>HISTORICAL : 予測範囲にまで及ばない関連する時系列のみを含めます。</li> <li>FORWARD_LOOKING : 予測範囲にまで及ぶ関連する時系列のみを含めます。</li> </ul> HISTORICAL はすべての履歴の関連する時系列を含み、FORWARD_LOOKING はすべてのフォワードルックの関連する時系列を含みます。HISTORICAL または FORWARD_LOOKING の関連する時系列のサブセットを選択することはできません。

Parameter Name	値	説明
use_item_metadata	有効な値	モデルに項目のメタデータが含まれるかどうかを決定します。
	ALL	2つのオプションのいずれかを選択します。
	NONE	
	デフォルト値	<ul style="list-style-type: none"> <li>ALL: 提供されたすべての項目メタデータを含めます。</li> <li>NONE: 提供されたすべての項目メタデータを除外します。</li> </ul>
	ALL	use_item_metadata は、提供されたすべての項目メタデータを含むか、何も含まないかのいずれかです。項目メタデータのサブセットを選択することはできません。
調整可能な HPO		
epochs	有効な値	トレーニングデータの完全なパスの最大数。データセットが小さいほど、より多くのエポックが必要になります。
	正の整数	ForecastHorizon と context_length の値が大きい場合は、エポックを減らしてトレーニング時間を改善することを検討してください。
	一般的な値	
	10 ~ 1000	100
	デフォルト値	
調整可能な HPO	No	

## ハイパーパラメータの最適化 (HPO)

ハイパーパラメータの最適化 (HPO) は、特定の学習目標に最適なハイパーパラメータ値を選択するタスクです。Forecast を使用すると、このプロセスを2つの方法で自動化できます。

1. AutoML を選択すると、CNN-QR に対して HPO が自動的に実行されます。
2. CNN-QR を手動で選択して PerformHPO = TRUE を設定します。

関連する時系列と項目のメタデータを追加しても、CNN-QR モデルの精度が常に向上するとは限りません。AutoML を実行するか、HPO を有効にすると、CNN-QR は、提供された関連する時系列と項目メタデータがある場合とない場合のモデルの精度をテストし、最も精度の高いモデルを選択します。

Amazon Forecast は、HPO 中に次の 3 つのハイパーパラメータを自動的に最適化し、最終的にトレーニングされた値を提供します。

- `context_length` - ネットワークが参照できる過去の長さを決定します。HPO プロセスは、トレーニング時間を考慮しながら、モデルの精度を最大化する `context_length` の値を自動的に設定します。
- `use_related_data` - モデルに含める関連する時系列データの形式を決定します。HPO プロセスは、関連する時系列データがモデルを改善するかどうかを自動的にチェックし、最適な設定を選択します。
- `use_item_metadata` - モデルに項目メタデータを含めるかどうかを決定します。HPO プロセスは、項目メタデータがモデルを改善するかどうかを自動的にチェックし、最適な設定を選択します。

#### Note

Holiday の補足的な特徴が選択されている場合に `use_related_data` が NONE もしくは HISTORICAL に設定されているときは、これは祝祭日データを含めてもモデルの精度が向上しないことを意味します。

`PerformHPO = TRUE` を手動選択中に設定した場合は、`context_length` ハイパーパラメータの HPO 構成を設定できます。ただし、AutoML を選択した場合、HPO 設定のどの側面も変更できません。HPO 設定の詳細については、[IntergerParameterRange](#) API を参照してください。

## ヒントとベストプラクティス

ForecastHorizon に大きな値を使用しないようにする - ForecastHorizon に 100 を超える値を使用すると、トレーニング時間が長くなり、モデルの精度が低下する可能性があります。未来の予測をさらに生成するには、より高い頻度で情報を集約することを検討してください。たとえば、1min ではなく 5min を使用します。

CNN はより長いコンテキスト長を可能にする - CNN-QR を使用することで、`context_length` を DeepAR+ よりもわずかに長く設定できます。これは、CNN が一般に RNN よりも効率的であることによります。

関連データの特徴エンジニアリング - モデルをトレーニングするときに、関連する時系列と項目メタデータのさまざまな組み合わせを試して、追加情報によって精度が向上するかどうかを評価します。関連する時系列と項目のメタデータのさまざまな組み合わせと変換により、さまざまな結果が得られます。

CNN-QR は平均分位数で予測しない - [CreateForecast](#) API で ForecastTypes を mean に設定すると、代わりに分位数の中央値 (0.5 または P50) で予測が生成されます。

## DeepAR+ アルゴリズム

Amazon Forecast DeepAR は、再帰型ニューラルネットワーク (RNN) を使用してスカラー (1 次元) 時系列を予測するための、教師あり学習アルゴリズムです。自己回帰和分移動平均 (ARIMA) や指数平滑法 (ETS) などの古典的な予測方法は、1 つのモデルを個々の時系列に適合させ、そのモデルを使用して時系列を未来に当てはめます。ただし多くのアプリケーションでは、一連の横断的な単位にわたって同様の時系列が多数あります。これらの時系列グループは、さまざまな製品、サーバーの負荷、およびウェブページのリクエストを要求します。この場合、これらすべての時系列で、連携して単一のモデルをトレーニングすることが効果的です。DeepAR+ はこのアプローチを使用します。データセットに何百もの特徴の時系列データが含まれている場合、DeepAR+ アルゴリズムは標準の ARIMA や ETS メソッドよりも優れています。トレーニングしたモデルを使用して、トレーニングしたモデルと似た新しい時系列の予測を生成することもできます。

### Python ノートブック

DeepAR+ アルゴリズムの使用に関するステップバイステップガイドについては、「[Getting Started with DeepAR+](#)」(DeepAR+ の開始方法) を参照してください。

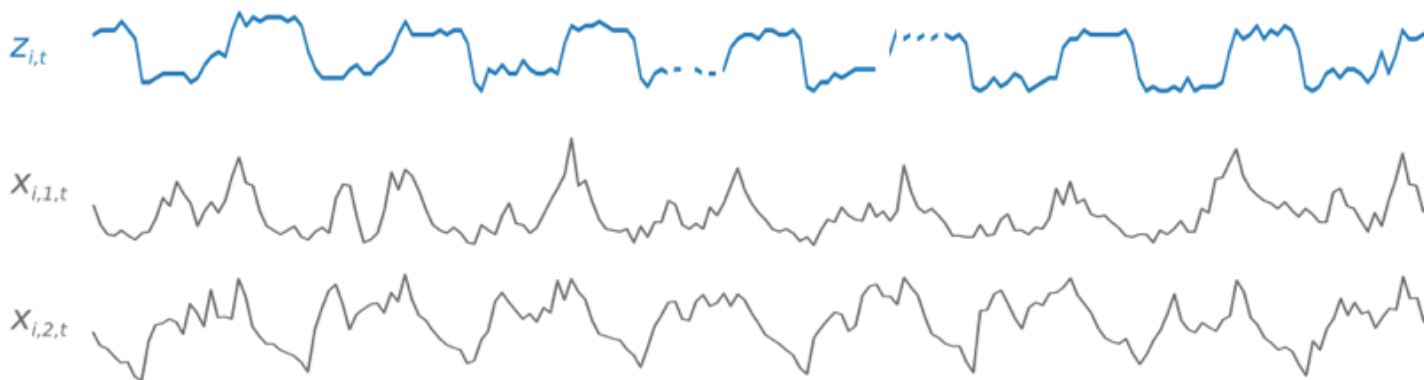
### トピック

- [DeepAR+ の仕組み](#)
- [DeepAR+ ハイパーパラメータ](#)
- [DeepAR+ モデルを調整する](#)

## DeepAR+ の仕組み

トレーニング中、DeepAR+ はトレーニングデータセットとオプションのテストデータセットを使用します。テストデータセットは、トレーニングされたモデルを評価するために使用されます。一般的に、トレーニングデータセットとテストデータセットに同じ一連の時系列を含める必要はありません。

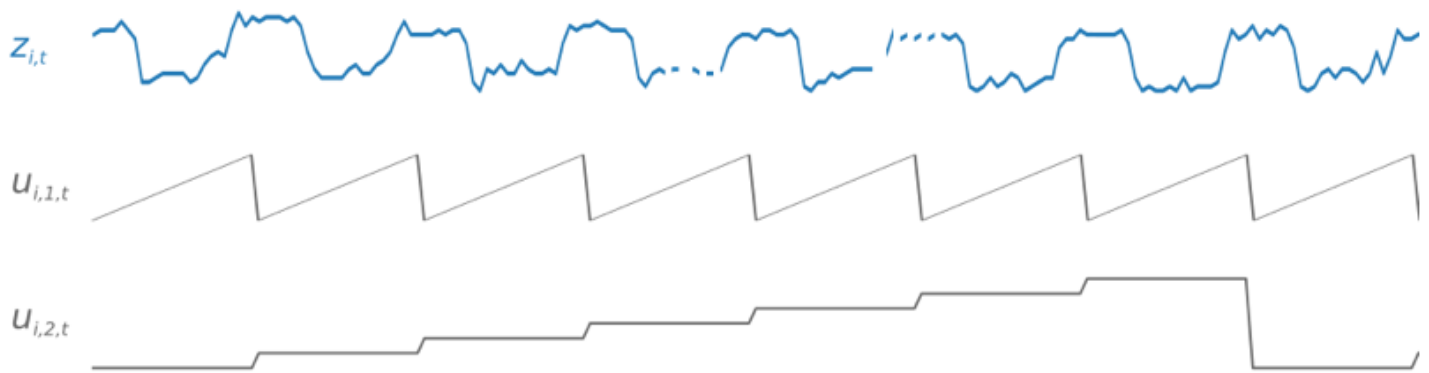
ません。特定のトレーニングセットでトレーニングされたモデルを使用すると、トレーニングセット内の時系列の今後の予測と他の時系列の予測を生成できます。トレーニングデータセットとテストデータセットはどちらも (可能な場合は複数の) ターゲット時系列で構成されています。オプションで、特徴の時系列のベクトルとカテゴリの特徴のベクトルに関連付けることができます (詳細については、SageMaker デベロッパーガイドの「[DeepAR Input/Output Interface](#)」(DeepAR の入出力インターフェイス) を参照してください)。次の例は、インデックス  $i$  が付されたトレーニングデータセットの要素に対してこれがどのように機能するかを示しています。トレーニングデータセットは、ターゲット時系列  $z_{i,t}$ 、および 2 つの関連する特徴の時系列  $x_{i,1,t}$  と  $x_{i,2,t}$  から構成されます。



ターゲット時系列には欠落した値が含まれている可能性があります (グラフでは時系列の切れ目で示されています)。DeepAR+ は、今後認識される特徴の時系列のみをサポートします。これにより、事実と反した「what-if」シナリオを実行できます。たとえば、「製品の価格を何らかの方法で変更した場合はどうなりますか?」などです。

各ターゲット時系列は、複数のカテゴリ別特徴に関連付けることもできます。これらを使用して、時系列が特定のグループに属していることをエンコードできます。カテゴリ別特徴を使用すると、モデルはそれらのグループ化の典型的な動作を学習することができ、それによって精度が向上します。モデルはこれを実装するために、グループ内のすべての時系列の共通プロパティをキャプチャする各グループの埋め込みベクトルを学習します。

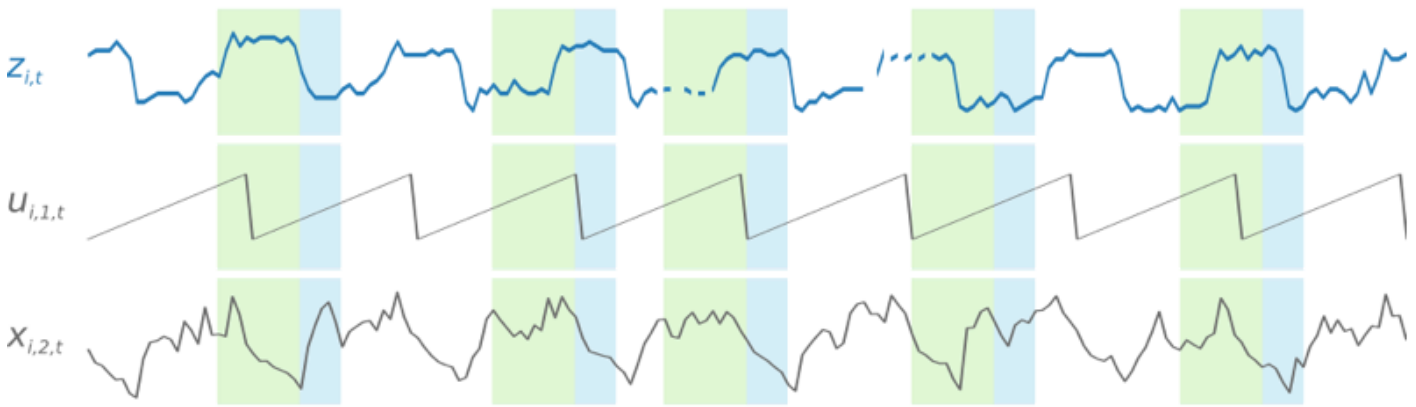
週末の急増などの時間依存パターンの学習を容易にするために、DeepAR+ は時系列の詳細度に基づいて特徴の時系列を自動的に作成します。たとえば、DeepAR+ は、週に 1 回の頻度で 2 つの特徴の時系列 (月初からの日付と年初からの日付) を作成します。DeepAR+ は、このように派生した特徴の時系列を使用するだけでなく、トレーニングと推論の間にユーザーが提供するカスタムの特徴の時系列も使用します。次の例は、派生した 2 つの時系列の特徴を示しています。 $u_{i,1,t}$  はその日の時間を表し、 $u_{i,2,t}$  はその曜日を表します。



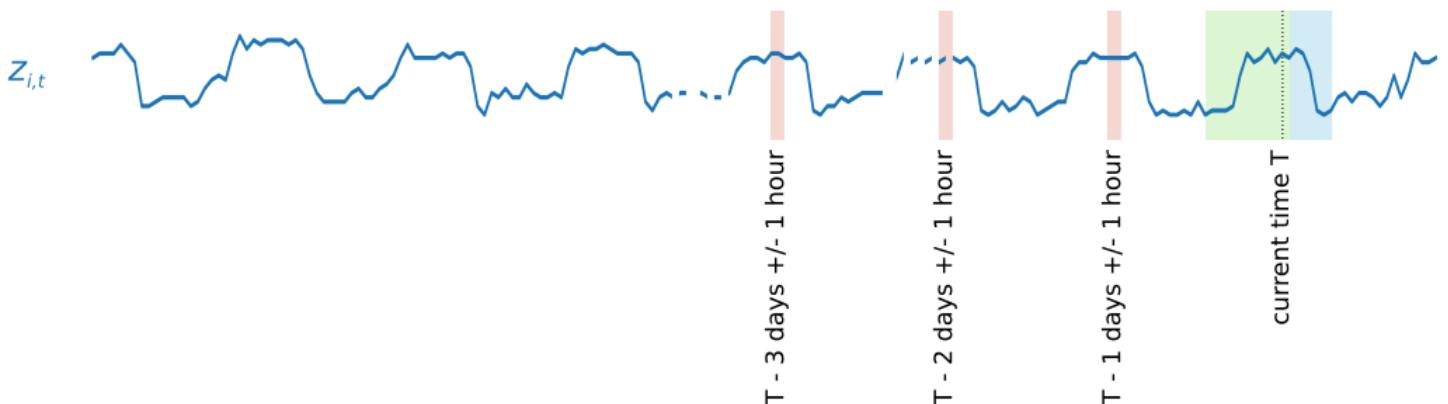
DeepAR+ は、データの頻度とトレーニングデータのサイズに基づいて、これらの特徴の時系列を自動的に組み込みます。次の表に、サポートされている基本的な頻度ごとに、派生させることのできる特徴を示します。

時系列の頻度	派生する特徴
分	分、時、曜日、月初からの日付、年初からの日付
時間	時、曜日、月初からの日付、年初からの日付
日	曜日、月初からの日付、年初からの日付
週	月初からの週、年初からの週
月	月

DeepAR+ モデルは、トレーニングデータセットの各時系列からいくつかのトレーニング例をランダムにサンプリングすることによってトレーニングされます。各トレーニング例は、事前定義された固定長を持つ対の隣接コンテキストと予測ウィンドウで構成されています。context\_length ハイパーパラメータは、どの程度の過去まで遡ってネットワークを調べられるのかを制御し、ForecastHorizon パラメータは、どの程度の未来まで予測を生成できるかを制御します。トレーニング中、Amazon Forecast は、指定された予測長より時系列が短いトレーニングデータセット内の要素を無視します。次の例は、要素  $i$  から抽出された、コンテキストの長さ (緑でハイライト表示) が 12 時間、予測の長さ (青でハイライト表示) が 6 時間の 5 つのサンプルを示しています。簡潔にするために、特徴の時系列  $x_{i,1,t}$  と  $u_{i,2,t}$  は除外しました。



季節的パターンをキャプチャするために、DeepAR+ はターゲット時系列から遅延した (過去の期間の) 値も自動的にフィードします。1 時間に 1 回の頻度で各時間インデックス  $t = T$  のサンプルを取得したこの例では、モデルは  $z_{i,t}$  値を公開します。これらは過去約 1、2、および 3 日 (ピンクでハイライト表示) に発生したものです。



トレーニングされたモデルは、推論のために、トレーニング中に使用されたかどうかにかかわらず、ターゲット時系列を入力として受け取り、それ以降の ForecastHorizon 値の確率分布を予測します。DeepAR+ はデータセット全体についてトレーニングされるため、予測では同様の時系列からの学習パターンが考慮されます。

DeepAR+ の背後にある数学の詳細については、コーネル大学ライブラリウェブサイトの [DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks \(DeepAR: 自己回帰反復ネットワークによる確率予測\)](#) を参照してください。

## DeepAR+ ハイパーパラメータ

次の表は、DeepAR+ アルゴリズムで使用できるハイパーパラメータのリストです。太字のパラメータは、ハイパーパラメータ最適化 (HPO) に関与します。



Parameter Name	説明
<b>context_length</b>	<p>予測を生成する前にモデルが読み込む時間ポイントの数。このパラメータの値は、ForecastHorizon とほぼ同じである必要があります。モデルはターゲットからの遅延入力も受信するため、context_length は一般的な季節性よりもはるかに小さくなります。たとえば、毎日の時系列が年次的な季節性を持つ場合があります。このモデルには 1 年間の遅延が自動的に含まれるため、コンテキストの長さは 1 年よりも短くなります。モデルが選択する遅延値は、時系列の頻度によって異なります。たとえば、毎日の頻度の遅延値は、前の週、2 週間、3 週間、4 週間、および 1 年です。</p> <p>有効な値</p> <p>正の整数</p> <p>一般的な値</p> <p><math>\text{ceil}(0.1 * \text{ForecastHorizon})</math> to <math>\text{min}(200, 10 * \text{ForecastHorizon})</math></p> <p>デフォルト値</p> <p><math>2 * \text{ForecastHorizon}</math></p>
<b>epochs</b>	<p>トレーニングデータへのパスの最大数。最適な値は、データサイズと学習レートによって異なります。データセットが小さく、学習率が低くなると、いずれもエポック数が多くなり、良好な結果が得られます。</p> <p>有効な値</p> <p>正の整数</p> <p>一般的な値</p> <p>10 ~ 1000</p> <p>デフォルト値</p> <p>500</p>
<b>learning_rate</b>	<p>トレーニングで使用する学習レート。</p>

Parameter Name	説明
	<p>有効な値</p> <p>正の浮動小数点数</p> <p>一般的な値</p> <p>0.0001 ~ 0.1</p> <p>デフォルト値</p> <p>0.001</p>
learning_rate_decay	<p>学習率が低下するレート。学習レートは多くても max_learning_rate_decays 回減少し、その後はトレーニングが停止します。このパラメータが使用されるのは、max_learning_rate_decays が 0 より大きい場合のみです。</p> <p>有効な値</p> <p>正の浮動小数点数</p> <p>一般的な値</p> <p>0.5 ~ 0.8</p> <p>デフォルト値</p> <p>0.5</p>

Parameter Name	説明
likelihood	<p>モデルは確率予測を生成し、分散の変位値を提供してサンプルを返すことができます。データによっては、不確実性予測に使用される適切な尤度 (ノイズモデル) を選択します。</p> <p>有効な値</p> <ul style="list-style-type: none"> <li>• beta: 0 ~ 1 の範囲内 (0 と 1 を含む) の実数値ターゲットに使用します。</li> <li>• deterministic-L1 : 不確実性を予測せず、ポイント予測のみ学習する損失関数。</li> <li>• gaussian: 実数値のデータに使用します。</li> <li>• negative-binomial : データ (負ではない整数) の合計に使用します。</li> <li>• piecewise-linear : 柔軟なディストリビューションに使用します。</li> <li>• student-T : バースト性データの実数値データにこの方法を使用します。</li> </ul> <p>デフォルト値</p> <p>student-T</p>
max_learning_rate_decays	<p>学習レートの減少が起きた場合のその最大回数。</p> <p>有効な値</p> <p>正の整数</p> <p>一般的な値</p> <p>0 ~ 10</p> <p>デフォルト値</p> <p>0</p>

Parameter Name	説明
num_averaged_models	<p>DeepAR+ では、トレーニングの軌跡に複数のモデルが含まれる可能性があります。各モデルの予測の長所と短所は異なる場合があります。DeepAR+ は、すべてのモデルの長所を活用するために、モデルの動作を平均化できます。</p> <p>有効な値</p> <ul style="list-style-type: none"><li>正の整数</li></ul> <p>一般的な値</p> <ul style="list-style-type: none"><li>1~5</li></ul> <p>デフォルト値</p> <ul style="list-style-type: none"><li>1</li></ul>
num_cells	<p>RNN の非表示のレイヤーごとに使用するセルの数。</p> <p>有効な値</p> <ul style="list-style-type: none"><li>正の整数</li></ul> <p>一般的な値</p> <ul style="list-style-type: none"><li>30~100</li></ul> <p>デフォルト値</p> <ul style="list-style-type: none"><li>40</li></ul>

Parameter Name	説明
num_layers	RNN の非表示レイヤーの数。  有効な値  正の整数  一般的な値  1~4  デフォルト値  2

## DeepAR+ モデルを調整する

Amazon Forecast DeepAR+ モデルを調整するには、以下の推奨事項に従って、トレーニングプロセスとハードウェア構成を最適化してください。

### プロセス最適化のためのベストプラクティス

最良の結果を得るためには、以下の推奨事項に従ってください。

- トレーニングとテストのデータセットを分割する場合を除き、トレーニングとテストのため、および推論のためにモデルを呼び出す際には、常に時系列全体を提供します。context\_length の設定方法にかかわらず、時系列を分割したり、その一部だけを指定したりしないでください。モデルは、遅延値の特徴には、context\_length より後ろのデータポイントを使用します。
- モデルを調整するために、データセットをトレーニングデータセットとテストデータセットに分割することができます。典型的な評価シナリオでは、トレーニングで使用したのと同じ時系列でモデルをテストする必要がありますが、トレーニング中に表示される最後の時間ポイントの直後に発生する ForecastHorizon 時間ポイントでモデルをテストします。これらの基準を満たすトレーニングデータセットとテストデータセットを作成するには、データセット全体 (すべての時系列) をテストデータセットとして使用し、トレーニング用の各時系列から最後の ForecastHorizon ポイントを削除します。このようにトレーニング中には、モデルはテスト時に評価される時間ポイントのターゲット値を確認しません。テストフェーズでは、テストデータセット内の各時系列の最後の ForecastHorizon ポイントが保留され、予測が生成されます。その後、予測が最後の ForecastHorizon ポイントの実際の値と比較されます。より複雑な評価を作成するには、テストデータセット内で時系列を複数回繰り返しますが、それぞれ異なる終了ポイントで切り捨てま

す。これにより、さまざまな時間ポイントからの複数の予測で平均化された精度メトリクスが生成されます。

- ForecastHorizon に非常に大きい値 (> 400) を使用することは避けてください。このような値を設定すると、モデルの速度が遅くなり、精度が低下します。未来の予測をさらに生成するには、より高い頻度で情報を集約することを検討してください。たとえば、1min ではなく 5min を使用します。
- 遅延のために、モデルは context\_length より前を確認することができます。そのため、このパラメータを大きい値に設定する必要はありません。このパラメータの開始点として推奨されるのは、ForecastHorizon と同じ値です。
- 可能な限り多くの時系列で DeepAR+ モデルをトレーニングします。単一の時系列でトレーニングされた DeepAR+ モデルも問題なく機能する可能性はありますが、ARIMA や ETS などの標準的な予測方法を使用すると、正確性が増し、そのユースケースへの適合性が高まるように調整される可能性があります。データセットに何百もの特徴の時系列データが含まれている場合、DeepAR+ は標準の方法より優れた性能を発揮するようになります。DeepAR+ では現在、すべてのトレーニング時系列にわたって利用可能な観測の総数が 300 以上であることが要求されます。

## 指数平滑法 (ETS) のアルゴリズム

指数平滑法 ([ETS](#)) は、時系列予測に一般的に使用される局所統計アルゴリズムです。Amazon Forecast の ETS アルゴリズムは、Comprehensive R Archive Network (CRAN) の Package 'forecast' の [ets 関数](#) を呼び出します。

### ETS の仕組み

ETS アルゴリズムは、季節性およびデータに関する他の事前の仮定を含むデータセットに特に役立ちます。ETS では、予測として、入力時系列データセットにおけるすべての観測値の加重平均を計算します。単純移動平均法では、重みは一定ではなく、時間の経過とともに指数関数的に減少します。重みは、定数パラメータ (平滑化パラメータとも呼ばれる) に依存します。

### ETS のハイパーパラメータおよびチューニング

ETS ハイパーパラメータとチューニングの詳細については、[CRAN](#) の「[Package 'forecast'](#)」(パッケージ「予測」) の [ets 関数](#) のドキュメントを参照してください。

Amazon Forecast は、次の表を使用して、[CreateDataset](#) オペレーションで指定された DataFrequency パラメータを R [ts](#) 関数の frequency パラメータに変換します。

DataFrequency (文字列)	R ts frequency (整数)
Y	1
M	12
W	52
D	7
H	24
30 分	2
15 分	4
10 分	6
5 分	12
1 min	60

表にないサポートされているデータ頻度のデフォルト値は、ts 頻度 1 に設定されます。

## ノンパラメトリック時系列 (NPTS) のアルゴリズム

Amazon Forecast ノンパラメトリック時系列 (NPTS) のアルゴリズムは、スケーラブルで確率的なベースライン予測機能です。過去の観測からサンプリングすることにより、特定の時系列の未来の価値分布を予測します。予測は観測値によって制限されます。時系列が断続的 (またはスプース、多くの 0 を含む) かつバースト的であり場合、NPTS は特に便利です。たとえば、時系列に多数の低カウントがある個々のアイテムの需要の予測などです。Amazon Forecast は、過去の観測のどれをサンプリングするかが異なり、かつどのようにサンプリングされるかが異なる、NPTS のバリエーションを提供します。NPTS バリエーションを使用するには、ハイパーパラメータ設定を選択します。

### NPTS の仕組み

指数平滑法 (ETS) や自己回帰和分移動平均 (ARIMA) などの古典的な予測方法と同様に、NPTS は各時系列の予測を個別に生成します。データセットの時系列は長さが異なっていて構いません。観察が利用可能である時間ポイントはトレーニング範囲と呼ばれ、予測が望まれる時間ポイントは予測範囲と呼ばれます。

Amazon Forecast NPTS の予測機能には、NPTS、季節性 NPTS、気候学的予測機能、および季節性気候学的予測機能というバリエーションがあります。

## トピック

- [NPTS](#)
- [季節性 NPTS](#)
- [気候学的予測機能](#)
- [季節性気候学的予測機能](#)
- [季節性の特徴](#)
- [ベストプラクティス](#)

## NPTS

このバリエーションで、予測は、時系列のトレーニング範囲内のすべての観測値からサンプリングすることによって生成されます。ただし、すべての観測値から一様にサンプリングするのではなく、このバリエーションでは、予測が必要な現在の時間ステップからどれほど離れているかに応じて、過去の各観測値に重みを割り当てます。特に、過去の観測値がどれほど離れているかに応じて指数関数的に減衰する重みを使用します。このように、最近の過去からの観測値は、遠い過去からの観測値よりもはるかに高い確率でサンプリングされます。これは、近い過去が遠い過去より未来のためにより指標的であるということを前提としています。重みの減衰量を制御するには、`exp_kernel_weights` ハイパーパラメータを使用します。

Amazon Forecast でこの NPTS バリエーションを使用するには、`use_seasonal_model` ハイパーパラメータを `False` に設定し、他のすべてのデフォルト設定を受け入れます。

## 季節性 NPTS

季節性 NPTS のバリエーションは、NPTS と似ていますが、すべての観測からサンプリングするのではなく、過去の季節の観測のみを使用する点が異なります。デフォルトでは、季節は時系列の詳細度によって決まります。たとえば、1 時間ごとの時系列で時間  $t$  の予測を生成するには、このバリエーションは前日の時間  $t$  に対応する観測値からサンプリングします。NPTS と同様に、前日の  $t$  時の観測値は、それより前の日の  $t$  時の観測値よりも重要度が高くなります。時系列の詳細度に基づいて季節性を判断する方法については、[the section called “季節性の特徴”](#) を参照してください。

## 気候学的予測機能

気候学的予測機能のバリエーションは、過去の観測すべてを一様な確率でサンプリングします。



気候学的予測機能を使用するには、`kernel_type` ハイパーパラメータを `uniform` に、`use_seasonal_model` ハイパーパラメータを `False` に設定します。他のすべてのハイパーパラメータのデフォルト設定を受け入れます。

### 季節性気候学的予測機能

季節性 NPTS と同様に、季節性気候学的予測機能は、過去の季節からの観測値をサンプリングしますが、それらを一様な確率でサンプリングします。

季節性気候学的予測機能を使用するには、`kernel_type` ハイパーパラメータを `uniform` に設定します。他のすべてのハイパーパラメータについては、他のすべてのデフォルト設定を受け入れます。

### 季節性の特徴

季節性 NPTS と季節性気候学的予測機能の季節に対応するものを判断するには、次の表にリストされている特徴を使用します。この表には、詳細度に基づいて、サポートされている基本的な時間頻度で派生する特徴をリストします。Amazon Forecast にはこれらの特徴の時系列が含まれているため、それらを提供する必要はありません。

時系列の頻度	季節性を判断する特徴
分	時間あたりの分
時間	日あたりの時間
日	曜日
週	月あたりの日
月	月

### ベストプラクティス

Amazon Forecast NPTS のアルゴリズムを使用する際には、データを準備して最適な結果を得るために、次のベストプラクティスを考慮してください。

- NPTS は各時系列の予測を個別に生成するため、予測のためにモデルを呼び出すときには時系列全体を指定します。また、`context_length` ハイパーパラメータのデフォルト値を受け入れます。これにより、アルゴリズムで時系列全体が使用されます。

- トレーニングデータが長すぎるために `context_length` を変更する場合は、十分な大きさであり、過去の複数の季節をカバーできることを確認してください。たとえば、毎日の時系列の場合、この値は 365 日以上でなければなりません (ただし、その量のデータがある場合)。

## NPTS のハイパーパラメータ

次の表は、NPTS アルゴリズムで使用できるハイパーパラメータのリストです。

Parameter Name	説明
<code>context_length</code>	<p>モデルが予測を生成するために使用する過去の時間ポイントの数。デフォルトでは、トレーニング範囲内のすべての時間ポイントポイントが使用されます。通常、このハイパーパラメータの値は大きくなければならず、過去の複数の季節をカバーする必要があります。たとえば、毎日の時系列の場合、この値は 365 日以上である必要があります。</p> <p>有効な値</p> <p>正の整数</p> <p>デフォルト値</p> <p>トレーニング時系列の長さ</p>
<code>kernel_type</code>	<p>過去の観測値のサンプリングに使用する重みを定義するためのカーネル。</p> <p>有効な値</p> <p><code>exponential</code>、または <code>uniform</code></p> <p>デフォルト値</p> <p><code>exponential</code></p>
<code>exp_kernel_weights</code>	<p><code>kernel_type</code> が <code>exponential</code> の場合にのみ有効です。</p> <p>カーネルのスケールパラメータ。遠い過去の観測に割り当てられた重みをより速く (指数関数的に) 減衰させるためには、大きい値を使用してください。</p>

Parameter Name	説明
	<p>有効な値</p> <p>正の浮動小数点数</p> <p>デフォルト値</p> <p>0.01</p>
use_seasonal_model	<p>季節性のバリエーションを使用するかどうか。</p> <p>有効な値</p> <p>True、または False</p> <p>デフォルト値</p> <p>True</p>
use_default_time_features	<p>季節性 NPTS および季節性気候学的予測機能のバリエーションにのみ有効です。</p> <p>季節性を判断するために時系列の詳細度に基づいて季節性の特徴を使用するかどうか。</p> <p>有効な値</p> <p>True、または False</p> <p>デフォルト値</p> <p>True</p>

## Prophet アルゴリズム

[Prophet](#) は、人気のある局所的なベイズ構造時系列モデルです。Amazon Forecast の Prophet のアルゴリズムでは、Prophet の Python 実装の [Prophet クラス](#) を使用します。

### Prophet の仕組み

Prophet は、以下のようなデータセットに特に役立ちます。

- 長期 (月次、年次) にわたる詳細な履歴観測 (毎時、日次、週次) を含む
- 複数の強い季節性を持つ
- 以前に認識されていた、重要だが不規則なイベントを含む
- データポイントがないか、大きな外れ値がある
- 限界に近づいている非線形成長トレンドがある

Prophet は、区分線形またはロジスティック成長曲線の傾向を持つ加法回帰モデルです。このモデルには、フーリエ級数を使用してモデル化された季節性 (年) コンポーネントと、ダミー変数を使用してモデル化された季節性 (週) コンポーネントがあります。

詳細については、「[Prophet: スケールに応じた予測](#)」を参照してください。

## Prophet のハイパーパラメータと関連する時系列

Amazon Forecast は、デフォルトの Prophet の [ハイパーパラメータ](#) を使用します。Prophet は、関連する時系列もサポートしています。この機能は、関連する時系列 CSV ファイルで Amazon Forecast に提供されています。

## 予測の生成

Amazon Forecast の予測子を作成したら、予想を作成する準備が整います。デフォルトでは、予測子のトレーニングに使用されたデータセットグループ内の各項目 (item\_id) の予測が含まれます。ただし、予想が生成されるアイテムのサブセットを指定できます。

予想を作成したら、Amazon Simple Storage Service (Amazon S3) バケットにエクスポートできます。

### トピック

- [予想を作成する](#)
- [時系列の指定](#)
- [予想をエクスポートする](#)
- [予想をクエリする](#)
- [コールドスタート予想](#)

## 予想を作成する

Forecast コンソール、AWS CLI、または AWS SDK を使用して予想を作成できます。予想を生成する前に、予測子のステータスがアクティブである必要があります。

### Console

予測を作成するには

1. AWS Management Console にサインインして、Amazon Forecast コンソール (<https://console.aws.amazon.com/forecast/>) を開きます。
2. [Dataset groups] (データセットグループ) から、データセットグループを選択します。
3. データセットグループの [ダッシュボード] の [予測の生成] で、[予想を作成] を選択します。[予想を作成] ページが表示されます。
4. [予想を作成] ページの [予測の詳細] で、予想の名前を入力し、予想の作成に使用する予測子を選択します。
5. 予想分位数について、確率的予想が生成される分位数をオプションで指定します。デフォルトの分位数は、予測子の作成時に指定した分位数です。

- オプションで、[選択項目] のラジオボタンを選択して、予想生成に使用する時系列のサブセットを指定します。
- 必要に応じて、予想に任意のタグを追加します。詳細については、「[Amazon Forecast のリソースのタグ付け](#)」を参照してください。
- [Start] (開始) を選択します。[予想] ページが表示されます。

[ステータス] 列には、予想のステータスが一覧表示されます。Amazon Forecast で、予測の作成が終了するまで待ちます。このプロセスには、数分以上かかることがあります。予想が作成されると、ステータスは [アクティブ] に変わります。

予想が作成されたので、予想をエクスポートできます。「[予想をエクスポートする](#)」を参照してください。

## CLI

create-forecast コマンドを使って、AWS CLI で予想を作成します。予想の名前と予測子の Amazon リソース名 (ARN) を提供します。forecast-types について、確率的予想が生成される分位数をオプションで指定します。デフォルト値は、予測子の作成時に指定した分位数です。必要に応じて、予想に任意のタグを追加します。詳細については、「[Amazon Forecast のリソースのタグ付け](#)」を参照してください。

必須とオプションのパラメータの詳細については、「[CreateForecast](#)」を参照してください。

```
aws forecast create-forecast \
--forecast-name forecast_name \
--forecast-types 0.1 0.5 0.9 \
--predictor-arn arn:aws:forecast:region:account_number:predictor/predictorName \
--tags Key=key1,Value=value1 Key=key2,Value=value2
```

## Python

SDK for Python (Boto3) を使用して予想を作成するには、create\_forecast メソッドを使用します。予想の名前と予測子の Amazon リソース名 (ARN) を提供します。ForecastTypes について、確率的予想が生成される分位数をオプションで指定します。デフォルト値は、予測子の作成時に指定した分位数です。必要に応じて、予想に任意のタグを追加します。詳細については、「[Amazon Forecast のリソースのタグ付け](#)」を参照してください。

必須とオプションのパラメータの詳細については、「[CreateForecast](#)」を参照してください。

```
import boto3
```

```
forecast = boto3.client('forecast')

create_forecast_response = forecast.create_forecast(
    ForecastName = "Forecast_Name",
    ForecastTypes = ["0.1", "0.5", "0.9"],          # optional, the default types/
    quantiles are what you specified for the predictor
    PredictorArn = "arn:aws:forecast:region:accountNumber:predictor/predictorName",
    Tags = [
        {
            "Key": "key1",
            "Value": "value1"
        },
        {
            "Key": "key2",
            "Value": "value2"
        }
    ]
)
forecast_arn = create_forecast_response['ForecastArn']
print(forecast_arn)
```

## 時系列の指定

### Note

時系列は、項目 (item\_id) とデータセット内のすべての次元の組み合わせです

時系列のリストを指定するには、item\_id と次元値によって時系列を識別する CSV ファイルを S3 バケットにアップロードします。また、スキーマで時系列の属性と属性タイプを定義する必要があります。

例えば、小売業者は、特定の店舗 (store\_location) での特定の商品 (item\_id) の売上について、広告キャンペーンがどのように影響するかを知りたい場合があります。このユースケースでは、item\_id と store\_location の組み合わせである時系列を指定します。

次の CSV ファイルは、次の 5 つの時系列を選択します。

1. Item\_id: 001、store\_location: シアトル

2. Item\_id: 001、store\_location: ニューヨーク
3. Item\_id: 002、store\_location: シアトル
4. Item\_id: 002、store\_location: ニューヨーク
5. Item\_id: 003、store\_location: デンバー

```
001, Seattle
001, New York
002, Seattle
002, New York
003, Denver
```

スキーマは、最初の列を `item_id` として定義し、2 番目の列を `store_location` として定義します。

入力データセットに含まれていない指定の時系列では、予想の作成はスキップされます。予想エクスポートファイルには、これらの時系列や予想値は含まれません。

## 予想をエクスポートする

予想を作成したら、それを Amazon S3 バケットにエクスポートできます。予想をエクスポートすると、予想が CSV ファイル (デフォルト) として Amazon S3 バケットにコピーされ、エクスポートされたデータには、項目予測に加えて項目メタデータデータセットのすべての属性が含まれます。予想をエクスポートするときに Parquet ファイル形式を指定できます。

エクスポートされる予想の精度 (時間単位、日単位、週単位など) は、予測子の作成時に指定した予想頻度です。必要に応じて、バケットに書き込まれる前にデータを暗号化する AWS Key Management Service キーを指定できます。

### Note

エクスポートファイルは、[Dataset Import] (データセットのインポート) から直接情報を返すことができます。これにより、インポートされたデータに式またはコマンドが含まれている場合、ファイルは CSV インジェクションに対して脆弱になります。このため、エクスポートされたファイルはセキュリティ警告を表示する可能性があります。悪意のあるアクティビティを回避するには、エクスポートされたファイルを読み取るときにリンクとマクロを無効にします。



## Console

予想をエクスポートするには

1. ナビゲーションペインで、データセットグループの [予測] を選択します。
2. 予想のラジオボタンを選択し、[予想エクスポートを作成] を選択します。[Create forecast export (予測エクスポートの作成)] ページが表示されます。
3. [Create forecast export (予測エクスポートの作成)] ページの [エクスポートの詳細] に次の情報を入力します。
  - [Export name] (エクスポート名) - 予測エクスポートジョブの名前を入力します。
  - [Generated forecast] (生成された予測) - ドロップダウンメニューから、Step 3: Create a Forecast で作成した予測を選択します。
  - IAM ロール — デフォルトの [カスタム IAM ロール ARN を入力] のままにするか、[新しいロールを作成] を選択すると Amazon Forecast がロールを自動的に作成します。
  - カスタム IAM ロール ARN - カスタム IAM ロールを入力する場合、[Amazon Forecast の IAM ロールを作成する \(IAM コンソール\)](#) で作成した IAM ロールの Amazon リソースネーム (ARN) を入力します。
  - KMS キー ARN – バケット暗号化のために AWS Key Management Service を使用する場合、AWS KMS キーの Amazon リソースネーム (ARN) を指定します。
  - [S3 予測エクスポートの場所] - 次の形式を使用して、Amazon Simple Storage Service (Amazon S3) バケットまたはバケットのフォルダの場所を入力します。

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

4. [Create forecast export (予測エクスポートの作成)] を選択します。[my\_forecast] ページが表示されます。

Amazon Forecast で、予測のエクスポートが終了するまで待ちます。このプロセスには、数分以上かかることがあります。予想がエクスポートされたら、ステータスが [アクティブ] に変わり、Amazon S3 バケットで予想ファイルを見つけることができます。

## CLI

export-forecast-job コマンドを使って、AWS CLI で予想をエクスポートします。予想エクスポートジョブに名前を付け、エクスポートする予想の ARN を指定し、必要に応じてタグを追加します。destination には、出力 Amazon S3 バケットへのパス、[Amazon Forecast の IAM](#)

[ロールを作成する \(IAM コンソール\)](#) で作成した IAM ロールの ARN を指定し、バケット暗号化に AWS KMS キーを使用する場合はキーの ARN を指定します。

必須のパラメータとオプションのパラメータの詳細については、「[CreateForecastExportJob operation](#)」を参照してください。

```
forecast create-forecast-export-job \  
--forecast-export-job-name exportJobName \  
--forecast-arn arn:aws:forecast:region:acctNumber:forecast/forecastName \  
--destination  
  S3Config="{Path='s3://bucket/folderName',RoleArn='arn:aws:iam::acctNumber:role/Role, KMSKeyArn='arn:aws:kms:region:accountNumber:key/keyID'}"  
--tags Key=key1,Value=value1 Key=key2,Value=value2
```

## Python

SDK for Python (Boto3) を使用して予想をエクスポートするには、`export_forecast_job` メソッドを使用します。予想エクスポートジョブに名前を付け、エクスポートする予想の ARN を指定し、必要に応じてタグを追加します。Destination には、出力 Amazon S3 バケットへのパス、[Amazon Forecast の IAM ロールを作成する \(IAM コンソール\)](#) で作成した IAM ロールの ARN を指定し、バケット暗号化に AWS KMS キーを使用する場合はキーの ARN を指定します。

必須のパラメータとオプションのパラメータの詳細については、「[CreateForecastExportJob operation](#)」を参照してください。

```
import boto3  
  
forecast = boto3.client('forecast')  
  
export_forecast_response = forecast.create_forecast_export_job(  
    Destination = {  
        "S3Config": {  
            "Path": "s3://bucketName/folderName/",  
            "RoleArn": "arn:aws:iam::accountNumber:role/roleName",  
            "KMSKeyArn": "arn:aws:kms:region:accountNumber:key/keyID"  
        }  
    },  
    ForecastArn = "arn:aws:forecast:region:accountNumber:forecast/forecastName",  
    ForecastExportJobName = "export_job_name",  
    Tags = [  
        {  
            "Key": "key1",
```

```
        "Value": "value1"
    },
    {
        "Key": "key2",
        "Value": "value2"
    }
]
)
forecast_export_job_arn = export_forecast_response["ForecastExportJobArn"]
print(forecast_export_job_arn)
```

## 予想をクエリする

[QueryForecast](#) オペレーションを使用して予想をクエリできます。デフォルトでは、予測の全範囲が返ります。完全な予測内で特定の日付範囲をリクエストできます。

予測をクエリする際、フィルタリング条件を指定する必要があります。フィルタはキーと値のペアです。キーは、予測の作成に使用されるいずれかのデータセットのスキーマ属性名 (予測ディメンションを含む) の 1 つです。値は、指定されたキーの有効な値です。複数のキーと値のペアを指定できます。返る予測には、すべての条件を満たすアイテムのみが含まれます。

## コールドスタート予想

小売、製造、消費財などの業界のユーザーが直面する一般的な課題は、履歴データがない項目の予想を作成することです。このシナリオはコールドスタート予想と呼ばれ、通常、企業が新製品を市場に投入したり、ブランドやカタログを組み込んだり、新しい地域で製品をクロスセルしたりするときに発生します。

Amazon Forecast では、コールドスタート予想を実行するために項目のメタデータが必要です。Forecast は、項目メタデータにある項目特性を利用して、項目メタデータ内で、履歴データがない項目に類似する項目を明示的に識別します。Forecast は、既存の項目の需要特性を使用して、新しい項目のコールドスタート予想を生成します。

Amazon Forecast では、コールドスタート項目は、項目メタデータファイルには含まれているが、ターゲットの時系列ファイルには含まれていない項目として識別されます。コールドスタート項目を正しく識別するには、コールドスタート項目の項目 ID が項目メタデータファイルの行として入力され、ターゲットの時系列ファイルには入力されていないことを確認してください。コールドスタート項目が複数ある場合は、各項目 ID を項目メタデータファイルの個別の行として入力します。コールド

ドスタート項目に項目 ID がない場合は、64 文字未満で、データセット内の別の項目でまだ使用されていない任意の英数字の組み合わせを使用できます。

コールドスタート予想には、項目メタデータデータセットと AutoPredictor の両方が必要です。

# Forecast Explainability

Forecast Explainability は、データセット内の属性が特定の時系列 (項目とディメンションの組み合わせ) および時間ポイントの予測にどのように影響するかをよりよく理解するのに役立ちます。Forecast は、Impact スコアと呼ばれるメトリクスを使用して、各属性の相対的な影響を定量化し、それらが予測値を増加させるか減少させるかを決定します。

例えば、ターゲットが sales であり、price および color といった関連する 2 つの属性がある予測シナリオについて考えてみます。Forecast では、商品の色が、特定の商品の売上には大きな影響を与えるが、他の商品に対する影響はごくわずかであると判断する可能性があります。また、夏のプロモーションは売上に大きな影響を与えるが、冬のプロモーションはほとんど効果がないと判断する可能性もあります。

Forecast Explainability を有効にするには、関連する時系列、項目メタデータ、または祝祭日や Weather Index などの追加のデータセットの少なくとも 1 つを予測子に含める必要があります。詳細については、「[制約事項とベストプラクティス](#)」を参照してください。

データセット内のすべての時系列と時間ポイントについての集約された Impact スコアを表示するには、Forecast Explainability の代わりに Predictor Explainability を使用します。「[Predictor Explainability](#)」(予測子の Explainability) を参照してください。

## Python ノートブック

Forecast Explainability の step-by-step ガイドについては、「[アイテムレベルの Explainability](#)」を参照してください。

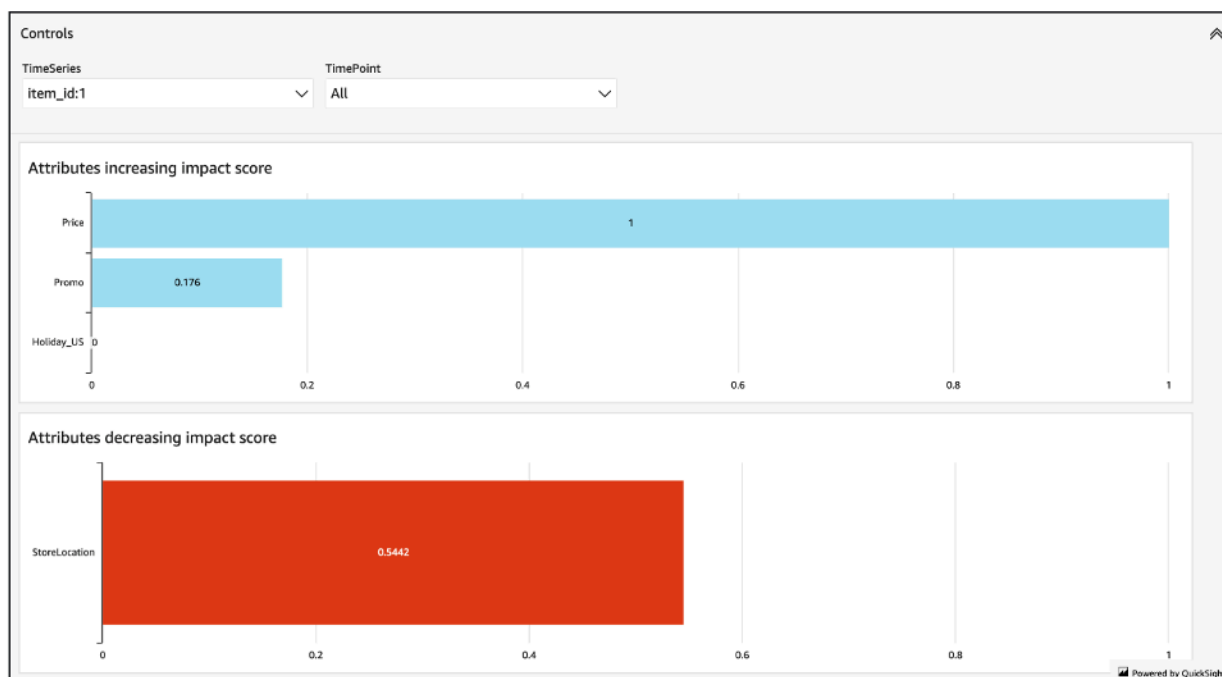
## トピック

- [影響スコアの解釈](#)
- [Forecast Explainability の作成](#)
- [Forecast Explainability の視覚化](#)
- [Forecast Explainability のエクスポート](#)
- [制約事項とベストプラクティス](#)

## 影響スコアの解釈

影響スコアは、予測値に対する相対的な影響属性を測定します。例えば、「price」属性の影響スコアが「store location」属性の2倍である場合、商品の料金は、予測値に対して、店舗の場所の2倍の影響を及ぼすと結論付けることができます。

影響スコアは、属性が予測値を増加させるか減少させるかに関する情報も提供します。コンソールでは、これは2つのグラフで示されます。青いバーのある属性は予測値を増やし、赤いバーのある属性は予測値を減らします。



Impact スコアは、絶対的な影響ではなく、属性の相対的な影響を測定することに注意することが重要です。したがって、Impact スコアで、特定の属性がモデルの精度が向上するかどうかを判断することはできません。属性の Impact スコアが低い場合、それは必ずしも予測値への影響が少ないことを意味するわけではありません。これは、予測子が使用する他の属性よりも予測値への影響が少ないことを意味します。

すべてまたは一部の影響スコアがゼロになる可能性があります。これは、特徴量が予測値に影響を与えない場合、ML 以外のアルゴリズムのみ AutoPredictor を使用した場合、または関連する時系列やアイテムメタデータを提供しなかった場合に発生する可能性があります。

[Forecast Explainability] については、Impact スコアには、Normalized impact スコアと Raw impact スコアの2つの形式があります。生の影響スコアはシャープレイ値に基づいており、スケーリング

または制限されていません。正規化された影響スコアは、生のスコアを -1 から 1 の間の値にスケールします。

生の影響スコアは、さまざまな Explainability リソース間でスコアを組み合わせて比較するのに役立ちます。例えば、予測子に 50 を超える時系列または 500 を超える時間ポイントが含まれている場合、複数の Forecast Explainability リソースを作成して、より多くの時系列または時間ポイントを合わせた数をカバーし、属性の生の影響スコアを直接比較できます。ただし、さまざまな予測からの Forecast Explainability リソースの生の影響スコアを直接比較することはできません。

コンソールで Impact スコアを表示すると、Normalized impact スコアのみが表示されます。Explainability をエクスポートすると、生のスコアと正規化されたスコアの両方が提供されます。

## Forecast Explainability の作成

Forecast Explainability を使用すると、属性が特定の時間ポイントでの特定の時系列の予測値にどのように影響するかを詳しく調べることができます。時系列と時間ポイントを指定した後、Amazon Forecast は、それらの特定の時系列と時間ポイントのみの Impact スコアを計算します。

ソフトウェア開発キット (SDK) または Amazon Forecast コンソールを使用して、予測子の Forecast Explainability を有効にできます。SDK を使用する場合は、[CreateExplainability](#) オペレーションを使用します。

トピック

- [時系列の指定](#)
- [時間ポイントの指定](#)

### 時系列の指定

#### Note

時系列は、項目 (item\_id) とデータセット内のすべてのディメンションの組み合わせです

Forecast Explainability に時系列 (項目とディメンションの組み合わせ) を指定すると、Amazon Forecast は、それらの特定の時系列のみについて、属性の Impact スコアを計算します。

時系列のリストを指定するには、`item_id` とディメンション値によって時系列を識別する CSV ファイルを S3 バケットにアップロードします。最大 50 個の時系列を指定できます。また、スキーマで時系列の属性と属性タイプを定義する必要があります。

例えば、小売業者は、特定の店舗の場所 (`store_location`) での特定の商品 (`item_id`) の売上について、プロモーションがどのように影響するかを知りたい場合があります。このユースケースでは、`item_id` と `store_location` の組み合わせである時系列を指定します。

次の CSV ファイルは、次の 5 つの時系列を選択します。

1. `Item_id: 001`、`store_location: シアトル`
2. `Item_id: 001`、`store_location: ニューヨーク`
3. `Item_id: 002`、`store_location: シアトル`
4. `Item_id: 002`、`store_location: ニューヨーク`
5. `Item_id: 003`、`store_location: デンバー`

```
001, Seattle
001, New York
002, Seattle
002, New York
003, Denver
```

スキーマは、最初の列を `item_id` として定義し、2 番目の列を `store_location` として定義します。

Forecast コンソールまたは Forecast ソフトウェア開発キット (SDK) を使用して、時系列を指定できます。

## Console

Forecast Explainability の時系列を指定するには

1. にサインイン AWS Management Console し、<https://console.aws.amazon.com/forecast/> で Amazon Forecast コンソールを開きます。
2. [Dataset groups] (データセットグループ) から、データセットグループを選択します。
3. ナビゲーションペインで、[Insights] を選択します。
4. [Create Explainability] (Explainability を作成) を選択します。



5. [Explainability name] (Explainability 名) のフィールドに、Forecast Explainability の一意の名前を入力します。
6. [Select forecast] (予測を選択) のフィールドで、予測を選択します。
7. [S3 のロケーション] フィールドに、時系列を含むファイルの場所を入力します。
8. [Data schema] (データスキーマ) のフィールドで、[attribute name] (属性名) を、時系列で 사용되는項目 ID とディメンションの [attribute type] (属性タイプ) に設定します。
9. [Create Explainability] (Explainability を作成) を選択します。

## SDK

Forecast Explainability の時系列を指定するには

[CreateExplainability](#) オペレーションを使用して、一意の名前を指定 ExplainabilityName し、の予測 ARN を指定します ResourceArn。

次のデータ型を設定します。

- ExplainabilityConfig - の値を TimeSeriesGranularity 「SPECIFIC」に設定し、「ALL TimePointGranularity」に設定します。(時間ポイントを指定するには、「SPECIFIC TimePointGranularity」に設定します。[「時間ポイントの指定」](#)を参照)
- S3Config - 「Path」の値を時系列ファイルの S3 の場所に設定し、RoleArn 「」の値を S3 バケットにアクセスできるロールに設定します。
- Schema - item\_id のAttributeName 「」と AttributeType 「」、および時系列のディメンションを定義します。

次の例は、「item\_id」と「store\_location」のディメンションの組み合わせを使用した時系列のスキーマを示しています。

```
{
  "ExplainabilityName" : [unique_name],
  "ResourceArn" : [forecast_arn],
  "ExplainabilityConfig" {
    "TimeSeriesGranularity": "SPECIFIC",
    "TimePointGranularity": "ALL"
  },
  "DataSource": {
    "S3Config": {
      "Path": [S3_path_to_file],
```

```
        "RoleArn": [role-to-access-s3-bucket]
      }
    },
    "Schema": {
      "Attributes": [
        {
          "AttributeName": "item_id",
          "AttributeType": "string"
        },
        {
          "AttributeName": "store_location",
          "AttributeType": "string"
        }
      ]
    }
  },
}
```

## 時間ポイントの指定

### Note

時間ポイント ("TimePointGranularity": "ALL") を指定しない場合、Amazon Forecast は、Impact スコアを計算するときに予測期間全体を考慮します。

Forecast Explainability の時間ポイントを指定すると、Amazon Forecast は、その特定の時間範囲について、属性の Impact スコアを計算します。予測期間内で最大 500 の連続する時間ポイントを指定できます。

例えば、小売業者は、その属性が冬季の売上にどのように影響するかを知りたい場合があります。このユースケースでは、予測期間中の冬季のみにまたがる時間ポイントを指定します。

Forecast コンソールまたは Forecast ソフトウェア開発キット (SDK) を使用して、時間ポイントを指定できます。

### Console

Forecast Explainability の時系列を指定するには

1. にサインイン AWS Management Console し、<https://console.aws.amazon.com/forecast/> で Amazon Forecast コンソールを開きます。

2. [Dataset groups] (データセットグループ) から、データセットグループを選択します。
3. ナビゲーションペインで、[Insights] を選択します。
4. [Create Explainability] (Explainability を作成) を選択します。
5. [Explainability name] (Explainability 名) のフィールドに、Forecast Explainability の一意の名前を入力します。
6. [Select forecast] (予測を選択) のフィールドで、予測を選択します。
7. [S3 のロケーション] フィールドに、時系列を含むファイルの場所を入力します。
8. [Data schema] (データスキーマ) のフィールドで、[attribute name] (属性名) を、時系列で使用する項目 ID とディメンションの [attribute type] (属性タイプ) に設定します。
9. [Time duration] (期間) のフィールドで、カレンダー内の開始日と終了日を指定します。
10. [Create Explainability] (Explainability を作成) を選択します。

## SDK

Forecast Explainability の時系列を指定するには

[CreateExplainability](#) オペレーションを使用して、一意の名前を指定 ExplainabilityName し、の予測 ARN を指定します ResourceArn。次のタイムスタンプ形式を使用して、開始日 (StartDateTime) と終了日 (EndDateTime) を設定します: yyyy-MM-ddTHH:mm:ss (例: 2015-01-01T20:00:00)。

次のデータ型を設定します。

- ExplainabilityConfig - の値を TimeSeriesGranularity 「SPECIFIC」と 「SPECIFIC TimePointGranularity」 に設定します。
- S3Config - 「Path」 の値を時系列ファイルの S3 の場所に設定し、RoleArn 「」 の値を S3 バケットにアクセスできるロールに設定します。
- Schema - item\_id のAttributeName 「」 と AttributeType 「」、および時系列のディメンションを定義します。

次の例は、「item\_id」と「store\_location」のディメンションの組み合わせを使用した時系列のスキーマを示しています。

```
{
  "ExplainabilityName" : [unique_name],
  "ResourceArn" : [forecast_arn],
```

```
"ExplainabilityConfig" {
  "TimeSeriesGranularity": "SPECIFIC",
  "TimePointGranularity": "SPECIFIC"
},
"DataSource": {
  "S3Config": {
    "Path": [S3_path_to_file],
    "RoleArn": [role-to-access-s3-bucket]
  }
},
"Schema": {
  "Attributes": [
    {
      "AttributeName": "item_id",
      "AttributeType": "string"
    },
    {
      "AttributeName": "store_location",
      "AttributeType": "string"
    }
  ]
},
"StartDateTime": "string",
"EndDateTime": "string",
}
```

## Forecast Explainability の視覚化

コンソールで Forecast Explainability を作成すると、Forecast は Impact スコアを自動的に視覚化します。[CreateExplainability](#) オペレーションで Forecast Explainability を作成する場合、`EnableVisualization` を「true」に設定すると、その Explainability リソースの影響スコアがコンソール内で視覚化されます。

影響スコアのビジュアライゼーションは、Explainability の作成日から 30 日間継続します。ビジュアライゼーションを再作成するには、Forecast の新しい Explainability を作成します。

# Forecast Explainability のエクスポート

## Note

エクスポートファイルは、[Dataset Import] (データセットのインポート) から直接情報を返すことができます。これにより、インポートされたデータに式またはコマンドが含まれている場合、ファイルは CSV インジェクションに対して脆弱になります。このため、エクスポートされたファイルはセキュリティ警告を表示する可能性があります。悪意のあるアクティビティを回避するには、エクスポートされたファイルを読み取るときにリンクとマクロを無効にします。

Forecast を使用すると、Impact スコアの CSV ファイルを S3 の場所にエクスポートできます。

エクスポートには、指定された時系列の生の影響スコアと正規化された影響スコア、および指定されたすべての時系列と指定されたすべての時間ポイントの正規化および集約された影響スコアが含まれます。時間ポイントを指定しなかった場合、影響スコアは、予測期間内のすべての時間ポイントで既に集計されています。

Item_id	timestamp	Price-RawImpactScore	Promo-RawImpactScore	StoreLocation-RawImpactScore	Holiday_US-RawImpactScore	Price-NormalizedImpactScore	Promo-NormalizedImpactScore	StoreLocation-NormalizedImpactScore	Holiday_US-NormalizedImpactScore
Aggregate	Aggregate	NaN	NaN	NaN	NaN	-0.4967	0.6072	-0.2302	0
1	Aggregate	-0.0296	0.0572	-0.0797	0	1	0.176	-0.5442	0
2	Aggregate	57.5804	150.0358	4.0403	0	0.3838	1	-0.0263	0
3	Aggregate	-0.0751	0.025	-0.002	0	0.7174	0.3335	-1	0
1	2015-01-26T00:00:00Z	-6.8968	-12.9865	-0.2756	0	-0.1178	-0.2219	-0.0047	0
1	2016-05-09T00:00:00Z	-1.9732	-11.4329	-14.744	0	-0.0337	-0.1953	-0.2519	0
1	2015-03-09T00:00:00Z	-2.8406	-13.0931	-1.9269	0	-0.0485	-0.2237	-0.0329	0
1	2015-06-22T00:00:00Z	-2.3571	-8.5324	-14.4815	0	-0.0403	-0.1458	-0.2474	0
1	2016-08-29T00:00:00Z	-3.1274	-5.0817	-19.1643	0	-0.0534	-0.0868	-0.3274	0
1	2016-12-12T00:00:00Z	-11.4177	-12.4537	-4.4453	0	-0.1951	-0.2128	-0.076	0
1	2017-05-01T00:00:00Z	-2.4247	-7.2943	-18.8764	0	-0.0414	-0.1246	-0.3225	0
1	2017-07-03T00:00:00Z	-1.4287	-13.7995	-6.1356	0	-0.0244	-0.2358	-0.1048	0
1	2016-04-11T00:00:00Z	-6.00E-04	-0.0012	-0.0043	0	-0.0368	-0.0776	-0.2842	0
2	2015-03-16T00:00:00Z	-2.4852	-7.6812	-7.4003	0	-0.0425	-0.1312	-0.1264	0
2	2016-05-09T00:00:00Z	-0.0037	-1.00E-04	-5.00E-04	0	-0.2692	-0.0038	-0.0394	0
2	2015-05-04T00:00:00Z	10.4991	-10.0409	-2.1008	0	0.1794	-0.1716	-0.0359	0
2	2014-11-03T00:00:00Z	-6.00E-04	-0.0055	-0.0022	0	-0.0416	-0.3628	-0.1457	0
2	2015-02-09T00:00:00Z	-0.0024	-0.0011	-0.0025	0	-0.1599	-0.0726	-0.1632	0
2	2014-08-25T00:00:00Z	-3.7739	-2.4773	-2.4888	0	-0.0645	-0.0423	-0.0425	0
2	2017-12-18T00:00:00Z	-6.00E-04	-0.0047	-0.0041	0	-0.0424	-0.3063	-0.2683	0
2	2015-05-11T00:00:00Z	-3.00E-04	-2.00E-04	-3.00E-04	0	-0.0227	-0.0146	-0.0231	0
2	2014-12-08T00:00:00Z	-5.4927	-8.1132	-0.0168	0	-0.0939	-0.1386	-3.00E-04	0
2	2015-04-06T00:00:00Z	-0.0018	-0.0016	-3.00E-04	0	-0.1172	-0.1022	-0.0197	0
3	2014-10-20T00:00:00Z	-0.0025	-0.0027	-0.0034	0	-0.1639	-0.1804	-0.2227	0
3	2015-04-13T00:00:00Z	-21.7456	3.1561	-16.2541	0	-0.3716	0.0539	-0.2777	0
3	2018-04-23T00:00:00Z	-1.2579	-0.2137	-5.6459	0	-0.0215	-0.0037	-0.0965	0
3	2015-01-19T00:00:00Z	-0.0031	-9.00E-04	-0.0045	0	-0.2304	-0.0691	-0.3313	0
3	2017-01-30T00:00:00Z	-0.0036	-0.0034	-0.0023	0	-0.2414	-0.2229	-0.1542	0
3	2017-05-08T00:00:00Z	16.5512	-2.5499	-15.8288	0	0.2828	-0.0436	-0.2705	0
3	2016-05-30T00:00:00Z	-0.0015	-0.0027	-4.00E-04	0	-0.1078	-0.2004	-0.0276	0
3	2018-05-28T00:00:00Z	-15.528	-15.369	-0.4334	0	-0.2653	-0.2626	-0.0074	0
3	2017-06-19T00:00:00Z	-16.0061	-9.0946	10.3333	0	-0.2735	-0.1554	0.1766	0
3	2017-12-25T00:00:00Z	-8.5566	-1.8031	-2.7768	0	-0.1462	-0.0308	-0.0474	0

Amazon Forecast のソフトウェア開発キット (SDK) および Amazon Forecast コンソールを使用して、Forecast Explainability をエクスポートできます。

## Console

Forecast Explainability をエクスポートするには

1. にサインイン AWS Management Console し、<https://console.aws.amazon.com/forecast/> で Amazon Forecast コンソールを開きます。
2. [Dataset groups] (データセットグループ) から、データセットグループを選択します。
3. ナビゲーションペインで、[Insights] を選択します。
4. Explainability を選択します。
5. [Actions] (アクション) のドロップダウンから、[Export] (エクスポート) を選択します。
6. [Export name] (エクスポート名) のフィールドに、Forecast Explainability エクスポートの一意の名前を入力します。
7. [S3 explainability export location] (S3 の Explainability のエクスポートの場所) のフィールドに、CSV ファイルをエクスポートするための S3 の場所を入力します。
8. [IAM Role] (IAM ロール) のフィールドで、選択した S3 の場所にアクセスできるロールを選択します。
9. [Create Explainability Export] (Explainability エクスポートを作成) を選択します。

## SDK

Forecast Explainability をエクスポートするには

[CreateExplainabilityエクスポート](#) オペレーションを使用して、Destination オブジェクトで S3 の場所と IAM ロールを、ExplainabilityArn および `ExplainabilityExportName` とともに指定します。

例:

```
{
  "Destination": {
    "S3Config": {
      "Path": "s3://bucket/example-path/",
      "RoleArn": "arn:aws:iam::000000000000:role/ExampleRole"
    }
  },
  "ExplainabilityArn": "arn:aws:forecast:region:explainability/example",
  "ExplainabilityName": "Explainability-export-name",
}
```

## 制約事項とベストプラクティス

Forecast Explainability を使用する場合は、次の制約事項とベストプラクティスを考慮してください。

- Forecast Explainability は、 から生成された一部の Forecasts でのみ使用できます AutoPredictor 。レガシー予測子 (AutoML または手動選択) から生成された Forecast Explainability を有効にすることはできません。 [「へのアップグレード AutoPredictor」](#) を参照してください。
- Forecast Explainability はすべてのモデルで利用できるわけではありません。ARIMA (AutoRegressive 統合移動平均)、ETS (指数平滑状態空間モデル)、および NPTS (非パラメトリック時系列) モデルには、外部時系列データが組み込まれていません。したがって、これらのモデルでは、追加のデータセットを含めても、説明可能性レポートは作成されません。
- Explainability には属性が必要 - 予測子には、関連する時系列、項目メタデータ、Holidays、または Weather Index の少なくとも 1 つを含める必要があります。
- 影響スコアがゼロの場合は影響がないことが示唆されている - 1 つ以上の属性の影響スコアが 0 の場合、これらの属性は予想値に大きな影響を与えません。が ML 以外のアルゴリズムのみ AutoPredictor を使用した場合、または関連する時系列や項目メタデータを指定しなかった場合も、スコアは 0 になることがあります。
- 最大 50 の時系列を指定 - Forecast Explainability ごとに最大 50 の時系列を指定できます。
- 最大 500 の時間ポイントを指定 - Forecast Explainability ごとに最大 500 の連続した時間ポイントを指定できます。
- Forecast がいくつかの集約された Impact スコアも計算 - Forecast は、指定された時系列および時間ポイントの集約された影響スコアも提供します。
- 単一の予測に対して複数の Forecast Explainability リソースを作成する - 50 を超える時系列または 500 の時間ポイントの影響スコアが必要な場合は、より広範囲にわたる Explainability リソースをバッチで作成できます。
- さまざまな Forecast Explainability リソース間での Raw impact スコアを比較する - Raw Impact スコアは、同じ予測の Explainability リソース間で直接比較できます。
- Forecast Explainability のビジュアライゼーションが作成後 30 日間利用可能 - 30 日後にビジュアライゼーションを表示するには、同じ設定で新しい Forecast Explainability のビジュアライゼーションを作成します。

# What-If 分析

What-If 分析は、さまざまなシナリオが Amazon Forecast によって作成されたベースライン予想にどのように影響するかを調査して説明するのに役立つツールです。ベースライン予想は、ユーザーが提供した元の関連時系列に基づいて Amazon Forecast によって作成された予想です。

What-If 分析では、関連する時系列をどのように変更したかに基づいて、一連の What-If 予想が作成されます。これらの What-If 予想をベースライン予想と比較対照することで、特定の変更がモデルにどのような影響を与えるかを理解しやすくなります。

修正された関連時系列を作成する方法は 2 つあります。Amazon S3 パスに変更された関連時系列を指定することも、既存の関連時系列に変換セットを指定することもできます。変換のセットを指定すると、それらの変更を含む元の関連時系列のコピーが作成されます。

変換により、関連する時系列のサブセットを作成したり、関連する時系列の特定の属性を変更したりできます。詳細については、[the section called “代替データセット”](#) および [the section called “変換関数”](#) を参照してください。

## トピック

- [What-If 分析の作成](#)
- [変換関数](#)
- [代替データセット](#)

## What-If 分析の作成

What-If 分析では、ベースラインに関連する時系列の変化が予測にどのように影響するかを調査します。What-If 分析は、AutoPredictor を使用する予想からのみ作成できます。What-If 分析を作成したら、1 つ以上の What-If 予想を作成します。What-If 予想とベースライン予想を比較し、What-If 予想を 1 つ以上エクスポートします。

### Note

What-If 分析を作成するには、データをカンマ区切り値 (CSV) 形式にする必要があります。

## トピック

- [What-If 分析の作成](#)



- [What-If 予想の作成](#)
- [What-If 予想を比較する](#)
- [What-If 予想をエクスポートする](#)
- [What-If 予想をクエリする](#)

## What-If 分析の作成

Forecast コンソールまたは Forecast Software Development Kit (SDK) を使用して、What-If 分析を作成できます。

### Console

What-If 分析を作成するには、次のステップを実行します。

1. AutoPredictor を使用して学習させた予想を作成します。
2. 関心のある予想を含むデータセットグループダッシュボードを開きます。
3. [What-If 分析を確認] を選択します。
4. [インサイト] ページの [What-If 分析] タブで、[作成] を選択します。
5. [What-If 分析名] フィールドに一意の名前を入力し、この分析のベースライン予測を選択します。
6. [項目の選択] エリアで、分析にすべての項目を自動的に含めるか、ファイルに含める項目を指定するかを選択します。

[ファイルから項目を選択] を選択した場合は、What-If 予想で変更する項目だけを含むデータセットを指定する必要があります。詳細については、「[時系列の指定](#)」を参照してください。

7. [What-If 分析を作成] を選択します。What-If 分析ページの上部にあるバナーに、What-If 分析作成ジョブのステータスが表示されます。

### SDK

[CreateWhatIfAnalysis](#) オペレーションを使用して、WhatIfAnalysisName に一意の名前を指定し、ForecastArn のベースライン予想の予想 ARN を指定します。次の例は、「item\_id」と「store\_location」の次元の組み合わせを使用した時系列のスキーマを示しています。詳細については、「[時系列の指定](#)」を参照してください。

```
{
```

```
"ForecastArn": "arn:aws:forecast:region:acctNumber:forecast/baselineForecast",
"WhatIfAnalysisName": "unique_name",
"TimeSeriesSelector": {
  "TimeSeriesIdentifiers": {
    "DataSource": {
      "S3Config": {
        "Path": "s3://bucket/example-path",
        "RoleArn": "arn:aws:iam::000000000000:role/ExampleRole"
      }
    },
    "Schema": {
      "Attributes": [
        {
          "AttributeName": "item_id",
          "AttributeType": "string"
        },
        {
          "AttributeName": "store_location",
          "AttributeType": "string"
        }
      ]
    }
  }
}
```

## What-If 予想の作成

Forecast コンソールまたは Forecast Software Development Kit (SDK) を使用して、What-If 予想を作成できます。

### Console

What-If 予想を作成するには、次のステップを実行します。

1. [インサイト] ページの [What-If 分析] タブで、関心がある What-If 分析を選択します。
2. [What-If 予想] セクションで、[作成] を選択します。
3. [What-If 予想の作成] ページで、固有の What-If 予想名を入力し、[変換関数を使用する] または [What-If 予想を代替データセットで定義する] を選択します。詳細については、[the section called “代替データセット”](#) および [the section called “変換関数”](#) を参照してください。

- a. [変換関数を使用する] を選択した場合は、[変換関数ビルダー] を使用して What-If 予想に含まれる行を選択して変更する必要があります。すべての変換は、指定された順序で適用されます。条件は指定された順序で適用され、AND 演算で結合されます。変換は、すべての条件が満たされた場合にのみ適用されます。
  - b. [代替データセットで What-If 予想を定義] を選択した場合は、What-If 予想で変更したい行のみを含む代替データセットを指定する必要があります。
4. [Create] (作成) を選択します。

## SDK - Transformation Function

[CreateWhatIfForecast](#) オペレーションを使用して、WhatIfAnalysisName に一意の名前を指定し、ForecastArn の予想 ARN を指定します。以下の例は、「store\_location」が「tacoma」でない場合に「price」に変換するスキーマを示しています。

```
{
  "WhatIfAnalysisArn": "arn:aws:forecast:us-west-2:666488130463:what-if-analysis/
jan2020forecast/PromotionAnalysis_01G8MB3PZM89J9V1VEXCC0BS63",
  "WhatIfForecastName": "unique_name",
  "TimeSeriesTransformations": [
    {
      "Action": {
        "AttributeName": "price",
        "Operation": "MULTIPLY",
        "Value": 0.85
      },
      "TimeSeriesConditions": [
        {
          "AttributeName": "store_location",
          "AttributeValue": "tacoma",
          "Condition": "NOT_EQUALS"
        }
      ]
    }
  ]
}
```

この例では、jan2020forecast はベースライン予想で、PromotionAnalysis\_01G8MB3PZM89J9V1VEXCC0BS63 は What-If 分析の名前です。

[TimeSeriesReplacementsDataSource](#) オペレーションでは代替データセットを指定することもできます。

## SDK - Replacement Dataset

[CreateWhatIfForecast](#) オペレーションを使用して、WhatIfAnalysisName に一意の名前を指定し、ForecastArn の予想 ARN を指定します。次の例は、代替データソースのスキーマを示しています。

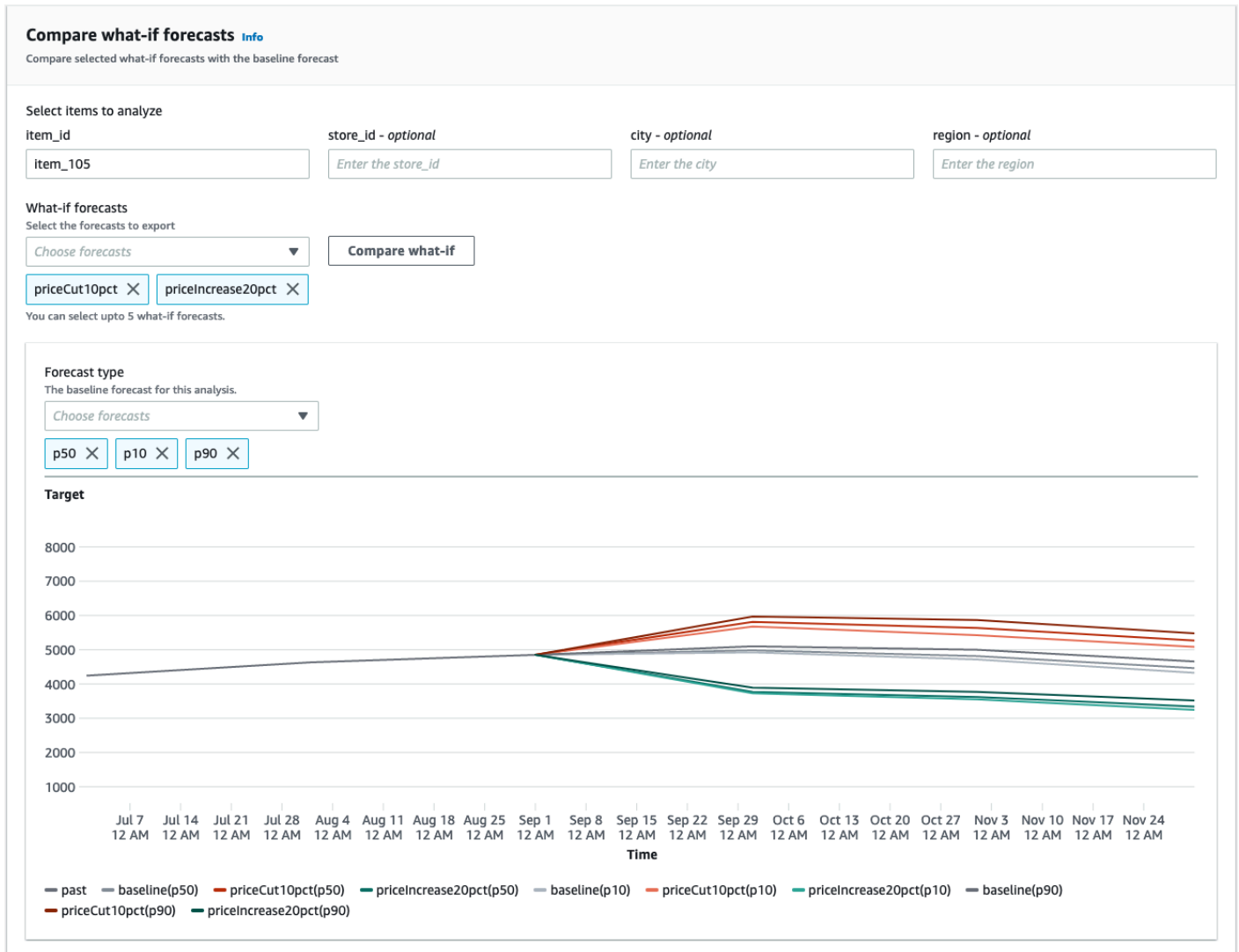
```
{
  "WhatIfAnalysisArn": "arn:aws:forecast:us-west-2:666488130463:what-if-analysis/
jan2020forecast/PromotionAnalysis_01G8MB3PZM89J9V1VEXCC0BS63",
  "WhatIfForecastName": "unique_name",
  "TimeSeriesReplacementsDataSource": {
    "S3config": {
      "Path" : "s3://bucket-name/replacementdatasource.csv",
      "RoleArn": "arn:aws:iam::acct-id:role/Role"
    },
    "Schema": {
      "Attributes" : [
        {
          "AttributeName": "item_id",
          "AttributeType": "string"
        },
        {
          "AttributeName": "timestamp",
          "AttributeType": "timestamp"
        },
        {
          "AttributeName": "price",
          "AttributeType": "float"
        },
        {
          "AttributeName": "stock_count",
          "AttributeType": "integer"
        }
      ]
    }
  }
}
```

[TimeSeriesTransformation](#) オペレーションで関連する時系列への変更を指定することもできます。

## What-If 予想を比較する

What-If 予想を比較するには、Forecast コンソールで次の手順を実行します。

1. [インサイト] ページの [What-If 分析] タブで、関心がある What-If 分析を選択します。
2. [What-If 予想の比較] セクションで、分析する項目、1 つ以上の What-If 予想、および少なくとも 1 つの予想タイプを指定します。



この例では、2 つの What-If 予想 priceCut10pct と priceIncrease20pct が、item\_105 の予想タイプ p50、p10、p90 と比較されます。グラフから、これらの予想がベースラインの時系列とどのように比較されるかを確認できます。

3. グラフにカーソルを合わせると、What-If 予想がベースライン予想とどのように比較されるかを調べることができます。

## What-If 予想をエクスポートする

Forecast コンソールまたは Forecast Software Development Kit (SDK) を使用して、What-If 予想をエクスポートできます。

### Console

What-If 予想をエクスポートするには、次のステップを実行します。

1. [インサイト] ページの [What-If 分析] タブで、関心がある What-If 分析を選択します。
2. [What-If 予想のエクスポート] セクションで、[エクスポートを作成] を選択します。
3. [What-If 予想エクスポートの作成] ページで、固有の What-If 予想エクスポート名を入力し、含める What-If 予想を指定し、[エクスポート場所]を選択して IAM ロールを指定します。
4. [Create Export] (エクスポートを作成) を選択します。

### SDK

[CreateWhatIfForecastExport](#) オペレーションを使用して、エクスポートを格納する Amazon S3 バケットを指すように「宛先」を設定します。どの What-If 予想をエクスポートするかを指定し、エクスポートに固有の名前を指定します。

```
{
  "WhatIfForecastArns": [ "arn:aws:forecast:region:acctNumber:what-if-forecast/
id1" , "arn:aws:forecast:region:acctNumber:what-if-forecast/id2" ],
  "WhatIfForecastExportName": "unique_export_name",
  "Destination": {
    "S3Config": {
      "Path": "s3://bucket/example-path",
      "RoleArn": "arn:aws:iam::000000000000:role/ExampleRole"
    }
  },
}
```

## What-If 予想をクエリする

[QueryWhatIfForecast](#) オペレーションを使用して What-If 予想をクエリします。デフォルトでは、予測の全範囲が返ります。完全な予測内で特定の日付範囲をリクエストできます。

What-If 予想をクエリする際、フィルタリング条件を指定する必要があります。フィルタはキーと値のペアです。キーは、予測の作成に使用されるいずれかのデータセットのスキーマ属性名 (予測ディメンションを含む) の 1 つです。値は、指定されたキーの有効な値です。複数のキーと値のペアを指定できます。返される What-If 予想には、すべての条件を満たすアイテムのみが含まれます。

例えば、次のコードを使用して product\_42 の What-If 予想を取得します。

```
{
  "Filters": {
    "item_id" : "product_42"
  },
  "WhatIfForecastArn": "arn:aws:forecast:region:acctNumber:what-if-forecast/
id1"
}
```

## 変換関数

変換関数は、関連する時系列の行を選択して変更する一連のオペレーションです。条件オペレーションで必要な行を選択します。次に、変換オペレーションを行って行を変更します。すべての条件は AND 演算で結合されます。つまり、変換を適用するにはすべての条件を満たす必要があります。変換は、記載された順序で適用されます。

What-If 予想を作成するときは、変換関数ビルダーを使用して適用する条件と変換を指定します。以下の画像は、この機能を示しています。

**What-if forecast details** [Info](#)

**What-if forecast name**  
A unique name to distinguish this what-if forecast from your other what-if forecast.

The what-if forecast name must have 1 to 63 characters. Valid characters: a-z, A-Z, 0-9, and \_

**What-if forecast definition method** [Info](#)  
Select how you want to define the what-if forecast. Use transformation functions for simple transforms. Use a replacement dataset for more complex transforms.

Use transformation functions  
Define the what-if forecast as a set of transformation functions on related time-series dataset

Use a replacement dataset  
Upload a dataset with just the changed related time series values

**Transformation function builder** [Info](#)  
Define your scenario by transforming your related time series. Define transformations as a set of operations to perform on the dataset

Multiply  by

where  Equals

where  Equals

Add  by

強調表示されたセクションでは、blue 色の商品について、tacoma (ワシントン州タコマなど) の店舗で、price 列に0.90を掛けています (つまり、10% の割引)。そのために、Amazon Forecast はまず、ベースラインに関連する時系列のサブセットを作成し、tacoma と等しい store の行のみを含みます。

そのサブセットは、blue と等しい color の行のみを含むようにさらに絞り込まれます。最後に、price 列のすべての値に 0.90 を掛けて、What-if 予測に使用する新しい関連時系列を作成します。

Amazon Forecast では、以下の条件がサポートされています。

- EQUALS - 列の値が条件に入力された値と同じ。
- NOT\_EQUALS - 列の値が条件に入力された値と同じではない。
- LESS\_THAN - 列の値が条件に入力された値より小さい。
- GREATER\_THAN - 列の値が条件に入力された値より大きい。

Amazon Forecast では、次のアクションがサポートされます。



- ADD - 指定した値を列のすべての行に加算する。
- SUBTRACT - 指定した値を列のすべての行に減算する。
- MULTIPLY - 指定した値を列のすべての行に乗算する。
- DIVIDE - 指定した値を列のすべての行に除算する。

SDK を使用して時系列変換を指定する方法の例を以下に示します。

### Example 1

この例では、シアトルのストアのすべての商品に 10% の割引を適用します。「City」は予測ディメンションであることに注意してください。

```
TimeSeriesTransformations=[
  {
    "Action": {
      "AttributeName": "price",
      "Operation": "MULTIPLY",
      "Value": 0.90
    },
    "TimeSeriesConditions": [
      {
        "AttributeName": "city",
        "AttributeValue": "seattle",
        "Condition": "EQUALS"
      }
    ]
  }
]
```

### Example 2

この例では、「electronics」カテゴリの全商品に 10% の割引を適用しています。「product\_category」はアイテムのメタデータであることに注意してください。

```
TimeSeriesTransformations=[
  {
    "Action": {
      "AttributeName": "price",
      "Operation": "MULTIPLY",
      "Value": 0.90
    },

```

```
"TimeSeriesConditions": [  
  {  
    "AttributeName": "product_category",  
    "AttributeValue": "electronics",  
    "Condition": "EQUALS"  
  }  
]  
}  
]
```

### Example 3

この例では、特定のアイテム ID BOA21314K に 20% のマークアップを適用しています。

```
TimeSeriesTransformations=[  
  {  
    "Action": {  
      "AttributeName": "price",  
      "Operation": "MULTIPLY",  
      "Value": 1.20  
    },  
    "TimeSeriesConditions": [  
      {  
        "AttributeName": "item_id",  
        "AttributeValue": "BOA21314K",  
        "Condition": "EQUALS"  
      }  
    ]  
  }  
]  
]
```

### Example 4

この例では、シアトル店とベルビュー店の全商品に 1 USD 加算されます。

```
TimeSeriesTransformations=[  
  {  
    "Action": {  
      "AttributeName": "price",  
      "Operation": "ADD",  
      "Value": 1.0  
    },  
    "TimeSeriesConditions": [  
      {  
        "AttributeName": "store",  
        "AttributeValue": "Seattle",  
        "Condition": "EQUALS"  
      },  
      {  
        "AttributeName": "store",  
        "AttributeValue": "Bellevue",  
        "Condition": "EQUALS"  
      }  
    ]  
  }  
]  
]
```

```
{
  "AttributeName": "city",
  "AttributeValue": "seattle",
  "Condition": "EQUALS"
}
],
{
  "Action": {
    "AttributeName": "price",
    "Operation": "ADD",
    "Value": 1.0
  },
  "TimeSeriesConditions": [
    {
      "AttributeName": "city",
      "AttributeValue": "bellevue",
      "Condition": "EQUALS"
    }
  ]
}
]
```

### Example 5

この例では、2022年9月のシアトルの全商品から1 USDを差し引きます。

```
TimeSeriesTransformations=[
  {
    "Action": {
      "AttributeName": "price",
      "Operation": "SUBTRACT",
      "Value": 1.0
    },
    "TimeSeriesConditions": [
      {
        "AttributeName": "city",
        "AttributeValue": "seattle",
        "Condition": "EQUALS"
      },
      {
        "AttributeName": "timestamp",
        "AttributeValue": "2022-08-31 00:00:00",
        "Condition": "GREATER_THAN"
      }
    ]
  }
]
```

```
    },
    {
      "AttributeName": "timestamp",
      "AttributeValue": "2022-10-01 00:00:00",
      "Condition": "LESS_THAN"
    }
  ]
}
```

### Example 6

この例では、価格に最初に 10 を掛け、次に 5 USD を価格から差し引きます。アクションは、宣言された順序で適用されることに注意してください。

```
TimeSeriesTransformations=[
  {
    "Action": {
      "AttributeName": "price",
      "Operation": "MULTIPLY",
      "Value": 10.0
    },
    "TimeSeriesConditions": [
      {
        "AttributeName": "city",
        "AttributeValue": "seattle",
        "Condition": "EQUALS"
      }
    ]
  },
  {
    "Action": {
      "AttributeName": "price",
      "Operation": "SUBTRACT",
      "Value": 5.0
    },
    "TimeSeriesConditions": [
      {
        "AttributeName": "city",
        "AttributeValue": "seattle",
        "Condition": "EQUALS"
      }
    ]
  }
]
```

```
}  
]
```

### Example 7

この例では空のセットを作成しているため、アクションはどの時系列にも適用されません。このコードは、シアトルとベルビューにあるすべての商品の価格を変更しようとしています。条件は AND 演算で結合され、店舗は 1 つの都市にしか存在できないため、結果は空のセットになります。そのため、アクションは適用されません。

```
TimeSeriesTransformations=[  
  {  
    "Action": {  
      "AttributeName": "price",  
      "Operation": "MULTIPLY",  
      "Value": 10.0  
    },  
    "TimeSeriesConditions": [  
      {  
        "AttributeName": "city",  
        "AttributeValue": "seattle",  
        "Condition": "EQUALS"  
      },  
      {  
        "AttributeName": "city",  
        "AttributeValue": "bellevue",  
        "Condition": "EQUALS"  
      },  
    ]  
  }  
]
```

複数の属性に条件を適用する方法の例については、例 4 を参照してください。

### Example 8

タイムスタンプを使用する変換条件は、未加工データではなく、境界が揃ったデータに適用されます。例えば、データを 1 時間ごとに入力し、予想を 1 日ごとに入力するとします。この場合、Forecast はタイムスタンプをその日に揃えるので、2020-12-31 01:00:00 は 2020-12-31 00:00:00 と合っています。このコードでは、境界に沿ったタイムスタンプのタイムスタンプが指定されていないため、空のセットが作成されます。

```
TimeSeriesTransformations=[
  {
    "Action": {
      "AttributeName": "price",
      "Operation": "MULTIPLY",
      "Value": 10.0
    },
    "TimeSeriesConditions": [
      {
        "AttributeName": "timestamp",
        "AttributeValue": "2020-12-31 01:00:00",
        "Condition": "EQUALS"
      }
    ]
  }
]
```

## 代替データセット

代替データセットは、What-If 予想で変更したい値のみを含むベースライン関連の時系列を修正したバージョンです。代替データセットには、ベースライン関連の時系列の予想ディメンション、アイテム識別子、タイムスタンプと、少なくとも 1 つの変更された時系列が含まれている必要があります。このデータセットをベースライン関連の時系列と統合して、What-If 予想に使用される変換済みデータセットを作成します。代替データセットは CSV 形式である必要があります。

このデータセットには、同じ時系列のタイムスタンプが重複してはいけません。

次に、代替時系列を指定する方法とそれらの仕様の解釈方法の例をいくつか示します。日次予測を行っていて、予想期間が 2022 年 8 月 1 日から 2022 年 8 月 3 日までの場合を考えてみましょう。すべての例のベースライン関連の時系列を以下の表に示します。

item_id	timestamp	price	stock_count
item_1	2022-08-01	100	50
item_1	2022-08-02	100	50
item_1	2022-08-03	100	50
item_2	2022-08-01	75	500

item_id	timestamp	price	stock_count
item_2	2022-08-02	75	500
item_2	2022-08-03	75	500

## Unchanged values

2022-08-02 と 2022-08-03 の item\_1 に 10% の割引を適用するには、代替データセットに以下を指定すれば十分です。

### 代替データセット

item_id	timestamp	price
item_1	2022-08-02	90
item_1	2022-08-03	90

ただし、代替データセットでは変更されていない値を指定することも有効です。代替データセットとして使用すると、次の 3 つのテーブルはそれぞれ、以前に提供したテーブルと同じ結果になります。

### 列が変更されていない代替データセット

item_id	timestamp	price	stock_count
item_1	2022-08-02	90	50
item_1	2022-08-03	90	50

### 行が変更されていない代替データセット

item_id	timestamp	price
item_1	2022-08-01	100
item_1	2022-08-02	90

item_id	timestamp	price
item_1	2022-08-03	90
item_2	2022-08-01	75
item_2	2022-08-02	75
item_2	2022-08-03	75

行と列が変更されていない代替データセット

item_id	timestamp	price	stock_count
item_1	2022-08-01	100	50
item_1	2022-08-02	90	50
item_1	2022-08-03	90	50
item_2	2022-08-01	75	500
item_2	2022-08-02	75	500
item_2	2022-08-03	75	500

## Missing values

代替時系列の欠損値は、ベースライン関連の時系列の値に置き換えられます。2022年8月2日と2022年8月3日に item\_1 に 10% の割引を適用し、2022年8月1日に item\_2 の在庫を増やすシナリオを考えてみましょう。この代替データセットで十分です。

値が欠落している代替データセット

item_id	timestamp	price	stock_count
item_1	2022-08-02	90	
item_1	2022-08-03	90	



item_id	timestamp	price	stock_count
item_2	2022-08-01		5000

このテーブルにない値は、ベースライン関連の時系列から代入されます。

### Extraneous values

What-If 予想を作成する場合、代替時系列の余分な値は無視されます。つまり、ベースラインに関連する時系列の値と一致しない代替データセットの値はモデル化されません。次の代替データセットを考えてみましょう。

#### 無関係な値を含む代替データセット

item_id	timestamp	price	stock_count
item_1	2022-08-01	100	50
item_1	2022-08-02	100	50
item_1	2022-08-03	100	50
item_2	2022-08-01	75	500
item_2	2022-08-02	75	500
item_2	2022-08-03	75	500
item_3	2022-08-01	50	125
item_3	2022-08-02	50	125
item_3	2022-08-03	50	125

item\_3 を含む行は無視され、What-If 分析には含まれません。

### Historical changes

予測範囲外の代替データセットの変更は無視されます。次の代替データセットを考えてみましょう。

## 予測範囲外の値を含む代替データセット

item_id	timestamp	price	stock_count
item_1	2022-07-31	100	50
item_1	2022-08-01	100	50
item_1	2022-08-02	100	50
item_1	2022-08-03	100	50
item_1	2022-08-04	100	50
item_2	2022-07-31	75	500
item_2	2022-08-01	75	500
item_2	2022-08-02	75	500
item_2	2022-08-03	75	500
item_3	2022-08-04	75	500

2022-07-31 と 2022-08-04 を含む行は無視され、What-If 分析には含まれません。

## 予測ディメンション

データセットに予想ディメンションを含める場合は、それを代替データセットに含める必要があります。ベースラインに関連する次の時系列を考えてみましょう。

item_id	store_id	timestamp	price	stock_count
item_1	store_1	2022-08-01	100	50
item_1	store_1	2022-08-02	100	50
item_1	store_1	2022-08-03	100	50
item_1	store_2	2022-08-01	75	500

item_id	store_id	timestamp	price	stock_count
item_1	store_2	2022-08-02	75	500
item_1	store_2	2022-08-03	75	500

したがって、2022年8月2日に全店舗での10%割引の代替データセットは次のようになります。

item_id	store_id	timestamp	price
item_1	store_1	2022-08-02	90
item_1	store_2	2022-08-02	67.5

# リソースの管理

進行中のジョブを停止し、完了または失敗したリソースを削除し、リソースにタグを付け、Amazon EventBridge および Amazon CloudWatch Events を介してイベント通知を設定することにより、Amazon Forecast リソースを管理できます。

トピック

- [リソースの停止](#)
- [リソースの削除](#)
- [Amazon Forecast のリソースのタグ付け](#)
- [ジョブステータス通知の受信](#)

## リソースの停止

Amazon Forecast Stop Resource ([StopResource](#)) オペレーションは、進行中のリソースジョブを停止します。次のリソースジョブを停止できます。

- データセットグループのインポート (CreateDatasetImportJob)
- Predictor のトレーニング (CreateAutoPredictor および CreatePredictor)
- 予測子バックテストエクスポート (CreatePredictorBacktestExportJob)
- Forecast (CreateForecast)
- Forecast エクスポート (CreateForecastExportJob)
- What-If 分析 (CreateWhatIfAnalysis)
- What-If 予想 (CreateWhatIfForecast)
- What-If 予想エクスポート (CreateWhatIfForecastExportJob)

リソースジョブが停止すると、再開することはできません。

リソースを停止するとワークフローは終了しますが、リソースは削除されません。コンソールおよび [Describe](#) オペレーションでリソースパラメータをプレビューすることは引き続き可能です。

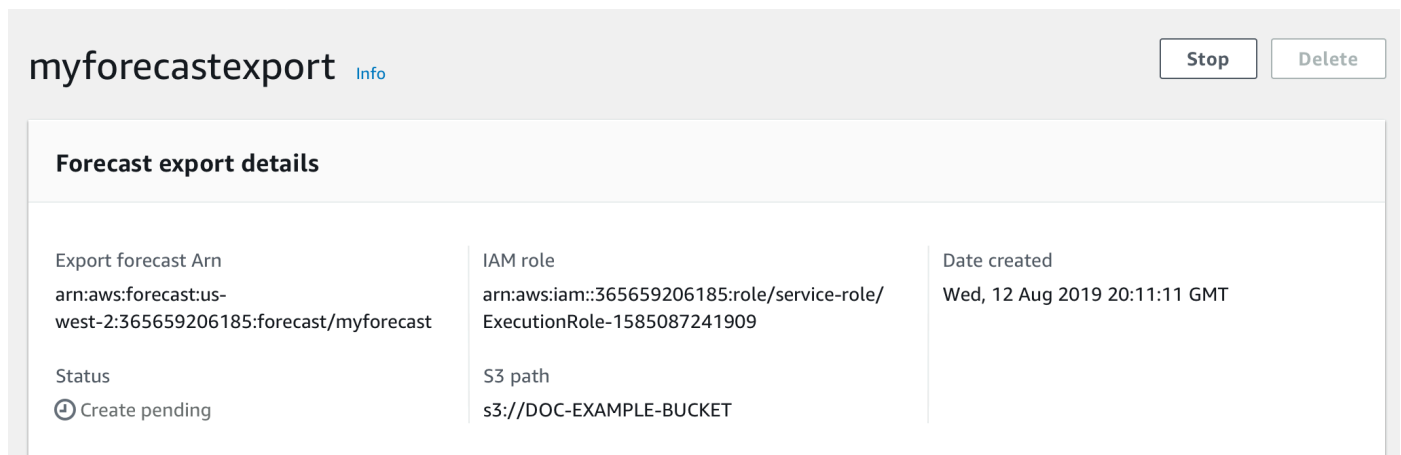
予測子または予測ジョブを停止すると、ジョブが停止した時点までに使用されたリソースについての料金が請求されます。

Forecast コンソールまたは AWS Software Development Kit (SDK) を使用して、リソースジョブを停止できます。

## Console

リソースジョブを停止するには

1. AWS Management Console にサインインして、Amazon Forecast コンソール (<https://console.aws.amazon.com/forecast/>) を開きます。
2. ナビゲーションペインで、リソースタイプを選択します。
3. リソースジョブを選択します。
4. [Stop] (停止) を選択します。



The screenshot shows the Amazon Forecast console interface for a resource job named "myforecastexport". At the top right, there are "Stop" and "Delete" buttons. Below the job name, there is a section titled "Forecast export details" containing a table of metadata.

Forecast export details		
Export forecast Arn arn:aws:forecast:us-west-2:365659206185:forecast/myforecast	IAM role arn:aws:iam::365659206185:role/service-role/ExecutionRole-1585087241909	Date created Wed, 12 Aug 2019 20:11:11 GMT
Status 🕒 Create pending	S3 path s3://DOC-EXAMPLE-BUCKET	

## SDK

リソースジョブを停止するには

[StopResource](#) オペレーションを使用して、ResourceArn の値を、停止するリソースジョブを識別する Amazon リソースネーム (ARN) に設定します。

```
{
  "ResourceArn": "arn:partition:service:region:account-id:resource-id"
}
```

# リソースの削除

Amazon Forecast コンソールと AWS Software Development Kit (SDK) を使用して、個々の Amazon Forecast リソースとリソースツリー全体を削除できます。

Forecast のリソースツリーは、親子の階層構造です。子リソースは、他のリソースから作成されたリソースです。例えば、データセットグループを使用して予測子を作成する場合、データセットグループは親リソースであり、予測子は子リソースです。Forecast のリソースを削除するときは、その子リソースも削除する必要があります。

リソースまたはリソースツリーの削除は、元に戻せないアクションです。開始後に停止することはできません。

## トピック

- [リソースツリーについて](#)
- [個々のリソースの削除](#)
- [リソースツリーの削除](#)

## リソースツリーについて

Forecast のリソースツリーは、親子の階層構造です。子リソースは、別のリソースから作成されたリソースです。例えば、予測が予測子から生成される場合、予測は子リソースであり、予測子は親リソースです。

Forecast リソースを削除するには、そのリソースツリー全体も削除する必要があります。これには、親リソースのすべての子リソースと、それらの子リソースの子リソースも含まれます。

### Note

リソースツリーを削除すると、Amazon Forecast のリソースのみが削除されます。Amazon Simple Storage Service (Amazon S3) に保存されているデータセットやエクスポートされたファイルは削除されません。

Forecast のリソースには、次の親子リソース階層があります。

例えば、予測子のリソースツリーには、予測子バックテストジョブ、予測、および予測エクスポートジョブが子リソースとして含まれます。予測のリソースツリーには、子リソースとしての予測エクスポートジョブのみが含まれます。

[dataset] (データセット) のリソースツリーには、子リソースとして [dataset import jobs] (データセットのインポートジョブ) が含まれています。データセットもデータセットのインポートジョブも、データセットグループのリソースツリーの一部ではありません。

親リソース	子リソース
データセット	データセットのインポートジョブ
データセットグループ	予測子、予測子バックテストエクスポートジョブ、Predictor Explainability、Predictor Explainability エクスポート、予測、予測エクスポートジョブ、Forecast Explainability、Forecast Explainability エクスポート
予測子	予測子バックテストエクスポートジョブ、Predictor Explainability、Predictor Explainability エクスポート、予測、予測エクスポートジョブ、Forecast Explainability、Forecast Explainability エクスポート
Forecast	予想エクスポートジョブ、Forecast Explainability、Forecast Explainability エクスポート、What-If 分析、What-If 予想、What-If 予想エクスポート
Explainability	Explainability のエクスポート
What-if 分析	What-If 予想、What-If 予想エクスポート
What-If 予想	What-If 予想エクスポート

リソースに子リソースがない場合は、個別に削除できます。リソースに子リソースがある場合は、リソースツリー全体を削除する必要があります。

Forecast コンソールを使用している場合、子リソースを含むリソースを削除すると、リソースツリー全体を削除するよう指示するプロンプトが自動的に表示されます。AWS Software Development Kit (SDK) を使用する場合は、[DeleteResourceTree](#) オペレーションを使用してリソースツリーを削除します。

## 個々のリソースの削除

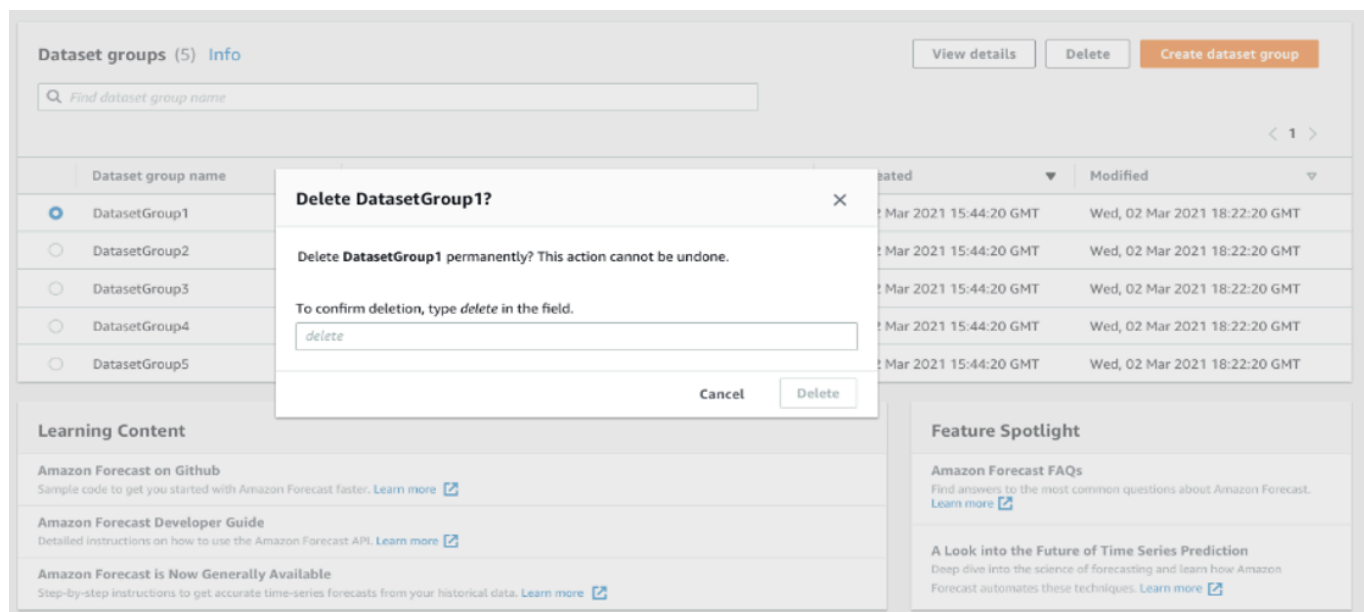
子リソースに関連付けられていない場合は、個々のリソースを削除できます。例えば、予測の作成やジョブのエクスポートに使用されていない個々の予測子を削除できます。

Amazon Forecast コンソールまたは AWS Software Development Kit (SDK) を使用して、リソースを削除できます。

### Console

リソースを削除するには

1. AWS Management Console にサインインして、Amazon Forecast コンソール (<https://console.aws.amazon.com/forecast/>) を開きます。
2. ナビゲーションペインで、削除するリソースのリソースタイプを選択します。
3. リソースを選択し、[Delete] (削除) を選択します。
4. 確認フィールドで、**delete** と入力します。
5. [Delete] (削除) をクリックします。



### SDK

リソースを削除するには



リソースの削除に使用するオペレーションは、そのリソースタイプによって異なります。削除するリソースタイプのオペレーションで、リソースの Amazon リソースネーム (ARN) を指定します。

- [DeleteDataset](#)
- [DeleteDatasetGroup](#)
- [DeleteDatasetImportJob](#)
- [DeletePredictor](#)
- [DeletePredictorBacktestExportJob](#)
- [DeleteForecast](#)
- [DeleteForecastExportJob](#)
- [DeleteExplainability](#)

例えば、[DeletePredictor](#) オペレーションを使用して予測子を削除するには、削除する予測子の ARN に `PredictorArn` の値を指定します。

```
{
  "PredictorArn": arn:partition:service:region:account-id:resource-id
}
```

## リソースツリーの削除

リソースツリーを削除すると、親リソースと関連するすべての子リソースが削除されます。例えば、予測子と、その予測子に関連付けられているすべての子リソース (予測子バックテストエクスポートジョブ、予測、および予測エクスポートジョブ) を削除できます。親リソースを指定して、リソースツリーを削除します。

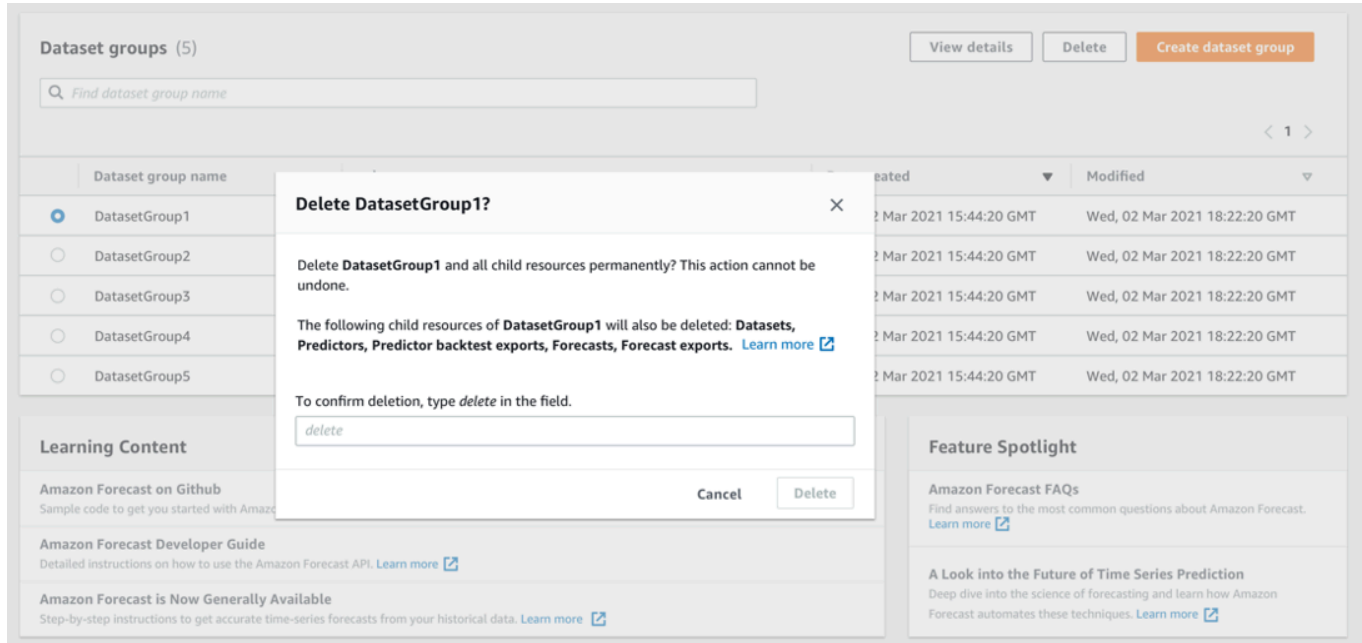
Amazon Forecast コンソールまたは AWS Software Development Kit (SDK) を使用して、リソースツリーを削除できます。

### Console

リソースツリーを削除するには

1. AWS Management Console にサインインして、Amazon Forecast コンソール (<https://console.aws.amazon.com/forecast/>) を開きます。

2. ナビゲーションペインで、親リソースのリソースタイプを選択します。
3. 削除する親リソースを選択して、[Delete] (削除) を選択します。
4. 確認フィールドで、**delete** と入力します。
5. [Delete] (削除) をクリックします。



## SDK

リソースツリーを削除するには

リソースツリーを削除するには、[DeleteResourceTree](#) オペレーションを使用します。ResourceArn の値を親リソースの Amazon リソースネーム (ARN) に設定します。

```
{
  "ResourceArn": arn:partition:service:region:account-id:resource-id
}
```

## Amazon Forecast のリソースのタグ付け

タグは、特定のタイプの AWS リソースなど、Amazon Forecast リソースをオプションで定義および関連付けるラベルです。タグを使用することで、目的、所有者、環境、その他の条件など、さまざまな方法でリソースを分類および管理できます。例えば、タグを使用してポリシーや自動化を適用し

たり、特定のコンプライアンス要件の対象となるリソースを識別したりできます。以下のタイプの Forecast リソースにタグを追加できます。

- データセットグループ
- データセット
- データセットのインポートジョブ
- 予測子
- 予測子エクスポートジョブ
- 予測
- 予測のエクスポートジョブ
- What-If 分析
- What-If 予想
- What-If 予想エクスポートジョブ

リソースには、最大 50 個のタグを含めることができます。

## タグを管理する

タグはそれぞれ、1 つの必須タグキーとオプションの 1 つのタグ値で構成されており、どちらもお客様側が定義します。タグキーは、より具体的なタグ値のカテゴリのように動作する、一般的なラベルです。タグ値は、タグキーの記述子として機能します。例えば、Forecast のデータセットのインポートジョブに内部テスト用と本番用の 2 つのバージョンがある場合、両方のプロジェクトに Environment タグキーを割り当てます。Environment タグキーの値は、場合によって、データセットのインポートジョブの 1 つのバージョンでは Test に、他のバージョンでは Production になります。

タグキーには最大 128 文字を含めることができます。タグ値は最大 256 文字を含めることができます。Unicode 文字、数字、空白、または次の記号のいずれか 1 つを使用できます。\_。:/=+- さらに、タグには以下のような制限が追加されます。

- タグのキーと値は大文字と小文字が区別されます。
- タグキーは、関連付けられたリソースごとにそれぞれ一意である必要があります。また、設定できる値は 1 つのみです。
- aws:、AWS:、またはその大文字または小文字の組み合わせを、キーのプレフィックスとして使用しないでください。これらの文字列は AWS による使用のために予約されています。このプレ

フィックスが含まれるタグキーを編集したり削除することはできません。値にはこのプレフィックスを付けることができます。タグ値には aws がプレフィックスとして付されているが、キーには付されていない場合、Forecast は、それをユーザータグとみなし、タグ数の上限である 50 個を計算する際に算入します。aws のキープレフィックスのみを持つタグは、リソース制限あたりのタグに算入されません。

- タグのみに基づいて、リソースを更新または削除することはできません。使用する操作に応じて、Amazon リソースネーム (ARN) またはリソース ID も指定する必要があります。
- パブリックリソースまたは共有リソースにタグを関連付けることができます。ただし、タグは自分の AWS アカウントにのみ使用でき、リソースを共有する他のアカウントには使用できません。さらに、AWS アカウント用に指定された AWS リージョンにあるリソースにのみタグを使用できます。

Forecast リソースからタグキーと値を追加、表示、更新、削除するには、AWS Command Line Interface (AWS CLI)、Forecast API、または AWS SDK を使用できます。

## IAM ポリシーでのタグの使用

タグの実装を開始した後、タグベースのリソースレベルのアクセス許可を AWS Identity and Access Management (IAM) ポリシーと API オペレーションに適用できます。これには、リソースの作成時にリソースへのタグの追加をサポートするオペレーションが含まれます。このようにタグを使用することで、AWS アカウントのどのグループとユーザーにリソースの作成とタグ付けのアクセス許可があって、どのグループとユーザーにタグの作成、更新、削除のアクセス許可があるかを、広範囲にきめ細かく制御できます。

例えば、リソースの Owner タグの値がユーザー名となっているすべての Forecast リソースに対して、ユーザーにフルアクセスを許可するポリシーを作成できます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ModifyResourceIfOwner",
      "Effect": "Allow",
      "Action": "forecast:*",
      "Resource": "*",
      "Condition": {
        "StringEqualsIgnoreCase": {
          "aws:ResourceTag/Owner": "${aws:username}"
        }
      }
    }
  ]
}
```

```
    }
  }
}
]
```

次の例は、データセットの作成と削除を許可するポリシーを作成する方法を示しています。これらのオペレーションは、ユーザー名が johndoe の場合にのみ許可されます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "forecast:CreateDataset",
        "forecast>DeleteDataset"
      ],
      "Resource": "arn:aws:forecast:*:*:dataset/*",
      "Condition": {
        "StringEquals": {"aws:username" : "johndoe"}
      }
    },
    {
      "Effect": "Allow",
      "Action": "forecast:DescribeDataset",
      "Resource": "*"
    }
  ]
}
```

タグをベースにしてリソースレベルでアクセス許可を定義した場合、そのアクセス許可は即座に反映されます。つまり、リソースが作成されるとすぐにリソースの安全性が増し、新しいリソースにタグの使用をすぐに強制できるようになります。リソースレベルのアクセス許可を使用して、新しいリソースと既存のリソースに、どのタグキーと値を関連付けるかを制御することもできます。詳細については、AWS IAM ユーザーガイドの「[タグを使用したアクセス制御](#)」を参照してください。

## リソースにタグを追加する

次の例は、[AWS CLI](#) と AWS Management Console を使用して Forecast リソースにタグを追加する方法を示しています。

## AWS CLI

AWS CLI を使用してリソースを新しく作成するときにタグを追加する場合は、リソースに対して適切な create コマンドを使用し、tags パラメータと値を含めます。例えば、次のコマンドは、CUSTOM ドメインの myDatasetGroup という名前の新しいデータセットグループを作成し、Test タグ値を含む Environment タグキー、Owner タグキーと xyzCorp 値というタグを追加します。

```
aws forecast create-dataset-group \  
--dataset-group-name myDatasetGroup \  
--dataset-arns arn:aws:forecast:region:acct-id:dataset/dataset_name \  
--domain CUSTOM \  
--tags Key=Environment,Value=Test Key=Owner,Value=xyzCorp
```

Forecast リソースの作成に使用できるコマンドの詳細については、[Forecast AWS CLI コマンド レファレンス](#)を参照してください。

既存のリソースにタグを追加するには、tag-resource コマンドを使用してリソースの ARN を指定し、tags-model パラメータにタグキーと値を指定します。

```
aws forecast tag-resource \  
--resource-arn resource ARN \  
--tags Key=key,Value=value
```

## AWS Management Console

Forecast でリソースを作成するときに、オプションのタグを追加できます。次の例では、データセットグループにタグを追加します。他のリソースへのタグの追加も同様のパターンに従います。

新しいデータセットグループにタグを追加するには

1. AWS Management Console にサインインして、Amazon Forecast コンソール (<https://console.aws.amazon.com/forecast/>) を開きます。
2. [データセットグループの作成] を選択します。
3. [Dataset group name] (データセットグループ名) で、名前を入力します。
4. [Forecasting domain] (予測ドメイン) で、ドメインを選択します。
5. [新しいタグを追加] をクリックします。

6. [Key] (キー) と [Value] (値) で、適切な値を入力します。

たとえば、それぞれ **Environment** および **Test** などです。

7. タグをさらに追加するには、[Add new tag] (新しいタグを追加) を選択します。

1 つのリソースに最大 50 個のタグを追加できます。

8. [Next] (次へ) を選択し、リソースの作成を続行します。

## 追加情報

タグ付けの詳細については、次のリソースを参照してください。

- [AWS Tagging Principles](#) (AWS 全般のリファレンス)
- [AWS Tagging Strategies](#) (ダウンロード可能な PDF)
- [AWS Access Control](#) (AWS IAM ユーザーガイド)
- [AWS Tagging Policies](#) (AWS Organizations ユーザーガイド)

## ジョブステータス通知の受信

Amazon EventBridge または Amazon CloudWatch Events に、予測子や予測の作成など、進行中の Amazon Forecast リソースジョブのステータスアップデートを通知させることができます。EventBridge および CloudWatch Events では、アマゾン ウェブ サービス (AWS) リソースの変更を記述した、システムイベントのほぼリアルタイムのストリーミングを配信します。例えば、Forecast の予測子がトレーニングを終了したときに通知するイベントを設定できます。

イベントは、ベストエフォートベースで出力されます。イベントの詳細については、[Amazon EventBridge ユーザーガイド](#) または [Amazon CloudWatch Events ユーザーガイド](#) を参照してください。

### Note

イベントの管理には Amazon EventBridge を使用することをお勧めします。CloudWatch Events と EventBridge は同じ API を使用し、同じ機能を提供しますが、EventBridge はより多くの機能を提供します。CloudWatch または EventBridge のいずれかで行った変更は、各コンソールに表示されます。詳細については、[Amazon EventBridge](#) を参照してください。

## トピック

- [Forecast リソースジョブのモニタリング](#)
- [ジョブステータス通知用の EventBridge ルールの作成](#)
- [ジョブステータス通知用の CloudWatch Events ルールの作成](#)

## Forecast リソースジョブのモニタリング

イベントは AWS 環境の変化を示し、ルールは着信イベントを照合し、処理のためにそれらをターゲットにルーティングします。Forecast イベントを照合するルールを設定して、1 つ以上のターゲット関数またはストリームにルートできます。EventBridge および CloudWatch Events は、発生したイベントを検出し、照合ルールでターゲットを呼び出します。

次の表に、モニタリングできる Forecast リソースジョブとそのステータス変更イベントを示します。

リソースジョブ	ステータス変更のイベント名	ステータス
<a href="#">CreateDatasetImportJob</a>	Forecast Dataset Import Job State Change	ACTIVE、CREATE_IN_PROGRESS、CREATE_FAILED、CREATE_STOPPED
<a href="#">CreatePredictor</a>	Forecast Predictor Creation State Change	ACTIVE、CREATE_IN_PROGRESS、CREATE_FAILED、CREATE_STOPPED
<a href="#">CreateForecast</a>	Forecast Forecast Creation State Change	ACTIVE、CREATE_IN_PROGRESS、CREATE_FAILED、CREATE_STOPPED
<a href="#">CreateExplainability</a>	Forecast の説明可能性作成の状態変更	ACTIVE、CREATE_IN_PROGRESS、CREATE_FAILED、CREATE_STOPPED
<a href="#">CreatePredictorBacktestExportJob</a>	Forecast Predictor Backtest Export Job State Change	ACTIVE、CREATE_IN_PROGRESS、CREATE_FAILED、CREATE_STOPPED



リソースジョブ	ステータス変更のイベント名	ステータス
<a href="#">CreateForecastExportJob</a>	Forecast Forecast Export Job State Change	ACTIVE、CREATE_IN_PROGRESS、CREATE_FAILED、CREATE_STOPPED
<a href="#">CreateExplainabilityExport</a>	Forecast の説明可能性エクスポート作成の状態変更	ACTIVE、CREATE_IN_PROGRESS、CREATE_FAILED、CREATE_STOPPED
<a href="#">CreateWhatIfAnalysis</a>	Forecast の What-If 分析作成の状態変更	ACTIVE、CREATE_IN_PROGRESS、CREATE_FAILED、CREATE_STOPPED
<a href="#">CreateWhatIfForecast</a>	Forecast の What-if 予測作成の状態変更	ACTIVE、CREATE_IN_PROGRESS、CREATE_FAILED、CREATE_STOPPED
<a href="#">CreateWhatIfForecastExport</a>	Forecast の What-if 予測エクスポート作成の状態変更	ACTIVE、CREATE_IN_PROGRESS、CREATE_FAILED、CREATE_STOPPED
<a href="#">DeleteDataset</a>	Forecast Dataset Deletion State Change	DELETE_IN_PROGRESS、DELETE_FAILED
<a href="#">DeleteDatasetImportJob</a>	Forecast Dataset Import Job Deletion State Change	DELETE_IN_PROGRESS、DELETE_FAILED
<a href="#">DeletePredictor</a>	Forecast Predictor Deletion State Change	DELETE_IN_PROGRESS、DELETE_FAILED
<a href="#">DeleteForecast</a>	Forecast Forecast Deletion State Change	DELETE_IN_PROGRESS、DELETE_FAILED
<a href="#">DeleteExplainability</a>	Forecast の説明可能性削除の状態変更	DELETE_IN_PROGRESS、DELETE_FAILED
<a href="#">DeleteExplainabilityExport</a>	Forecast の説明可能性エクスポート削除の状態変更	DELETE_IN_PROGRESS、DELETE_FAILED

リソースジョブ	ステータス変更のイベント名	ステータス
<a href="#">DeleteWhatIfAnalysis</a>	Forecast の What-If 分析削除の状態変更	DELETE_IN_PROGRESS、DELETE_FAILED
<a href="#">DeleteWhatIfForecast</a>	Forecast の What-If 予測削除の状態変更	DELETE_IN_PROGRESS、DELETE_FAILED
<a href="#">DeleteWhatIfForecastExportJob</a>	Forecast の What-If 予測エクスポート削除の状態変更	DELETE_IN_PROGRESS、DELETE_FAILED

通知には、Amazon リソースネーム (ARN)、ジョブステータス、ジョブ期間 (分)、エラーメッセージ (ジョブが失敗した場合) など、リソースに関する情報が含まれます。削除イベントの通知には Duration フィールドが含まれていません。通知例を次に示します。

```
{
  "version": "0",
  "id": "017fcb6d-7ca3-ebf8-819e-3e0fa956ee17",
  "detail-type": "Forecast Dataset Import Job State Change",
  "source": "aws.forecast",
  "account": "000000000001",
  "time": "2021-02-19T05:45:51Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:forecast:us-west-2:000000000001:dataset/example_data"
  ],
  "detail": {
    "Arn": "arn:aws:forecast:us-west-2:000000000001:dataset/example_data",
    "Duration": 60,
    "Status": "ACTIVE",
  }
}
```

## ジョブステータス通知用の EventBridge ルールの作成

進行中の Forecast リソースジョブのステータス変更を通知する EventBridge ルールを作成するには、「Amazon EventBridge ユーザーガイド」の「[Creating a rule for an AWS service](#)」を参照してください。手順で、[Service name] (サービス名) に、[Amazon Forecast] を選択します。[Event type]

(イベントタイプ) で、モニタリングする Forecast のイベントを選択します。Forecast のイベントのリストについては、「[Forecast リソースジョブのモニタリング](#)」を参照してください。

## ジョブステータス通知用の CloudWatch Events ルールの作成

進行中の Forecast リソースジョブのステータス変更を通知する CloudWatch Events ルールを作成するには、「Amazon CloudWatch ユーザーガイド」の「[イベントでトリガーする CloudWatch Events のルールの作成](#)」を参照してください。手順で、[Service name] (サービス名) に、[Amazon Forecast] を選択します。[Event type] (イベントタイプ) で、モニタリングする Forecast のイベントを選択します。Forecast のイベントのリストについては、「[Forecast リソースジョブのモニタリング](#)」を参照してください。

# ガイドラインとクォータ

次のセクションには、Amazon Forecast のガイドラインとクォータに関する情報が含まれています。

## トピック

- [AWS でサポートされているリージョン](#)
- [コンプライアンス](#)
- [Service Quotas](#)
- [条件と制限](#)

## AWS でサポートされているリージョン

Forecast をサポートしている AWS リージョンのリストについては、アマゾン ウェブ サービス全般のリファレンスの「[AWS リージョンとエンドポイント](#)」をご参照ください。

## コンプライアンス

Amazon Forecast コンプライアンスプログラムの詳細については、「[AWS コンプライアンス](#)」、「[AWS コンプライアンスプログラム](#)」、および「[コンプライアンスプログラムによる AWS 対象範囲内のサービス](#)」を参照してください。

## Service Quotas

### Note

調整可能なクォータの引き上げをリクエストするには、[Service Quotas コンソール](#)を使用し、「Service Quotas ユーザーガイド」の「[Requesting a quota increase](#)」(クォータ引き上げリクエスト)のセクションの手順に従います。

Forecast には、次のサービスクォータがあります。

CreateDatasetImportJob API によって課されるクォータ

リソース	デフォルトのクォータ	調整可能
Amazon S3 バケットのファイルの最大数	10,000	いいえ
Amazon S3 バケット内のすべてのファイルの最大累積サイズ	30 GB	はい
データセットグループのデータセットの最大数	3 (タイプごとに 1)	いいえ
データセットの行の最大数	30 億 注: ap-south-1 リージョンのクォータは 10 億です。	はい
ターゲット時系列データセットの列の最大数 (必須列 + 追加の予測ディメンション)	13 (3 + 10)	いいえ
関連する時系列データセットの列の最大数 (必須列 + 追加の予測ディメンション + 関連する特徴)	25 (2 + 10 + 13)	いいえ
項目メタデータデータセットの列の最大数	10	いいえ
他のデータセットの列の最大数	36	いいえ

CreatePredictor API によって課されるクォータ

リソース	デフォルトのクォータ	調整可能
最大 number of backtest windows ( <a href="#">EvaluationParameters</a> )	5	いいえ
予測子あたりの時系列の最大数  (項目数 X ターゲット時系列データセットの予測次元の固有値の数)	すべてのターゲット時系列の項目および次元で 5,000,000  注: ap-south-1 リージョンのクォータは 1,000,000 です。  100,000 アイテムを超える場合、Forecast は、よりきめ細かい頻度 (1 時間ごとなど) ではなく、年次、月次、週次、および日次の頻度をサポートします。	はい
最大予測期間	CNN-QR、DeepAR+、AutoML: 500 データポイントまたはターゲット時系列データセットの長さの 1/3 の小さい方  ETS、NPTS、Prophet、ARIMA: 500 のデータポイントまたはターゲット時系列データセットの長さから 1 を減じたもののいずれか小さい方。	いいえ

## 一般的なリソースのクォータ

リソース	デフォルトのクォータ	調整可能
CreateDatasetImportJob タスクの最大同時実行数	3	はい
CreatePredictor タスクの最大同時実行数	3	はい
AutoML を使用した CreatePredictor タスクの最大同時実行数	3	はい
CreateAutoPredictor タスクの最大同時実行数	3	いいえ
最大並列実行 CreateExplainability タスク	3	いいえ
最大並列実行 CreateExplainabilityExport タスク	3	いいえ
CreatePredictorBacktestExportJob タスクの最大同時実行数	3	はい
CreateForecast タスクの最大同時実行数	3	はい
CreateForecastExportJob タスクの最大同時実行数	3	はい
リソースタイプあたりの最大並列実行 StopResource タスク	3	はい
データセットの最大数	1500	はい

リソース	デフォルトのクォータ	調整可能
データセットグループの最大数	500	はい
データセットインポートジョブの最大数	1,000	はい
予測子の最大数	500	はい
の最大数 AutoPredictors	500	いいえ
予測子バックテストエクスポートジョブの最大数	1,000	はい
予測の最大数	100	はい
予測エクスポートジョブの最大数	1,000	はい
予測をクエリできる最大時間 (コンソールまたは <a href="#">QueryForecast API</a> )	30 日間	いいえ
リソースに追加できるタグの最大数	50	いいえ
QueryForecast API タスクの最大同時実行数	<p>大規模なデータセット (20 GB または 100,000 アイテムを超えるもの) で作成された 5 つを含む 10 個の予測。</p> <p>大規模なデータセットで作成された 6 個以上の予測がある場合、QueryForecast がアクセスできるのは最新の 5 つの大規模なデータセットの予測のみです。</p>	いいえ



リソース	デフォルトのクォータ	調整可能
説明可能性の最大数	1,000	いいえ
説明可能性エクスポートジョブの最大数	1,000	いいえ

### What-if 分析のクォータ

リソース	デフォルトのクォータ	調整可能
CreateWhatIfAnalysis タスクの最大同時実行数	3	はい
What-If 分析の最大数	500	はい
CreateWhatIfForecast タスクの最大同時実行数	3	はい
What-If 予想の最大数	100	はい
CreateWhatIfForecastExport タスクの最大同時実行数	3	はい
What-If 予想エクスポートの最大数	1,000	はい
エクスポートジョブの What-If 予想エクスポートの最大数	3	いいえ

## 条件と制限

Weather Index を使用する場合は、次の条件と制約事項が適用されます。

- 使用可能なアルゴリズム: レガシー予測子を使用している場合、CNN-QR、DeepAR+、および Prophet アルゴリズムを使用して予測子をトレーニングすると、Weather Index を有効にできます。Weather Index は、ARIMA、ETS、および NPTS には適用されません。

- 予想頻度: 有効な予想頻度は、Minutely、Hourly、および Daily です。
- [Forecast horizon] (予測期間): 予測期間は、今後 14 日間を超えて設定することはできません。各予測頻度の予測期間の制限については、以下のリストを参照してください。
  - 1 minute - 500
  - 5 minutes - 500
  - 10 minutes - 500
  - 15 minutes - 500
  - Hourly - 330
  - Daily - 14
- 時系列の長さ: Weather Index を使用してモデルをトレーニングする場合、Forecast は、Forecast 気象データセットの特徴量化の開始日前のタイムスタンプを持つすべての時系列データセットを切り捨てます。Forecast weather データセットの機能には、次の開始日が含まれています。
  - 米国リージョン: 2018 年 7 月 2 日
  - 欧州リージョン: 2018 年 7 月 2 日
  - アジアパシフィックリージョン: 2018 年 7 月 2 日
  - カナダリージョン: 2019 年 7 月 2 日
  - 南米リージョン: 2020 年 1 月 2 日
  - 中米リージョン: 2020 年 9 月 2 日
  - アフリカおよび中東リージョン: 2021 年 3 月 25 日

Weather Index を有効にすると、開始日の前のタイムスタンプを持つデータポイントは、予測子のトレーニング中に使用されません。

- 場所の数: ターゲット時系列データセットは、2000 の一意の場所を超えることはできません。
- リージョンの境界: データセット内のすべてのアイテムは 1 つのリージョン内にある必要があります。
- [Minimum time series length] (最小時系列長): Weather Index をテストする際の追加のデータ要件により、時系列データセットの最小長は次のとおりです:

$$3 \times \text{ForecastHorizon} + (\text{BacktestWindows} + 1) \times \text{BacktestWindowOffset}$$

時系列データセットがこの要件を満たしていない場合は、以下を減らすことを検討してください。

- ForecastHorizon - 予測期間を短縮します。

- BacktestWindowOffset - バックテスト中にテストセットの長さを短くします。

- `BacktestWindows` - バックテストの数を減らします。

## 予約フィールド名

Amazon Forecast では次の名前が予約されています。スキーマフィールドまたはデータセットヘッダーにこれらの名前を使用することはできません。

### A

- A
- ABORT
- ABS
- ABSOLUTE
- ACCESS
- ACTION
- ADA
- ADD
- ADMIN
- AFTER
- AGGREGATE
- ALIAS
- ALL
- ALLOCATE
- ALSO
- ALTER
- ALWAYS
- ANALYSE
- ANALYZE
- AND
- ANY
- ARE
- ARRAY
- AS

- ASC
- ASENSITIVE
- ASSERTION
- ASSIGNMENT
- ASYMMETRIC
- AT
- ATOMIC
- ATTRIBUTE
- ATTRIBUTES
- AUDIT
- AUTHORIZATION
- AUTO\_INCREMENT
- AVG
- AVG\_ROW\_LENGTH

## B

- BACKUP
- BACKWARD
- BEFORE
- BEGIN
- BERNOULLI
- BETWEEN
- BIGINT
- BINARY
- BIT
- BIT\_LENGTH
- BITVAR
- BLOB
- BOOL
- BOOLEAN

- BOTH
- BREADTH
- BREAK
- BROWSE
- BULK
- BY

## C

- C
- CACHE
- CALL
- CALLED
- CARDINALITY
- CASCADE
- CASCADED
- CASE
- CAST
- CATALOG
- CATALOG\_NAME
- CEIL
- CEILING
- CHAIN
- CHANGE
- CHAR
- CHAR\_LENGTH
- CHARACTER
- CHARACTER\_LENGTH
- CHARACTER\_SET\_CATALOG
- CHARACTER\_SET\_NAME
- CHARACTER\_SET\_SCHEMA

- CHARACTERISTICS
- CHARACTERS
- CHECK
- CHECKED
- CHECKPOINT
- CHECKSUM
- CLASS
- CLASS\_ORIGIN
- CLOB
- CLOSE
- CLUSTER
- CLUSTERED
- COALESCE
- COBOL
- COLLATE
- COLLATION
- COLLATION\_CATALOG
- COLLATION\_NAME
- COLLATION\_SCHEMA
- COLLECT
- COLUMN
- COLUMN\_NAME
- COLUMNS
- COMMAND\_FUNCTION
- COMMAND\_FUNCTION\_CODE
- COMMENT
- COMMIT
- COMMITTED
- COMPLETION
- COMPRESS

- COMPUTE
- CONDITION
- CONDITION\_NUMBER
- CONNECT
- CONNECTION
- CONNECTION\_NAME
- CONSTRAINT
- CONSTRAINT\_CATALOG
- CONSTRAINT\_NAME
- CONSTRAINT\_SCHEMA
- CONSTRAINTS
- CONSTRUCTOR
- CONTAINS
- CONTAINSTABLE
- CONTINUE
- CONVERSION
- CONVERT
- COPY
- CORR
- CORRESPONDING
- COUNT
- COVAR\_POP
- COVAR\_SAMP
- CREATE
- CREATEDB
- CREATEROLE
- CREATEUSER
- CROSS
- CSV
- CUBE



- CUME\_DIST
- CURRENT
- CURRENT\_DATE
- CURRENT\_DEFAULT\_TRANSFORM\_GROUP
- CURRENT\_PATH
- CURRENT\_ROLE
- CURRENT\_TIME
- CURRENT\_TIMESTAMP
- CURRENT\_TRANSFORM\_GROUP\_FOR\_TYPE
- CURRENT\_USER
- CURSOR
- CURSOR\_NAME
- CYCLE

## D

- DATA
- DATABASE
- DATABASES
- DATETIME
- DATETIME\_INTERVAL\_CODE
- DATETIME\_INTERVAL\_PRECISION
- DAY
- DAY\_HOUR
- DAY\_MICROSECOND
- DAY\_MINUTE
- DAY\_SECOND
- DAYOFMONTH
- DAYOFWEEK
- DAYOFYEAR
- DBCC

- DEALLOCATE
- DEC
- DECIMAL
- DECLARE
- DEFAULT
- DEFAULTS
- DEFERRABLE
- DEFERRED
- DEFINED
- DEFINER
- DEGREE
- DELAY\_KEY\_WRITE
- DELAYED
- DELETE
- DELIMITER
- DELIMITERS
- DENSE\_RANK
- DENY
- DEPTH
- Deref
- DERIVED
- DESC
- DESCRIBE
- DESCRIPTOR
- DESTROY
- DESTRUCTOR
- DETERMINISTIC
- DIAGNOSTICS
- DICTIONARY
- DISABLE

- DISCONNECT
- DISK
- DISPATCH
- DISTINCT
- DISTINCTROW
- DISTRIBUTED
- DIV
- DO
- DOMAIN
- DOUBLE
- DROP
- DUAL
- DUMMY
- DUMP
- DYNAMIC
- DYNAMIC\_FUNCTION
- DYNAMIC\_FUNCTION\_CODE

## E

- EACH
- ELEMENT
- ELSE
- ELSEIF
- ENABLE
- ENCLOSED
- ENCODING
- ENCRYPTED
- END
- END-EXEC
- ENUM

- EQUALS
- ERRLVL
- ESCAPE
- ESCAPED
- EVERY
- EXCEPT
- EXCEPTION
- EXCLUDE
- EXCLUDING
- EXCLUSIVE
- EXEC
- EXECUTE
- EXISTING
- EXISTS
- EXIT
- EXP
- EXPLAIN
- EXTERNAL
- EXTRACT

## F

- FALSE
- FETCH
- FIELDS
- FILE
- FILLFACTOR
- FILTER
- FINAL
- FIRST
- FLOAT

- FLOAT4
- FLOAT8
- FLOOR
- FLUSH
- FOLLOWING
- FOR
- FORCE
- FOREIGN
- FORTRAN
- FORWARD
- FOUND
- FREE
- FREETEXT
- FREETEXTTABLE
- FREEZE
- FROM
- FULL
- FULLTEXT
- FUNCTION
- FUSION

## G

- G
- GENERAL
- GENERATED
- GET
- GLOBAL
- GO
- GOTO
- GRANT

- GRANTED
- GRANTS
- GREATEST
- GROUP
- GROUPING

## H

- HANDLER
- HAVING
- HEADER
- HEAP
- HIERARCHY
- HIGH\_PRIORITY
- HOLD
- HOLDLOCK
- HOST
- HOSTS
- HOUR
- HOUR\_MICROSECOND
- HOUR\_MINUTE
- HOUR\_SECOND

## I

- IDENTIFIED
- IDENTITY
- IDENTITY\_INSERT
- IDENTITYCOL
- IF
- IGNORE
- ILIKE

- IMMEDIATE
- IMMUTABLE
- IMPLEMENTATION
- IMPLICIT
- IN
- INCLUDE
- INCLUDING
- INCREMENT
- INDEX
- INDICATOR
- INFILE
- INFIX
- INHERIT
- INHERITS
- INITIAL
- INITIALIZE
- INITIALLY
- INNER
- INOUT
- INPUT
- INSENSITIVE
- INSERT
- INSERT\_ID
- INSTANCE
- INSTANTIABLE
- INSTEAD
- INT
- INT1
- INT2
- INT3

- INT4
- INT8
- INTEGER
- INTERSECT
- INTERSECTION
- INTERVAL
- INTO
- INVOKER
- IS
- ISAM
- ISNULL
- ISOLATION
- ITERATE

## J

- JOIN

## K

- K
- KEY
- KEY\_MEMBER
- KEY\_TYPE
- KEYS
- KILL

## L

- LANCOMPILER
- LANGUAGE
- LARGE



- LAST
- LAST\_INSERT\_ID
- LATERAL
- LEADING
- LEAST
- LEAVE
- LEFT
- LENGTH
- LESS
- LEVEL
- LIKE
- LIMIT
- LINENO
- LINES
- LISTEN
- LN
- LOAD
- LOCAL
- LOCALTIME
- LOCALTIMESTAMP
- LOCATOR
- LOCK
- LOGIN
- LOGS
- LONG
- LONGBLOB
- LONGTEXT
- LOOP
- LOW\_PRIORITY
- LOWER

## M

- M
- MAP
- MATCH
- MATCHED
- MAX
- MAX\_ROWS
- MAXEXTENTS
- MAXVALUE
- MEAN
- MEDIUMBLOB
- MEDIUMINT
- MEDIUMTEXT
- MEMBER
- MERGE
- MESSAGE\_LENGTH
- MESSAGE\_OCTET\_LENGTH
- MESSAGE\_TEXT
- METHOD
- MIDDLEINT
- MIN
- MIN\_ROWS
- MINUS
- MINUTE
- MINUTE\_MICROSECOND
- MINUTE\_SECOND
- MINVALUE
- MLSLABEL
- MOD
- MODE

- MODIFIES
- MODIFY
- MODULE
- MONTH
- MONTHNAME
- MORE
- MOVE
- MULTISSET
- MUMPS
- MYISAM

## N

- NAME
- NAMES
- NATIONAL
- NATURAL
- NCHAR
- NCLOB
- NESTING
- NEW
- NEXT
- NO
- NO\_WRITE\_TO\_BINLOG
- NOAUDIT
- NOCHECK
- NOCOMPRESS
- NOCREATEDB
- NOCREATEROLE
- NOCREATEUSER
- NOINHERIT

- NOLOGIN
- NONCLUSTERED
- NONE
- NORMALIZE
- NORMALIZED
- NOSUPERUSER
- NOT
- NOTHING
- NOTIFY
- NOTNULL
- NOWAIT
- NULL
- NULLABLE
- NULLIF
- NULLS
- NUMBER
- NUMERIC

## O

- OBJECT
- OCTET\_LENGTH
- OCTETS
- OF
- OFF
- OFFLINE
- OFFSET
- OFFSETS
- OIDS
- OLD
- ON

- ONLINE
- ONLY
- OPEN
- OPENDATASOURCE
- OPENQUERY
- OPENROWSET
- OPENXML
- OPERATION
- OPERATOR
- OPTIMIZE
- OPTION
- OPTIONALLY
- OPTIONS
- OR
- ORDER
- ORDERING
- ORDINALITY
- OTHERS
- OUT
- OUTER
- OUTFILE
- OUTPUT
- OVER
- OVERLAPS
- OVERLAY
- OVERRIDING
- OWNER

## P

- PACK\_KEYS

- PAD
- PARAMETER
- PARAMETER\_MODE
- PARAMETER\_NAME
- PARAMETER\_ORDINAL\_POSITION
- PARAMETER\_SPECIFIC\_CATALOG
- PARAMETER\_SPECIFIC\_NAME
- PARAMETER\_SPECIFIC\_SCHEMA
- PARAMETERS
- PARTIAL
- PARTITION
- PASCAL
- PASSWORD
- PATH
- PCTFREE
- PERCENT
- PERCENT\_RANK
- PERCENTILE\_CONT
- PERCENTILE\_DISC
- PLACING
- PLAN
- PLI
- POSITION
- POSTFIX
- POWER
- PRECEDING
- PRECISION
- PREFIX
- PREORDER
- PREPARE

- PREPARED
- PRESERVE
- PRIMARY
- PRINT
- PRIOR
- PRIVILEGES
- PROC
- PROCEDURAL
- PROCEDURE
- PROCESS
- PROCESSLIST
- PUBLIC
- PURGE

## Q

- QUOTE

## R

- RAID0
- RAISERROR
- RANGE
- RANK
- RAW
- READ
- READS
- READTEXT
- REAL
- RECHECK
- RECONFIGURE
- RECURSIVE

- REF
- REFERENCES
- REFERENCING
- REGEXP
- REGR\_AVGX
- REGR\_AVGY
- REGR\_COUNT
- REGR\_INTERCEPT
- REGR\_R2
- REGR\_SLOPE
- REGR\_SXX
- REGR\_SXY
- REGR\_SYY
- REINDEX
- RELATIVE
- RELEASE
- RELOAD
- RENAME
- REPEAT
- REPEATABLE
- REPLACE
- REPLICATION
- REQUIRE
- RESET
- RESIGNAL
- RESOURCE
- RESTART
- RESTORE
- RESTRICT
- RESULT



- RETURN
- RETURNED\_CARDINALITY
- RETURNED\_LENGTH
- RETURNED\_OCTET\_LENGTH
- RETURNED\_SQLSTATE
- RETURNS
- REVOKE
- RIGHT
- RLIKE
- ROLE
- ROLLBACK
- ROLLUP
- ROUTINE
- ROUTINE\_CATALOG
- ROUTINE\_NAME
- ROUTINE\_SCHEMA
- ROW
- ROW\_COUNT
- ROW\_NUMBER
- ROWCOUNT
- ROWGUIDCOL
- ROWID
- ROWNUM
- ROWS
- RULE

## S

- SAVE
- SAVEPOINT
- SCALE

- SCHEMA
- SCHEMA\_NAME
- SCHEMAS
- SCOPE
- SCOPE\_CATALOG
- SCOPE\_NAME
- SCOPE\_SCHEMA
- SCROLL
- SEARCH
- SECOND
- SECOND\_MICROSECOND
- SECTION
- SECURITY
- SELECT
- SELF
- SENSITIVE
- SEPARATOR
- SEQUENCE
- SERIALIZABLE
- SERVER\_NAME
- SESSION
- SESSION\_USER
- SET
- SETOF
- SETS
- SETUSER
- SHARE
- SHOW
- SHUTDOWN
- SIGNAL

- SIMILAR
- SIMPLE
- SIZE
- SMALLINT
- SOME
- SONAME
- SOURCE
- SPACE
- SPATIAL
- SPECIFIC
- SPECIFIC\_NAME
- SPECIFICTYPE
- SQL
- SQL\_BIG\_RESULT
- SQL\_BIG\_SELECTS
- SQL\_BIG\_TABLES
- SQL\_CALC\_FOUND\_ROWS
- SQL\_LOG\_OFF
- SQL\_LOG\_UPDATE
- SQL\_LOW\_PRIORITY\_UPDATES
- SQL\_SELECT\_LIMIT
- SQL\_SMALL\_RESULT
- SQL\_WARNINGS
- SQLCA
- SQLCODE
- SQLERROR
- SQLEXCEPTION
- SQLSTATE
- SQLWARNING
- SQRT

- SSL
- STABLE
- START
- STARTING
- STATE
- STATEMENT
- STATIC
- STATISTICS
- STATUS
- STDDEV\_POP
- STDDEV\_SAMP
- STDIN
- STDOUT
- STORAGE
- STRAIGHT\_JOIN
- STRICT
- STRING
- STRUCTURE
- STYLE
- SUBCLASS\_ORIGIN
- SUBLIST
- SUBMULTISET
- SUBSTRING
- SUCCESSFUL
- SUM
- SUPERUSER
- SYMMETRIC
- SYNONYM
- SYSDATE
- SYSID

- SYSTEM
- SYSTEM\_USER

## T

- TABLE
- TABLE\_NAME
- TABLES
- TABLESAMPLE
- TABLESPACE
- TEMP
- TEMPLATE
- TEMPORARY
- TERMINATE
- TERMINATED
- TEXT
- TEXTSIZE
- THAN
- THEN
- TIES
- TIME
- TIMEZONE\_HOUR
- TIMEZONE\_MINUTE
- TINYBLOB
- TINYINT
- TINYTEXT
- TO
- TOAST
- TOP
- TOP\_LEVEL\_COUNT
- TRAILING

- TRAN
- TRANSACTION
- TRANSACTION\_ACTIVE
- TRANSACTIONS\_COMMITTED
- TRANSACTIONS\_ROLLED\_BACK
- TRANSFORM
- TRANSFORMS
- TRANSLATE
- TRANSLATION
- TREAT
- TRIGGER
- TRIGGER\_CATALOG
- TRIGGER\_NAME
- TRIGGER\_SCHEMA
- TRIM
- TRUE
- TRUNCATE
- TRUSTED
- TSEQUAL
- TYPE

## U

- UESCAPE
- UID
- UNBOUNDED
- UNCOMMITTED
- UNDER
- UNDO
- UNENCRYPTED
- UNION

- UNIQUE
- UNKNOWN
- UNLISTEN
- UNLOCK
- UNNAMED
- UNNEST
- UNSIGNED
- UNTIL
- UPDATE
- UPDATETEXT
- UPPER
- USAGE
- USE
- USER
- USER\_DEFINED\_TYPE\_CATALOG
- USER\_DEFINED\_TYPE\_CODE
- USER\_DEFINED\_TYPE\_NAME
- USER\_DEFINED\_TYPE\_SCHEMA
- USING
- UTC\_DATE
- UTC\_TIME
- UTC\_TIMESTAMP

## V

- VACUUM
- VALID
- VALIDATE
- VALIDATOR
- VALUE
- VALUES

- VAR\_POP
- VAR\_SAMP
- VARBINARY
- VARCHAR
- VARCHAR2
- VARCHARACTER
- VARIABLE
- VARIABLES
- VARYING
- VERBOSE
- VIEW
- VOLATILE

## W

- WAITFOR
- WHEN
- WHENEVER
- WHERE
- WHILE
- WIDTH\_BUCKET
- WINDOW
- WITH
- WITHIN
- WITHOUT
- WORK
- WRITE
- WRITETEXT

## X

- X509



- XOR

## Y

- YEAR
- YEAR\_MONTH

## Z

- ZEROFILL
- ZONE

# AWS SDKsコード例

次のコード例は、AWS Software Development Kit (SDK) で Forecast を使用方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、関連するシナリオやサービス間の例ではアクションのコンテキストが確認できます。

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してください [AWS SDK での Forecast の使用](#)。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

## コードの例

- [AWS SDKsアクション](#)
  - [AWS SDK または CLI CreateDatasetで を使用する](#)
  - [AWS SDK または CLI CreateForecastで を使用する](#)
  - [AWS SDK または CLI DeleteDatasetで を使用する](#)
  - [AWS SDK または CLI DeleteForecastで を使用する](#)
  - [AWS SDK または CLI DescribeForecastで を使用する](#)
  - [AWS SDK または CLI ListDatasetGroupsで を使用する](#)
  - [AWS SDK または CLI ListForecastsで を使用する](#)

## AWS SDKsアクション

次のコード例は、AWS SDKs を使用して個々の Forecast アクションを実行する方法を示しています。これらの抜粋は Forecast API を呼び出し、コンテキスト内で実行する必要がある大規模なプログラムからのコードの抜粋です。各例には へのリンクが含まれており GitHub、コードの設定と実行の手順を確認できます。

以下の例には、最も一般的に使用されるアクションのみ含まれています。詳細なリストについては、「[Amazon Forecast API リファレンス](#)」を参照してください。

### 例

- [AWS SDK または CLI CreateDatasetで を使用する](#)
- [AWS SDK または CLI CreateForecastで を使用する](#)

- [AWS SDK または CLI DeleteDatasetで を使用する](#)
- [AWS SDK または CLI DeleteForecastで を使用する](#)
- [AWS SDK または CLI DescribeForecastで を使用する](#)
- [AWS SDK または CLI ListDatasetGroupsで を使用する](#)
- [AWS SDK または CLI ListForecastsで を使用する](#)

## AWS SDK または CLI **CreateDataset**で を使用する

次の例は、CreateDataset を使用する方法を説明しています。

### Java

#### SDK for Java 2.x

#### Note

については、「」を参照してください GitHub。 [AWS コード例リポジトリ](#) で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.forecast.ForecastClient;
import software.amazon.awssdk.services.forecast.model.CreateDatasetRequest;
import software.amazon.awssdk.services.forecast.model.Schema;
import software.amazon.awssdk.services.forecast.model.SchemaAttribute;
import software.amazon.awssdk.services.forecast.model.CreateDatasetResponse;
import software.amazon.awssdk.services.forecast.model.ForecastException;
import java.util.ArrayList;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class CreateDataSet {
```

```
public static void main(String[] args) {
    final String usage = ""

        Usage:
        <name>\s

        Where:
        name - The name of the data set.\s
        """;

    if (args.length != 1) {
        System.out.println(usage);
        System.exit(1);
    }

    String name = args[0];
    Region region = Region.US_WEST_2;
    ForecastClient forecast = ForecastClient.builder()
        .region(region)
        .build();

    String myDataSetARN = createForecastDataSet(forecast, name);
    System.out.println("The ARN of the new data set is " + myDataSetARN);
    forecast.close();
}

public static String createForecastDataSet(ForecastClient forecast, String
name) {
    try {
        Schema schema = Schema.builder()
            .attributes(getSchema())
            .build();

        CreateDatasetRequest datasetRequest = CreateDatasetRequest.builder()
            .datasetName(name)
            .domain("CUSTOM")
            .datasetType("RELATED_TIME_SERIES")
            .dataFrequency("D")
            .schema(schema)
            .build();

        CreateDatasetResponse response =
forecast.createDataset(datasetRequest);
        return response.datasetArn();
    }
}
```

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

    return "";
}

// Create a SchemaAttribute list required to create a data set.
private static List<SchemaAttribute> getSchema() {

    List<SchemaAttribute> schemaList = new ArrayList<>();
    SchemaAttribute att1 = SchemaAttribute.builder()
        .attributeName("item_id")
        .attributeType("string")
        .build();

    SchemaAttribute att2 = SchemaAttribute.builder()
        .attributeName("timestamp")
        .attributeType("timestamp")
        .build();

    SchemaAttribute att3 = SchemaAttribute.builder()
        .attributeName("target_value")
        .attributeType("float")
        .build();

    // Push the SchemaAttribute objects to the List.
    schemaList.add(att1);
    schemaList.add(att2);
    schemaList.add(att3);
    return schemaList;
}
}
```

- API の詳細については、「API リファレンス [CreateDataset](#)」の「」を参照してください。  
AWS SDK for Java 2.x

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してください [AWS SDK での Forecast の使用](#)。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

## AWS SDK または CLI `CreateForecast` で使用する

次の例は、`CreateForecast` を使用する方法を説明しています。

Java

SDK for Java 2.x

### Note

については、「」を参照してください GitHub。 [AWS コード例リポジトリ](#) で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.forecast.ForecastClient;
import software.amazon.awssdk.services.forecast.model.CreateForecastRequest;
import software.amazon.awssdk.services.forecast.model.CreateForecastResponse;
import software.amazon.awssdk.services.forecast.model.ForecastException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class CreateForecast {
    public static void main(String[] args) {
        final String usage = ""

                Usage:
                <name> <predictorArn>\s

                Where:
                name - The name of the forecast.\s
    }
}
```

```
        predictorArn - The arn of the predictor to use.\s

        """;

    if (args.length != 2) {
        System.out.println(usage);
        System.exit(1);
    }

    String name = args[0];
    String predictorArn = args[1];
    Region region = Region.US_WEST_2;
    ForecastClient forecast = ForecastClient.builder()
        .region(region)
        .build();

    String forecastArn = createNewForecast(forecast, name, predictorArn);
    System.out.println("The ARN of the new forecast is " + forecastArn);
    forecast.close();
}

public static String createNewForecast(ForecastClient forecast, String name,
String predictorArn) {
    try {
        CreateForecastRequest forecastRequest =
CreateForecastRequest.builder()
            .forecastName(name)
            .predictorArn(predictorArn)
            .build();

        CreateForecastResponse response =
forecast.createForecast(forecastRequest);
        return response.forecastArn();

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

- APIの詳細については、「API リファレンス [CreateForecast](#)」の「」を参照してください。  
AWS SDK for Java 2.x

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してください [AWS SDK での Forecast の使用](#)。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

## AWS SDK または CLI `DeleteDataset` で を使用する

次の例は、`DeleteDataset` を使用する方法を説明しています。

### Java

#### SDK for Java 2.x

#### Note

については、「」を参照してください [GitHub](#)。 [AWS コード例リポジトリ](#) で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.forecast.ForecastClient;
import software.amazon.awssdk.services.forecast.model.DeleteDatasetRequest;
import software.amazon.awssdk.services.forecast.model.ForecastException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class DeleteDataset {

    public static void main(String[] args) {
        final String usage = ""

        Usage:
```



```
        <datasetARN>\s

        Where:
            datasetARN - The ARN of the data set to delete.\s
        """;

    if (args.length != 1) {
        System.out.println(usage);
        System.exit(1);
    }

    String datasetARN = args[0];
    Region region = Region.US_WEST_2;
    ForecastClient forecast = ForecastClient.builder()
        .region(region)
        .build();

    deleteForecastDataSet(forecast, datasetARN);
    forecast.close();
}

public static void deleteForecastDataSet(ForecastClient forecast, String
myDataSetARN) {
    try {
        DeleteDatasetRequest deleteRequest = DeleteDatasetRequest.builder()
            .datasetArn(myDataSetARN)
            .build();

        forecast.deleteDataset(deleteRequest);
        System.out.println("The Data Set was deleted");

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

- APIの詳細については、「APIリファレンス[DeleteDataset](#)」の「」を参照してください。  
AWS SDK for Java 2.x

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してください [AWS SDK での Forecast の使用](#)。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

## AWS SDK または CLI `DeleteForecast` を使用する

次の例は、`DeleteForecast` を使用方法を説明しています。

Java

SDK for Java 2.x

### Note

については、「」を参照してください [GitHub](#)。 [AWS コード例リポジトリ](#) で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.forecast.ForecastClient;
import software.amazon.awssdk.services.forecast.model.DeleteDatasetRequest;
import software.amazon.awssdk.services.forecast.model.ForecastException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class DeleteDataset {

    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <datasetARN>\s

            Where:
                datasetARN - The ARN of the data set to delete.\s
    }
```

```
        """);

    if (args.length != 1) {
        System.out.println(usage);
        System.exit(1);
    }

    String datasetARN = args[0];
    Region region = Region.US_WEST_2;
    ForecastClient forecast = ForecastClient.builder()
        .region(region)
        .build();

    deleteForecastDataSet(forecast, datasetARN);
    forecast.close();
}

public static void deleteForecastDataSet(ForecastClient forecast, String
myDataSetARN) {
    try {
        DeleteDatasetRequest deleteRequest = DeleteDatasetRequest.builder()
            .datasetArn(myDataSetARN)
            .build();

        forecast.deleteDataset(deleteRequest);
        System.out.println("The Data Set was deleted");

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

- APIの詳細については、「APIリファレンス[DeleteForecast](#)」の「」を参照してください。  
AWS SDK for Java 2.x

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してください[AWS SDK での Forecast の使用](#)。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

## AWS SDK または CLI `DescribeForecast`で を使用する

次の例は、`DescribeForecast` を使用する方法を説明しています。

### Java

#### SDK for Java 2.x

#### Note

については、「」を参照してください GitHub。 [AWS コード例リポジトリ](#) で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.forecast.ForecastClient;
import software.amazon.awssdk.services.forecast.model.DescribeForecastRequest;
import software.amazon.awssdk.services.forecast.model.DescribeForecastResponse;
import software.amazon.awssdk.services.forecast.model.ForecastException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class DescribeForecast {
    public static void main(String[] args) {
        final String usage = ""

                Usage:
                <forecastarn>\s

                Where:
                forecastarn - The arn of the forecast (for example,
                "arn:aws:forecast:us-west-2:xxxxx322:forecast/my_forecast)
                """;

        if (args.length != 1) {
```

```
        System.out.println(usage);
        System.exit(1);
    }

    String forecastarn = args[0];
    Region region = Region.US_WEST_2;
    ForecastClient forecast = ForecastClient.builder()
        .region(region)
        .build();

    describe(forecast, forecastarn);
    forecast.close();
}

public static void describe(ForecastClient forecast, String forecastarn) {
    try {
        DescribeForecastRequest request = DescribeForecastRequest.builder()
            .forecastArn(forecastarn)
            .build();

        DescribeForecastResponse response =
forecast.describeForecast(request);
        System.out.println("The name of the forecast is " +
response.forecastName());

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

- API の詳細については、「API リファレンス[DescribeForecast](#)」の「」を参照してください。AWS SDK for Java 2.x

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してください[AWS SDK での Forecast の使用](#)。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

## AWS SDK または CLI `ListDatasetGroups` を使用する

次の例は、`ListDatasetGroups` を使用方法を説明しています。

### Java

#### SDK for Java 2.x

#### Note

については、「」を参照してください GitHub。 [AWS コード例リポジトリ](#) で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.forecast.ForecastClient;
import software.amazon.awssdk.services.forecast.model.DatasetGroupSummary;
import software.amazon.awssdk.services.forecast.model.ListDatasetGroupsRequest;
import software.amazon.awssdk.services.forecast.model.ListDatasetGroupsResponse;
import software.amazon.awssdk.services.forecast.model.ForecastException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class ListDataSetGroups {
    public static void main(String[] args) {
        Region region = Region.US_WEST_2;
        ForecastClient forecast = ForecastClient.builder()
            .region(region)
            .build();

        listDataGroups(forecast);
        forecast.close();
    }
}
```

```
public static void listDataGroups(ForecastClient forecast) {
    try {
        ListDatasetGroupsRequest group = ListDatasetGroupsRequest.builder()
            .maxResults(10)
            .build();

        ListDatasetGroupsResponse response =
forecast.listDatasetGroups(group);
        List<DatasetGroupSummary> groups = response.datasetGroups();
        for (DatasetGroupSummary myGroup : groups) {
            System.out.println("The Data Set name is " +
myGroup.datasetGroupName());
        }

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

- APIの詳細については、「API リファレンス」の[ListDataset「グループ」](#)を参照してください。AWS SDK for Java 2.x

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してください[AWS SDK での Forecast の使用](#)。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

## AWS SDK または CLI **ListForecasts**で を使用する

次の例は、ListForecasts を使用する方法を説明しています。

Java

SDK for Java 2.x

### Note

については、「」を参照してください GitHub。 [AWS コード例リポジトリ](#) で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.forecast.ForecastClient;
import software.amazon.awssdk.services.forecast.model.ListForecastsResponse;
import software.amazon.awssdk.services.forecast.model.ListForecastsRequest;
import software.amazon.awssdk.services.forecast.model.ForecastSummary;
import software.amazon.awssdk.services.forecast.model.ForecastException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class ListForecasts {

    public static void main(String[] args) {
        Region region = Region.US_WEST_2;
        ForecastClient forecast = ForecastClient.builder()
            .region(region)
            .build();

        listAllForecasts(forecast);
        forecast.close();
    }

    public static void listAllForecasts(ForecastClient forecast) {
        try {
            ListForecastsRequest request = ListForecastsRequest.builder()
                .maxResults(10)
                .build();

            ListForecastsResponse response = forecast.listForecasts(request);
            List<ForecastSummary> forecasts = response.forecasts();
            for (ForecastSummary forecastSummary : forecasts) {
                System.out.println("The name of the forecast is " +
                    forecastSummary.forecastName());
            }
        } catch (ForecastException e) {
```



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

- API の詳細については、「API リファレンス[ListForecasts](#)」の「」を参照してください。  
AWS SDK for Java 2.x

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してください[AWS SDK での Forecast の使用](#)。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

# Amazon Forecast のセキュリティ

のクラウドセキュリティが最優先事項 AWS です。お客様は AWS、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャからメリットを得られます。

セキュリティは、AWS とユーザーの間で共有される責任です。[責任共有モデル](#)では、これをクラウドのセキュリティおよびクラウド内のセキュリティと説明しています。

- **クラウドのセキュリティ** — クラウドで AWS サービスを実行するインフラストラクチャを保護する責任 AWS は にあります AWS。AWS また、では、安全に使用できるサービスも提供しています。コンプライアンス [AWS プログラム](#) コンプライアンスプログラム の一環として、サードパーティーの監査者は定期的にセキュリティの有効性をテストおよび検証。Amazon Forecast に適用するコンプライアンスプログラムの詳細については、「[コンプライアンスプログラムによる対象範囲内のAWS のサービス](#)」「」を参照してください。
- **クラウドのセキュリティ** — お客様の責任は、使用する AWS サービスによって決まります。また、お客様は、データの機密性、会社の要件、適用される法律や規制など、その他の要因についても責任を負います。

このドキュメントは、Forecast を使用する際に共有責任モデルを適用する方法を理解するのに役立ちます。以下のトピックでは、セキュリティおよびコンプライアンスの目的を達成するために Forecast を設定する方法を示します。また、Forecast リソースのモニタリングや保護 AWS のサービス に役立つ他の の使用方法についても説明します。

## トピック

- [Amazon Forecast におけるデータ保護](#)
- [Amazon Forecast のための Identity and Access Management](#)
- [Amazon Forecast でのログ記録とモニタリング](#)
- [Amazon Forecast のコンプライアンス検証](#)
- [Amazon Forecast の回復力](#)
- [Amazon Forecast のインフラストラクチャセキュリティ](#)
- [Forecast とインターフェイス VPC エンドポイント \(AWS PrivateLink\)](#)

## Amazon Forecast におけるデータ保護

責任 AWS [共有モデル](#)、Amazon Forecast でのデータ保護に適用されます。このモデルで説明されているように、AWS はすべての を実行するグローバルインフラストラクチャを保護する責任があります AWS クラウド。お客様は、このインフラストラクチャでホストされているコンテンツに対する管理を維持する責任があります。また、使用する AWS のサービスのセキュリティ設定と管理タスクもユーザーの責任となります。データプライバシーの詳細については、「[データプライバシーのよくある質問](#)」を参照してください。欧州でのデータ保護の詳細については、AWS セキュリティブログに投稿された「[AWS 責任共有モデルおよび GDPR](#)」のブログ記事を参照してください。

データ保護の目的で、認証情報を保護し AWS アカウント、AWS IAM Identity Center または AWS Identity and Access Management (IAM) を使用して個々のユーザーを設定することをお勧めします。この方法により、それぞれのジョブを遂行するために必要な権限のみが各ユーザーに付与されます。また、次の方法でデータを保護することもお勧めします:

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。TLS 1.2 は必須であり TLS 1.3 がお勧めです。
- で API とユーザーアクティビティのログ記録を設定します AWS CloudTrail。
- AWS 暗号化ソリューションと、内のすべてのデフォルトのセキュリティコントロールを使用します AWS のサービス。
- Amazon Macie などの高度なマネージドセキュリティサービスを使用します。これらは、Amazon S3 に保存されている機密データの検出と保護を支援します。
- コマンドラインインターフェイスまたは API AWS を介して にアクセスするときに FIPS 140-2 検証済みの暗号化モジュールが必要な場合は、FIPS エンドポイントを使用します。利用可能な FIPS エンドポイントの詳細については、「[連邦情報処理規格 \(FIPS\) 140-2](#)」を参照してください。

お客様の E メールアドレスなどの極秘または機密情報は、タグ、または名前フィールドなどの自由形式のテキストフィールドに配置しないことを強くお勧めします。これは、コンソール、API、または SDK を使用して Forecast AWS CLI または他の AWS のサービスを使用する場合も同様です。AWS SDKs 名前に使用する自由記述のテキストフィールドやタグに入力したデータは、課金や診断ログに使用される場合があります。外部サーバーへの URL を提供する場合は、そのサーバーへのリクエストを検証するための認証情報を URL に含めないように強くお勧めします。

## 保管時の暗号化

Amazon Forecast では、[CreateDataset](#) および [CreatePredictor](#) オペレーション中に暗号化設定が提供されます。CreateDataset オペレーションで暗号化設定が指定されている場合、CMK と保管時の暗号化用の IAM ロールが [CreateDatasetImportJob](#) オペレーションで使用されます。

例えば、CreateDataset オペレーションの EncryptionConfig ステートメント RoleArn でキーの KMS KeyArn とを指定すると、Forecast はそのロールを引き受け、キーを使用してデータセットを暗号化します。構成が指定されていない場合、Forecast は暗号化にデフォルトのサービスキーを使用します。さらに、CreatePredictor オペレーション EncryptionConfig の情報を指定する CreateForecast と、CreatePredictorExplanability、などの後続のすべてのオペレーションは CreatePredictorBacktestExportJob、同じ設定を使用して保管時の暗号化を実行します。繰り返しのようになりますが、暗号化設定を指定しない場合、Forecast はデフォルトのサービス暗号化を使用します。

Amazon S3 バケットに保存されているデータは、デフォルトの Amazon S3 キーによって暗号化されます。独自の AWS KMS キーを使用してデータを暗号化し、Forecast にこのキーへのアクセスを許可することもできます。Amazon S3 でのデータ暗号化の詳細については、「[暗号化によるデータの保護](#)」を参照してください。独自の AWS KMS キーの管理については、「AWS Key Management Service デベロッパーガイド」の「[キーの管理](#)」を参照してください。

## 転送時と処理時の暗号化

Amazon Forecast は、AWS 証明書で TLS を使用して、他の AWS サービスに送信されるデータを暗号化します。他のサービスとの通信は HTTPS 経由で AWS 行われ、Forecast エンドポイントは HTTPS 経由の安全な接続のみをサポートします。

Amazon Forecast は、アカウントからデータをコピーし、内部 AWS システムで処理します。データを処理する際に、Forecast は Forecast AWS KMS キーまたは指定した AWS KMS キーを使用してデータを暗号化します。

## Amazon Forecast が AWS KMS で許可を使用する方法

Amazon Forecast には、カスタマー管理キーを使用するための[グラント](#)が必要です。

Forecast は、[CreatePredictor](#)または [CreateDataset](#)オペレーションの中に渡される IAM EncryptionConfig ロールを使用して許可を作成します。Forecast がロールを引き受け、ユーザーに代わってグラント作成操作を行います。詳細については、[IAM ロールの設定](#)について参照してください。

ただし、カスタマーマネージドキーで暗号化された予測子を作成すると、Amazon Forecast は [CreateGrant](#) リクエストを送信してユーザーに代わって許可を作成します AWS KMS。の許可 AWS KMS は、顧客アカウントの AWS KMS キーへのアクセスを Amazon Forecast に許可するために使用されます。

Amazon Forecast では、暗号化されたデータセットアーティファクトを読み取る AWS KMS ために、カスタマーマネージドキーを使用して Decrypt リクエストを に送信できるように、グラントが必要です。Forecast は、トレーニングアーティファクトを Amazon S3 に [暗号化](#) AWS KMS するために、グラントを使用して `GenerateDataKey` リクエストを送信します。

任意のタイミングで、許可に対するアクセス権を取り消したり、カスタマーマネージドキーに対するサービスからのアクセス権を削除したりできます。これを行うと、Amazon Forecast はカスタマーマネージドキーによって暗号化されたすべてのデータにアクセスできなくなり、そのデータに依存しているオペレーションが影響を受けます。例えば、Amazon Forecast がアクセスできない暗号化された予測子に対して `CreateForecast` オペレーションを実行しようとする、オペレーションは `AccessDeniedException` エラーを返します。

## カスタマーマネージドキーを作成する

対称カスタマーマネージドキーは、AWS Management Console または AWS KMS API を使用して作成できます。対称カスタマーマネージドキーを作成するには、AWS Key Management Service デベロッパーガイドの「[対称暗号化 KMS キーの作成](#)」のステップに従います。

キーポリシーは、カスタマーマネージドキーへのアクセスを制御します。すべてのカスタマーマネージドキーには、キーポリシーが 1 つだけ必要です。このポリシーには、そのキーを使用できるユーザーとその使用方法を決定するステートメントが含まれています。カスタマーマネージドキーを作成する際に、キーポリシーを指定することができます。詳細については、AWS Key Management Service デベロッパーガイドの「[カスタマーマネージドキーへのアクセスの管理](#)」を参照してください。

Amazon Forecast リソースでカスタマーマネージドキーを使用するには、キーポリシーで次の API オペレーションを許可する必要があります。

- [kms:DescribeKey](#) — Amazon Forecast がキーを検証できるようにするカスタマーマネージドキーの詳細を提供します。
- [kms:CreateGrant](#) – カスタマーマネージドキーに許可を追加します。指定された AWS KMS キーへのアクセスを制御する権限を付与します。これにより、Amazon Forecast が必要とする [オペレーションを付与](#) するためのアクセスを許可します。このオペレーションにより、Amazon Forecast は `GenerateDataKey` を呼び出して、暗号化されたデータキーを生成して保存します。デー

タキーは暗号化にすぐには使用されないからです。また、この操作により Amazon Forecast が Decrypt の呼び出しを行い、保存されている暗号化されたデータキーを使用して暗号化されたデータにアクセスできるようになります。

- [kms:RetireGrant](#) - CreateGrant オペレーションの完了後に、オペレーション中に提供されたすべての権限を廃止します。

#### Note

Amazon Forecast は、発信者の ID に `kms:Decrypt` と `kms:GenerateDataKey` の検証を行います。発信者 `AccessDeniedException` に関連するアクセス許可がない場合、を受け取ります。キーポリシーは次のコードのようになっているはずで

```
"Effect": "Allow",
"Principal": {
  "AWS": "AWS Invoking Identity"
},
"Action": [
  "kms:Decrypt",
  "kms:GenerateDataKey"
],
"Resource": "*"
}
```

詳細については、「[IAM ポリシー](#)」を参照してください。

Amazon Forecast に追加できるポリシーステートメントの例を以下に示します。これらは最低限必要な権限ですが、IAM ポリシーを使用して追加することもできます。

```
"Statement" : [
  {"Sid" : "Allow access to principals authorized to use Amazon Forecast",
   "Effect" : "Allow",
   "Principal" : {"AWS" : "arn:aws:iam::111122223333:role/ROLE_PASSED_TO_FORECAST"},
   "Action" : [
     "kms:DescribeKey",
     "kms:CreateGrant",
     "kms:RetireGrant"
   ],
   "Resource" : "*"
  }
]
```

```

    "Condition" : {"StringEquals" : {"kms:ViaService" :
"forecast.region.amazonaws.com",
    "kms:CallerAccount" : "111122223333"
    }
  },
  {"Sid": "Allow access for key administrators",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:root"
  },
  "Action" : [
    "kms:*"
  ],
  "Resource": "arn:aws:kms:region:111122223333:key/key_ID"
}
]

```

[ポリシーのアクセス権限の指定](#)および[キーアクセスのトラブルシューティング](#)の詳細については、「AWS Key Management Service デベロッパーガイド」を参照してください。

## Amazon Forecast Service の暗号化キーを監視する

Amazon Forecast Service リソースで AWS KMS カスタマーマネージドキーを使用する場合、[AWS CloudTrail](#)または [Amazon CloudWatch Logs](#) を使用して、Forecast が に送信するリクエストを追跡できます AWS KMS。次の例は CreateGrant、カスタマーマネージドキーによって暗号化されたデータにアクセス DescribeKey するために Amazon Forecast によって呼び出される AWS KMS オペレーションをモニタリングするための RetireGrant、およびの AWS CloudTrail イベントです。

### DescribeKey

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIGDTESTANEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIGDTESTANEXAMPLE:Sampleuser01",

```

```

        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "userName": "Admin"
    },
    "webIdFederationData": {},
    "attributes": {
        "creationDate": "2022-10-05T21:16:23Z",
        "mfaAuthenticated": "false"
    }
}
},
"eventTime": "2022-10-05T21:16:23Z",
"eventSource": "kms.amazonaws.com",
"eventName": "DescribeKey",
"awsRegion": "region",
"sourceIPAddress": "172.12.34.56",
"userAgent": "ExampleDesktop/1.0 (V1; OS)",
"requestParameters": {
    "keyId":
"arn:aws:kms:region:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
},
"responseElements": null,
"requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"readOnly": true,
"resources": [
    {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN":
"arn:aws:kms:region:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"eventCategory": "Management",
"tlsDetails": {
    "tlsVersion": "TLSv1.2",
    "cipherSuite": "ECDHE-RSA-AES256-GCM-SHA384",
    "clientProvidedHostHeader": "kms.region.amazonaws.com"
}
}

```



## CreateGrant

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-10-05T23:10:27Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2022-10-05T23:10:27Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateGrant",
  "awsRegion": "region",
  "sourceIPAddress": "172.12.34.56",
  "userAgent": "ExampleDesktop/1.0 (V1; OS)",
  "requestParameters": {
    "operations": [
      "Decrypt",
      "GenerateDataKey"
    ],
    "granteePrincipal": "AWS Internal",
    "keyId":
"arn:aws:kms:region:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
  },
  "responseElements": {
    "grantId":
"0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE"
  },

```

```

"requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"readOnly": false,
"resources": [
  {
    "accountId": "111122223333",
    "type": "AWS::KMS::Key",
    "ARN":
"arn:aws:kms:region:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"eventCategory": "Management",
"tlsDetails": {
  "tlsVersion": "TLSv1.2",
  "cipherSuite": "ECDHE-RSA-AES256-GCM-SHA384",
  "clientProvidedHostHeader": "kms.region.amazonaws.com"
}
}

```

## RetireGrant

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-10-06T04:56:14Z",

```

```

        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2022-10-06T04:56:14Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "RetireGrant",
  "awsRegion": "region",
  "sourceIPAddress": "172.12.34.56",
  "userAgent": "ExampleDesktop/1.0 (V1; OS)",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData": {
    "grantId":
"0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE"
  },
  "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN":
"arn:aws:kms:region:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management",
  "tlsDetails": {
    "tlsVersion": "TLSv1.2",
    "cipherSuite": "ECDHE-RSA-AES256-GCM-SHA384",
    "clientProvidedHostHeader": "kms.region.amazonaws.com"
  }
}

```

## Amazon Forecast のための Identity and Access Management

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全に制御 AWS のサービス するのに役立つです。IAM 管理者は、誰を認証 (サインイン) し、誰に Forecast

リソースの使用を許可する (許可を持たせる) かを制御します。IAM は、追加料金なしで AWS のサービス 使用できる です。

## トピック

- [対象者](#)
- [アイデンティティを使用した認証](#)
- [ポリシーを使用したアクセスの管理](#)
- [IAM を使用する Amazon Forecast の仕組み](#)
- [Amazon Forecast のアイデンティティベースのポリシーの例](#)
- [Amazon Forecast アイデンティティとアクセスのトラブルシューティング](#)

## 対象者

AWS Identity and Access Management (IAM) の使用方法は、Forecast で行う作業によって異なります。

[Service user] (サービスユーザー) - Forecast のサービスを使用してジョブを実行する場合は、必要な許可と認証情報を管理者が用意します。作業を実行するためにさらに多くの Forecast の特徴を使用するとき、追加の許可が必要になる場合があります。アクセスの管理方法を理解しておく、管理者に適切な許可をリクエストするうえで役立ちます。Forecast の特徴にアクセスできない場合は、「[Amazon Forecast アイデンティティとアクセスのトラブルシューティング](#)」を参照してください。

[Service administrator] (サービス管理者) - 社内の Forecast リソースを担当している場合は、通常、Forecast へのフルアクセスがあります。サービスのユーザーがどの Forecast 機能やリソースにアクセスするかを決めるのは管理者の業務です。その後、IAM 管理者にリクエストを送信して、サービスユーザーの権限を変更する必要があります。このページの情報を点検して、IAM の基本概念を理解してください。貴社が Forecast で IAM を利用する方法の詳細については、「[IAM を使用する Amazon Forecast の仕組み](#)」を参照してください。

IAM 管理者 - 管理者は、Forecast へのアクセスを管理するポリシーの作成方法の詳細について確認する場合があります。IAM で使用できる Forecast アイデンティティベースのポリシーの例を表示するには、「[Amazon Forecast のアイデンティティベースのポリシーの例](#)」を参照してください。

## アイデンティティを使用した認証

認証とは、ID 認証情報 AWS を使用して にサインインする方法です。として、IAM ユーザーとして AWS アカウントのルートユーザー、または IAM ロールを引き受けて認証 ( にサインイン AWS) される必要があります。

ID ソースを介して提供された認証情報を使用して、フェデレーテッド ID AWS として にサインインできます。AWS IAM Identity Center ( IAM Identity Center) ユーザー、会社のシングルサインオン認証、Google または Facebook の認証情報は、フェデレーテッド ID の例です。フェデレーテッド ID としてサインインする場合、IAM ロールを使用して、前もって管理者により ID フェデレーションが設定されています。フェデレーション AWS を使用して にアクセスすると、間接的にロールを引き受けることになります。

ユーザーのタイプに応じて、AWS Management Console または AWS アクセスポータルにサインインできます。へのサインインの詳細については AWS、「ユーザーガイド」の「[へのサインイン AWS アカウント](#)方法AWS サインイン」を参照してください。

AWS プログラムで にアクセスする場合、 は Software Development Kit (SDK) とコマンドラインインターフェイス (CLI) AWS を提供し、認証情報を使用してリクエストに暗号で署名します。AWS ツールを使用しない場合は、リクエストに自分で署名する必要があります。推奨される方法を使用してリクエストを自分で署名する方法の詳細については、IAM [ユーザーガイドの API AWS リクエスト](#)の署名を参照してください。

使用する認証方法を問わず、追加セキュリティ情報の提供をリクエストされる場合もあります。例えば、AWS では、多要素認証 (MFA) を使用してアカウントのセキュリティを向上させることをお勧めします。詳細については、「AWS IAM Identity Center ユーザーガイド」の「[多要素認証](#)」および「IAM ユーザーガイド」の「[AWSでの多要素認証 \(MFA\) の使用](#)」を参照してください。

### AWS アカウント ルートユーザー

を作成するときは AWS アカウント、アカウント内のすべての およびリソースへの AWS のサービス完全なアクセス権を持つ1つのサインインアイデンティティから始めます。この ID は AWS アカウント ルートユーザーと呼ばれ、アカウントの作成に使用した E メールアドレスとパスワードでサインインすることでアクセスできます。日常的なタスクには、ルートユーザーを使用しないことを強くお勧めします。ルートユーザーの認証情報は保護し、ルートユーザーでしか実行できないタスクを実行するときに使用します。ルートユーザーとしてサインインする必要があるタスクの完全なリストについては、「IAM ユーザーガイド」の「[ルートユーザー認証情報が必要なタスク](#)」を参照してください。

## フェデレーティッドアイデンティティ

ベストプラクティスとして、管理者アクセスを必要とするユーザーを含む人間のユーザーに、一時的な認証情報を使用してにアクセスするための ID プロバイダーとのフェデレーションの使用を要求 AWS のサービスします。

フェデレーティッド ID は、エンタープライズユーザーディレクトリ、ウェブ ID プロバイダー、AWS Directory Service、アイデンティティセンターディレクトリのユーザー、または ID ソースを通じて提供された認証情報 AWS のサービスを使用してにアクセスするユーザーです。フェデレーティッド ID がにアクセスすると AWS アカウント、ロールを引き受け、ロールは一時的な認証情報を提供します。

アクセスを一元管理する場合は、AWS IAM Identity Centerを使用することをお勧めします。IAM Identity Center でユーザーとグループを作成することも、独自の ID ソース内のユーザーとグループのセットに接続して同期して、すべての AWS アカウント とアプリケーションで使用することもできます。IAM Identity Center の詳細については、「AWS IAM Identity Center ユーザーガイド」の「[IAM Identity Center とは](#)」を参照してください。

## IAM ユーザーとグループ

[IAM ユーザー](#)は、単一のユーザーまたはアプリケーションに対して特定のアクセス許可 AWS アカウント を持つ 内のアイデンティティです。可能であれば、パスワードやアクセスキーなどの長期的な認証情報を保有する IAM ユーザーを作成する代わりに、一時認証情報を使用することをお勧めします。ただし、IAM ユーザーでの長期的な認証情報が必要な特定のユースケースがある場合は、アクセスキーをローテーションすることをお勧めします。詳細については、IAM ユーザーガイドの「[長期的な認証情報を必要とするユースケースのためにアクセスキーを定期的にローテーションする](#)」を参照してください。

[IAM グループ](#)は、IAM ユーザーの集団を指定するアイデンティティです。グループとしてサインインすることはできません。グループを使用して、複数のユーザーに対して一度に権限を指定できます。多数のユーザーグループがある場合、グループを使用することで権限の管理が容易になります。例えば、IAMAdmins という名前のグループを設定して、そのグループに IAM リソースを管理する許可を与えることができます。

ユーザーは、ロールとは異なります。ユーザーは 1 人の人または 1 つのアプリケーションに一意に関連付けられますが、ロールはそれを必要とする任意の人が引き受けるようになっています。ユーザーには永続的な長期の認証情報がありますが、ロールでは一時的な認証情報が提供されます。詳細については、「IAM ユーザーガイド」の「[IAM ユーザー \(ロールではなく\) の作成が適している場合](#)」を参照してください。

## IAM ロール

[IAM ロール](#)は、特定のアクセス許可 AWS アカウント を持つ 内のアイデンティティです。これは IAM ユーザーに似ていますが、特定のユーザーには関連付けられていません。ロール を切り替える AWS Management Console ことで、[で IAM ロール](#)を一時的に引き受けることができます。ロール を引き受けるには、または AWS API AWS CLI オペレーションを呼び出すか、カスタム URL を使用します。ロールを使用する方法の詳細については、「IAM ユーザーガイド」の「[IAM ロールの使用](#)」を参照してください。

IAM ロールと一時的な認証情報は、次の状況で役立ちます:

- フェデレーションユーザーアクセス - フェデレーティッド ID に許可を割り当てるには、ロールを作成してそのロールの許可を定義します。フェデレーティッド ID が認証されると、その ID はロールに関連付けられ、ロールで定義されている許可が付与されます。フェデレーションの詳細については、「IAM ユーザーガイド」の「[Creating a role for a third-party Identity Provider](#)」(サードパーティーアイデンティティプロバイダー向けロールの作成)を参照してください。IAM Identity Center を使用する場合は、許可セットを設定します。アイデンティティが認証後にアクセスできるものを制御するため、IAM Identity Center は、権限セットを IAM のロールに関連付けます。アクセス許可セットの詳細については、「AWS IAM Identity Center ユーザーガイド」の「[アクセス許可セット](#)」を参照してください。
- 一時的な IAM ユーザー権限 - IAM ユーザーまたはロールは、特定のタスクに対して複数の異なる権限を一時的に IAM ロールで引き受けることができます。
- クロスアカウントアクセス - IAM ロールを使用して、自分のアカウントのリソースにアクセスすることを、別のアカウントの人物 (信頼済みプリンシパル) に許可できます。クロスアカウントアクセス権を付与する主な方法は、ロールを使用することです。ただし、一部の では AWS のサービス、(ロールをプロキシとして使用する代わりに) ポリシーをリソースに直接アタッチできます。クロスアカウントアクセスのロールとリソースベースのポリシーの違いについては、IAM ユーザーガイドの「[IAM でのクロスアカウントリソースアクセス](#)」を参照してください。
- クロスサービスアクセス — 一部の は、他の の機能 AWS のサービス を使用します AWS のサービス。例えば、あるサービスで呼び出しを行うと、通常そのサービスによって Amazon EC2 でアプリケーションが実行されたり、Amazon S3 にオブジェクトが保存されたりします。サービスでは、呼び出し元プリンシパルの許可、サービスロール、またはサービスリンクロールを使用してこれを行う場合があります。
- 転送アクセスセッション (FAS) — IAM ユーザーまたはロールを使用して でアクションを実行する場合 AWS、ユーザーはプリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、 を呼び出すプリンシパルのアクセス許可を AWS のサービス、ダウンストリー

ムサービス AWS のサービス へのリクエストリクエストリクエストと組み合わせて使用します。FAS リクエストは、サービスが他の AWS のサービス またはリソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「[転送アクセスセッション](#)」を参照してください。

- サービスロール - サービスがユーザーに代わってアクションを実行するために引き受ける [IAM ロール](#)です。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、「IAM ユーザーガイド」の「[AWS のサービスにアクセス許可を委任するロールの作成](#)」を参照してください。
- サービスにリンクされたロール - サービスにリンクされたロールは、にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント、サービスによって所有されます。IAM 管理者は、サービスリンクロールの許可を表示できますが、編集することはできません。
- Amazon EC2 で実行されているアプリケーション - IAM ロールを使用して、EC2 インスタンスで実行され、AWS CLI または AWS API リクエストを行うアプリケーションの一時的な認証情報を管理できます。これは、EC2 インスタンス内でのアクセスキーの保存に推奨されます。AWS ロールを EC2 インスタンスに割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされたインスタンスプロファイルを作成します。インスタンスプロファイルにはロールが含まれ、EC2 インスタンスで実行されるプログラムは一時的な認証情報を取得できます。詳細については、「IAM ユーザーガイド」の「[Amazon EC2 インスタンスで実行されるアプリケーションに IAM ロールを使用して許可を付与する](#)」を参照してください。

IAM ロールと IAM ユーザーのどちらを使用するかについては、「IAM ユーザーガイド」の「[\(IAM ユーザーではなく\) IAM ロールをいつ作成したら良いのか?](#)」を参照してください。

## ポリシーを使用したアクセスの管理

でアクセスを制御する AWS には、ポリシーを作成し、AWS ID またはリソースにアタッチします。ポリシーは、アイデンティティまたはリソースに関連付けられているときにアクセス許可を定義するオブジェクトです。は、プリンシパル(ユーザー、ルートユーザー、またはロールセッション) AWS がリクエストを行うときに、これらのポリシー AWS を評価します。ポリシーでの権限により、リクエストが許可されるか拒否されるかが決まります。ほとんどのポリシーは JSON ドキュメント AWS として に保存されます。JSON ポリシードキュメントの構造と内容の詳細については、「IAM ユーザーガイド」の「[JSON ポリシー概要](#)」を参照してください。



管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

デフォルトでは、ユーザーやロールに権限はありません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き継ぐことができます。

IAM ポリシーは、オペレーションの実行方法を問わず、アクションの許可を定義します。例えば、iam:GetRole アクションを許可するポリシーがあるとします。そのポリシーを持つユーザーは、AWS Management Console、AWS CLI または AWS API からロール情報を取得できます。

## アイデンティティベースのポリシー

アイデンティティベースポリシーは、IAM ユーザー、ユーザーのグループ、ロールなど、アイデンティティにアタッチできる JSON 権限ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「[IAM ポリシーの作成](#)」を参照してください。

アイデンティティベースのポリシーは、さらにインラインポリシーまたはマネージドポリシーに分類できます。インラインポリシーは、単一のユーザー、グループ、またはロールに直接埋め込まれています。管理ポリシーは、内の複数のユーザー、グループ、ロールにアタッチできるスタンドアロンポリシーです AWS アカウント。管理ポリシーには、AWS 管理ポリシーとカスタマー管理ポリシーが含まれます。マネージドポリシーまたはインラインポリシーのいずれかを選択する方法については、「IAM ユーザーガイド」の「[マネージドポリシーとインラインポリシーの比較](#)」を参照してください。

## リソースベースのポリシー

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシー や Amazon S3 バケットポリシー があげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、[プリンシパルを指定する](#)必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、またはを含めることができます AWS のサービス。

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。リソースベースのポリシーで IAM の AWS マネージドポリシーを使用することはできません。

## アクセスコントロールリスト (ACL)

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

Amazon S3、AWS WAF、および Amazon VPC は、ACLs。ACL の詳細については、『Amazon Simple Storage Service デベロッパーガイド』の「[アクセスコントロールリスト \(ACL\) の概要](#)」を参照してください。

## その他のポリシータイプ

AWS は、一般的ではない追加のポリシータイプをサポートします。これらのポリシータイプでは、より一般的なポリシータイプで付与された最大の権限を設定できます。

- **アクセス許可の境界** - アクセス許可の境界は、アイデンティティベースのポリシーによって IAM エンティティ (IAM ユーザーまたはロール) に付与できる権限の上限を設定する高度な機能です。エンティティにアクセス許可の境界を設定できます。結果として得られる権限は、エンティティのアイデンティティベースポリシーとそのアクセス許可の境界の共通部分になります。Principal フィールドでユーザーまたはロールを指定するリソースベースのポリシーでは、アクセス許可の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。アクセス許可の境界の詳細については、「IAM ユーザーガイド」の「[IAM エンティティのアクセス許可の境界](#)」を参照してください。
- **サービスコントロールポリシー (SCPs)** - SCPs は、の組織または組織単位 (OU) に対する最大アクセス許可を指定する JSON ポリシーです AWS Organizations。AWS Organizations は、AWS アカウント ビジネスが所有する複数の をグループ化して一元管理するサービスです。組織内のすべての機能を有効にすると、サービスコントロールポリシー (SCP) を一部またはすべてのアカウントに適用できます。SCP は、各 を含むメンバーアカウントのエンティティのアクセス許可を制限します AWS アカウントのルートユーザー。Organizations と SCP の詳細については、AWS Organizations ユーザーガイドの「[SCP の仕組み](#)」を参照してください。
- **セッションポリシー** - セッションポリシーは、ロールまたはフェデレーションユーザーの一時的なセッションをプログラムで作成する際にパラメータとして渡す高度なポリシーです。結果としてセッションの権限は、ユーザーまたはロールのアイデンティティベースポリシーとセッションポリシーの共通部分になります。また、リソースベースのポリシーから権限が派生する場合があります。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。詳細については、「IAM ユーザーガイド」の「[セッションポリシー](#)」を参照してください。

## 複数のポリシータイプ

1つのリクエストに複数のタイプのポリシーが適用されると、結果として作成される権限を理解するのがさらに難しくなります。複数のポリシータイプが関与する場合にリクエストを許可するかどうか AWS を決定する方法については、IAM ユーザーガイドの「[ポリシー評価ロジック](#)」を参照してください。

## IAM を使用する Amazon Forecast の仕組み

IAM を使用して Forecast へのアクセスを管理する前に、Forecast で利用できる IAM の機能について学びます。

### Amazon Forecast で使用できる IAM の機能

IAM 機能	Forecast サポート
<a href="#">アイデンティティベースのポリシー</a>	Yes
<a href="#">リソースベースのポリシー</a>	No
<a href="#">ポリシーアクション</a>	Yes
<a href="#">ポリシーリソース</a>	はい
<a href="#">ポリシー条件キー (サポート固有)</a>	いいえ
<a href="#">ACL</a>	No
<a href="#">ABAC (ポリシー内のタグ)</a>	はい
<a href="#">一時的な認証情報</a>	Yes
<a href="#">プリンシパル権限</a>	Yes
<a href="#">サービスロール</a>	あり
<a href="#">サービスリンクロール</a>	いいえ

Forecast およびその他の AWS のサービスがほとんどの IAM 機能と連携する方法の概要を把握するには、「IAM ユーザーガイド」の[AWS 「IAM と連携する のサービス](#)」を参照してください。

## Forecast のアイデンティティベースのポリシー

アイデンティティベースポリシーをサポートする **Yes**

アイデンティティベースポリシーは、IAM ユーザー、ユーザーのグループ、ロールなど、アイデンティティにアタッチできる JSON 許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「[IAM ポリシーの作成](#)」を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、およびアクションを許可または拒否する条件を指定できます。プリンシパルは、それが添付されているユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。JSON ポリシーで使用できるすべての要素については、「IAM ユーザーガイド」の「[IAM JSON ポリシーの要素のリファレンス](#)」を参照してください。

### Forecast のアイデンティティベースのポリシーの例

Forecast アイデンティティベースのポリシーの例を表示するには、「[Amazon Forecast のアイデンティティベースのポリシーの例](#)」を参照してください。

## Forecast 内のリソースベースのポリシー

リソースベースのポリシーのサポート **No**

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシー や Amazon S3 バケットポリシー があげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、[プリンシパルを指定する](#)必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーテッドユーザー、またはを含めることができます AWS のサービス。

クロスアカウントアクセスを有効にするには、アカウント全体、または別のアカウントの IAM エンティティをリソースベースのポリシーのプリンシパルとして指定します。リソースベースのポリシーにクロスアカウントのプリンシパルを追加しても、信頼関係は半分しか確立されない点に注意してください。プリンシパルとリソースが異なる がある場合 AWS アカウント、信頼されたアカウントの IAM 管理者は、プリンシパルエンティティ (ユーザーまたはロール) にリソースへのアクセス許可も付与する必要があります。IAM 管理者は、アイデンティティベースのポリシーをエンティティにアタッチすることで権限を付与します。ただし、リソースベースのポリシーで、同じアカウントのプリンシパルへのアクセス権が付与されている場合は、アイデンティティベースのポリシーをさらに付与する必要はありません。詳細については、[「IAM ユーザーガイド」の「IAM でのクロスアカウントリソースアクセス」](#)を参照してください。

## Forecast のポリシーアクション

ポリシーアクションに対するサポート はい

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

JSON ポリシーの Action 要素には、ポリシー内のアクセスを許可または拒否するために使用できるアクションが記述されます。ポリシーアクションの名前は通常、関連付けられた AWS API オペレーションと同じです。一致する API オペレーションのない許可のみのアクションなど、いくつかの例外があります。また、ポリシーに複数のアクションが必要なオペレーションもあります。これらの追加アクションは、依存アクションと呼ばれます。

このアクションは、関連付けられたオペレーションを実行するための権限を付与するポリシーで使用されます。

Forecast アクションのリストを確認するには、「サービス認証リファレンス」の「[Amazon Forecast で定義されるアクション](#)」を参照してください。

Forecast のポリシーアクションは、アクションの前に次のプレフィックスを使用します。

```
forecast
```

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。

```
"Action": [
```

```
"forecast:action1",  
"forecast:action2"  
]
```

## Forecast に関するポリシーリソース

ポリシーリソースに対するサポート はい

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースにどのような条件でアクションを実行できるかということです。

Resource JSON ポリシー要素は、アクションが適用されるオブジェクトを指定します。ステートメントには、Resource または NotResource 要素を含める必要があります。ベストプラクティスとして、[Amazon リソースネーム \(ARN\)](#) を使用してリソースを指定します。これは、リソースレベルの許可と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

オペレーションのリスト化など、リソースレベルの権限をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (\*) を使用します。

```
"Resource": "*"
```

Forecast リソースのタイプとその ARN の一覧を確認するには、「サービス認証リファレンス」の「[Amazon Forecast で定義されるリソース](#)」を参照してください。どのアクションで各リソースの ARN を指定できるかについては、「[Amazon Forecast で定義されるアクション](#)」を参照してください。

Forecast アイデンティティベースのポリシーの例を表示するには、「[Amazon Forecast のアイデンティティベースのポリシーの例](#)」を参照してください。

## Forecast のポリシー条件キー

サービス固有のポリシー条件キーのサポート いいえ

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

Condition 要素 (または Condition ブロック) を使用すると、ステートメントが有効な条件を指定できます。Condition 要素はオプションです。イコールや未満などの [条件演算子](#) を使用して条件式を作成することで、ポリシーの条件とリクエスト内の値を一致させることができます。

1つのステートメントに複数の Condition 要素を指定するか、1つの Condition 要素に複数のキーを指定すると、AWS は AND 論理演算子を使用してそれら进行评估します。1つの条件キーに複数の値を指定すると、は論理ORオペレーションを使用して条件 AWS を評価します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば IAM ユーザーに、IAM ユーザー名がタグ付けされている場合のみリソースにアクセスできる権限を付与することができます。詳細については、「IAM ユーザーガイド」の [「IAM ポリシーの要素: 変数およびタグ」](#) を参照してください。

AWS は、グローバル条件キーとサービス固有の条件キーをサポートします。すべての AWS グローバル条件キーを確認するには、「IAM ユーザーガイド」の [AWS 「グローバル条件コンテキストキー」](#) を参照してください。

Forecast での条件キーの一覧については、「サービス認証リファレンス」の [「Amazon Forecast の条件キー」](#) を参照してください。どのアクションおよびリソースと条件キーを使用できるかについては、「[Amazon Forecast で定義されるアクション](#)」を参照してください。

Forecast アイデンティティベースのポリシーの例を表示するには、「[Amazon Forecast のアイデンティティベースのポリシーの例](#)」を参照してください。

## Forecast の ACL

ACL のサポート	No
-----------	----

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための許可を持つかをコントロールします。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

## Forecast での ABAC

ABAC のサポート (ポリシー内のタグ)	はい
-----------------------	----

属性ベースのアクセス制御 (ABAC) は、属性に基づいてアクセス許可を定義するアクセス許可戦略です。では AWS、これらの属性はタグと呼ばれます。タグは、IAM エンティティ (ユーザーまたはロール) および多くの AWS リソースにアタッチできます。エンティティとリソースのタグ付けは、ABAC の最初の手順です。その後、プリンシパルのタグがアクセスしようとしているリソースのタグと一致した場合にオペレーションを許可するように ABAC ポリシーをします。

ABAC は、急成長する環境やポリシー管理が煩雑になる状況で役立ちます。

タグに基づいてアクセスを管理するには、`aws:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の条件キーを使用して、ポリシーの [条件要素](#) でタグ情報を提供します。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値ははいです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

ABAC の詳細については、IAM ユーザーガイドの「[ABAC とは?](#)」を参照してください。ABAC をセットアップするステップを説明するチュートリアルについては、「IAM ユーザーガイド」の「[属性ベースのアクセス制御 \(ABAC\) を使用する](#)」を参照してください。

## Forecast を使用した一時的な認証情報の使用

一時的な認証情報のサポート	はい
---------------	----

一部の は、一時的な認証情報を使用してサインインすると機能 AWS のサービスしません。一時的な認証情報 AWS のサービス を使用する などの詳細については、IAM ユーザーガイドの [AWS のサービス「IAM と連携する](#)」を参照してください。

ユーザー名とパスワード以外の AWS Management Console 方法で にサインインする場合、一時的な認証情報を使用します。例えば、会社の Single Sign-On (SSO) リンク AWS を使用して にアクセスすると、そのプロセスによって一時的な認証情報が自動的に作成されます。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時的な認証情報が自動的に作成されます。ロールの切り替えに関する詳細については、「IAM ユーザーガイド」の「[ロールへの切り替え \(コンソール\)](#)」を参照してください。

一時的な認証情報は、AWS CLI または AWS API を使用して手動で作成できます。その後、これらの一時的な認証情報を使用して .AWS recommends にアクセスできます AWS。この際、長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成することをお勧めします。詳細については、「[IAM の一時的セキュリティ認証情報](#)」を参照してください。



## Forecast のクロスサービスプリンシパル許可

フォワードアクセスセッション (FAS) をサポート  はい  
 否

IAM ユーザーまたはロールを使用してアクションを実行すると AWS、プリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、 を呼び出すプリンシパルのアクセス許可を AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストリクエストリクエストと組み合わせて使用します。FAS リクエストは、サービスが他の AWS のサービス またはリソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「[転送アクセスセッション](#)」を参照してください。

## Forecast のサービスロール

サービスロールに対するサポート  あり  
 ない

サービスロールとは、サービスがユーザーに代わってアクションを実行するために引き受ける [IAM ロール](#)です。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、「IAM ユーザーガイド」の「[AWS のサービスにアクセス許可を委任するロールの作成](#)」を参照してください。

### Warning

サービスロールの許可を変更すると、Forecast の機能が破損する可能性があります。Forecast が指示する場合以外は、サービスロールを編集しないでください。

## Forecast のサービスリンクロール

サービスにリンクされたロールのサポート  いいえ  
 はい

サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービ

スにリンクされたロールは に表示され AWS アカウント、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールの権限を表示できますが、編集することはできません。

サービスにリンクされたロールの作成または管理の詳細については、「[IAM と提携するAWS のサービス](#)」を参照してください。表の中から、Service-linked role (サービスにリンクされたロール) 列に Yes と記載されたサービスを見つけます。サービスリンクロールに関するドキュメントをサービスで表示するには、はい リンクを選択します。

## Amazon Forecast のアイデンティティベースのポリシーの例

デフォルトでは、ユーザーおよびロールには、Forecast リソースを作成または変更する許可はありません。また、AWS Command Line Interface (AWS CLI) AWS Management Console、または AWS API を使用してタスクを実行することはできません。IAM 管理者は、リソースに必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き受けることができます。

これらサンプルの JSON ポリシードキュメントを使用して、IAM アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「[IAM ポリシーの作成](#)」を参照してください。

Forecast が定義するアクションとリソースタイプ (リソースタイプごとの ARN の形式を含む) の詳細については、「サービス認可リファレンス」の「[Amazon Forecast のアクション、リソース、および条件キー](#)」を参照してください。

オペレーションが呼び出されるたびに、Amazon Forecast は呼び出し元の権限について一連の認証チェックを実行します。これらのチェックには以下が含まれます。

- 呼び出し元がオペレーションを呼び出す権限が検証されます。
- オペレーション内でロールが指定されている場合、Amazon Forecast はロールの PassRole アクセス許可を検証します。
- 暗号化設定で KMS キーが指定されている場合、発信者のアクセス許可に対して kms:Decrypt および kms:GenerateDataKey validation が実行されます。このキーは、Amazon Forecast で実行されるオペレーションごとに異なる場合があります。関連するアクセス許可 AccessDeniedException が無い場合、を受け取ります。キーポリシーは次のコードのようになっているはずで

### Example

```
"Effect": "Allow",
"Principal": {
```

```
    "AWS": "AWS Invoking Identity",
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
  ],
  "Resource": "*"
}
```

## トピック

- [ポリシーのベストプラクティス](#)
- [Forecast コンソールの使用](#)
- [自分の権限の表示をユーザーに許可する](#)
- [AWS Amazon Forecast のマネージド \(事前定義\) ポリシー](#)
- [お客様が管理するポリシーの例](#)

## ポリシーのベストプラクティス

ID ベースのポリシーは、ユーザーのアカウントで誰かが Forecast リソースを作成、アクセス、または削除できるかどうかを決定します。これらのアクションを実行すると、AWS アカウントに料金が発生する可能性があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS 管理ポリシーを開始し、最小特権のアクセス許可に移行する – ユーザーとワークロードにアクセス許可を付与するには、多くの一般的なユースケースにアクセス許可を付与する AWS 管理ポリシーを使用します。これらはで使用できます AWS アカウント。ユースケースに固有の AWS カスタマー管理ポリシーを定義して、アクセス許可をさらに減らすことをお勧めします。詳細については、「IAM ユーザーガイド」の「[AWS マネージドポリシー](#)」または「[AWS ジョブ機能の管理ポリシー](#)」を参照してください。
- 最小特権を適用する – IAM ポリシーで許可を設定する場合は、タスクの実行に必要な許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用して許可を適用する方法の詳細については、「IAM ユーザーガイド」の「[IAM でのポリシーとアクセス許可](#)」を参照してください。
- IAM ポリシーで条件を使用してアクセスをさらに制限する - ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。例えば、ポリシー条件を記述して、すべてのリクエ

ストを SSL を使用して送信するように指定できます。条件を使用して、などの特定の を介して サービスアクションが使用される場合に AWS のサービス、サービスアクションへのアクセスを許可することもできます AWS CloudFormation。詳細については、「IAM ユーザーガイド」の [IAM JSON policy elements: Condition](#) (IAM JSON ポリシー要素: 条件) を参照してください。

- IAM Access Analyzer を使用して IAM ポリシーを検証し、安全で機能的な権限を確保する - IAM Access Analyzer は、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM アクセスアナライザーは 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーの作成をサポートします。詳細については、「IAM ユーザーガイド」の「[IAM Access Analyzer ポリシーの検証](#)」を参照してください。
- 多要素認証 (MFA) を要求する - で IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、セキュリティを強化するために MFA を有効にします。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、「IAM ユーザーガイド」の「[MFA 保護 API アクセスの設定](#)」を参照してください。

IAM でのベストプラクティスの詳細については、「IAM ユーザーガイド」の「[IAM でのセキュリティのベストプラクティス](#)」を参照してください。

## Forecast コンソールの使用

Amazon Forecast コンソールにアクセスするには、許可の最小限のセットが必要です。これらのアクセス許可により、 の Forecast リソースの詳細を一覧表示および表示できます AWS アカウント。最小限必要な許可よりも制限が厳しいアイデンティティベースのポリシーを作成すると、そのポリシーを持つエンティティ (ユーザーまたはロール) に対してコンソールが意図したとおりに機能しません。

AWS CLI または AWS API のみを呼び出すユーザーには、最小限のコンソールアクセス許可を付与する必要はありません。代わりに、実行しようとしている API オペレーションに一致するアクションのみへのアクセスが許可されます。

ユーザーとロールが引き続き Forecast コンソールを使用できるようにするには、エンティティに次の AWS 管理ポリシーもアタッチします。詳細については、「IAM ユーザーガイド」の「[ユーザーへの許可の追加](#)」を参照してください。

```
AWSForecastFullAccess
```

次のポリシーは、コンソールを使用するときすべての Amazon Forecast アクションへのフルアクセスを許可します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "forecast:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "forecast.amazonaws.com"
        }
      }
    }
  ]
}
```

## 自分の権限の表示をユーザーに許可する

この例では、ユーザーアイデンティティにアタッチされたインラインおよびマネージドポリシーの表示を IAM ユーザーに許可するポリシーの作成方法を示します。このポリシーには、コンソールで、または AWS CLI または AWS API を使用してプログラムでこのアクションを実行するアクセス許可が含まれています。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsForUser",
        "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": "*"
}
]
```

## AWS Amazon Forecast のマネージド (事前定義) ポリシー

AWS は、によって作成および管理されるスタンドアロン IAM ポリシーを提供することで、多くの一般的なユースケースに対処します AWS。これらの AWS 管理ポリシーは、一般的なユースケースに必要なアクセス許可を付与するため、どのアクセス許可が必要かを調査する必要がなくなります。詳細については、「IAM ユーザーガイド」の「[AWS マネージドポリシー](#)」を参照してください。

アカウントのユーザーにアタッチできる以下の AWS マネージドポリシーは、Amazon Forecast に固有のものであります。

- AmazonForecastFullAccess – Amazon Forecast リソースとサポートされているすべてのオペレーションへのフルアクセスを許可します。

これらの許可ポリシーについては、IAM コンソールにサインインしてそれらを検索することで確認できます。

独自のカスタム IAM ポリシーを作成して、Amazon Forecast アクションとリソースのための権限を許可することもできます。これらのカスタムポリシーは、それらを必要とする IAM ユーザーにアタッチできます。

## お客様が管理するポリシーの例

このセクションでは、さまざまな Amazon Forecast アクションの許可を付与するユーザーポリシー例を示しています。これらのポリシーは、AWS SDKsまたはを使用しているときに機能します AWS CLI。コンソールを使用している場合は、「[Forecast コンソールの使用](#)」を参照してください。

### 例

- [例 1: アカウント管理者にアクセス許可を付与する](#)
- [例 2: すべての Amazon Forecast および IAM PassRole アクションを許可する](#)
- [例 3: IAM アクションを制限しながらすべての Amazon Forecast PassRole アクションを許可する](#)
- [例 4: アクションベースのポリシー: Amazon Forecast 読み取り専用アクセス](#)
- [例 5: パスロールと KMS アクションを含むすべての Amazon Forecast アクションを許可する](#)

### 例 1: アカウント管理者にアクセス許可を付与する

アカウントを設定した後で（「[AWS へのサインアップ](#)」を参照）、管理者ユーザーを作成し、アカウントを管理します。管理者ユーザーは、ユーザーを作成してそのアクセス許可を管理できます。

アカウントで使用可能なすべてのアクセス許可を管理者ユーザーに付与するには、そのユーザーに次のアクセス許可ポリシーをアタッチします

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

## 例 2: すべての Amazon Forecast および IAM PassRole アクションを許可する

すべての Amazon Forecast アクションに対する許可を持ち、他のサービスに対する許可を持たないユーザーを作成することを選択できます (このユーザーをサービス固有の管理者とみなします)。このユーザーに以下のアクセス許可ポリシーをアタッチします。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "forecast:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "forecast.amazonaws.com"
        }
      }
    }
  ]
}
```

## 例 3: IAM アクションを制限しながらすべての Amazon Forecast PassRole アクションを許可する

IAM アクションを制限しながら、すべての Amazon Forecast PassRole アクションに対するアクセス許可を持つユーザーを作成することもできます。このユーザーに以下のアクセス許可ポリシーをアタッチします。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```



```

    "Action": [
      "forecast:*"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "iam:PassRole"
    ],
    "Resource": "arn:aws:iam::EXAMPLE_ACCOUNT_ID_12349858:role/
EXAMPLE_ROLE_TO_ALLOW_TO_PASS",
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": "forecast.amazonaws.com"
      }
    }
  }
]
}

```

#### 例 4: アクションベースのポリシー: Amazon Forecast 読み取り専用アクセス

以下のポリシーでは、リソースを一覧表示して説明する Amazon Forecast アクションを実行するための許可をユーザーに付与します。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "forecast:DescribeDataset",
        "forecast:DescribeDatasetGroup",
        "forecast:DescribeDatasetImportJob",
        "forecast:DescribeForecast",
        "forecast:DescribeForecastExportJob",
        "forecast:DescribePredictor",
        "forecast:ListDatasetGroups",
        "forecast:ListDatasetImportJobs",
        "forecast:ListDatasets",
        "forecast:ListDatasetExportJobs",
        "forecast:ListForecasts",

```

```
        "forecast:ListPredictors"
    ],
    "Resource": "*"
}
]
```

#### 例 5: パスロールと KMS アクションを含むすべての Amazon Forecast アクションを許可する

Amazon Forecast の暗号化用のクロスアカウントカスタマーマネージドキーを使用して、すべての Amazon Forecast アクションに対する許可を持ち、他のサービスに対する許可を持たないユーザーを作成できます。詳細については、「AWS Key Management Service デベロッパーガイド」の [AWS「クロスアカウントキーポリシー」](#) を参照してください。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "forecast:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "forecast.amazonaws.com"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": "arn:aws:iam::1234567890:key/example_key"
    }
  ]
}
```

```
    }  
  ]  
}
```

## Amazon Forecast アイデンティティとアクセスのトラブルシューティング

次の情報は、Forecast と IAM の使用に伴って発生する可能性がある一般的な問題の診断や修復に役立ちます。

### トピック

- [Forecast でアクションを実行する権限がない](#)
- [iam を実行する権限がありません。PassRole](#)
- [管理者として Forecast へのアクセスを他のユーザーに許可したい](#)
- [自分の 以外のユーザーに Forecast リソース AWS アカウント へのアクセスを許可したい](#)

### Forecast でアクションを実行する権限がない

「I am not authorized to perform an action in Amazon Bedrock」というエラーが表示された場合、そのアクションを実行できるようにポリシーを更新する必要があります。

次のエラー例は、mateojackson IAM ユーザーがコンソールを使用して、ある *my-example-widget* リソースに関する詳細情報を表示しようとしたことを想定して、その際に必要な `forecast:GetWidget` アクセス許可を持っていない場合に発生するものです。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:  
forecast:GetWidget on resource: my-example-widget
```

この場合、`forecast:GetWidget` アクションを使用して *my-example-widget* リソースへのアクセスを許可するように、mateojackson ユーザーのポリシーを更新する必要があります。

サポートが必要な場合は、AWS 管理者にお問い合わせください。サインイン認証情報を提供した担当者が管理者です。

### iam を実行する権限がありません。PassRole

`iam:PassRole` アクションを実行する権限がないというエラーが表示された場合は、ポリシーを更新して Forecast にロールを渡すことができるようにする必要があります。

一部の AWS のサービスでは、新しいサービスロールまたはサービスにリンクされたロールを作成する代わりに、そのサービスに既存のロールを渡すことができます。そのためには、サービスにロールを渡す権限が必要です。

以下の例のエラーは、marymajor という IAM ユーザーがコンソールを使用して Forecast でアクションを実行しようとする場合に発生します。ただし、このアクションをサービスが実行するには、サービスロールから付与された権限が必要です。メアリーには、ロールをサービスに渡す許可がありません。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

この場合、Mary のポリシーを更新してメアリーに iam:PassRole アクションの実行を許可する必要があります。

サポートが必要な場合は、AWS 管理者にお問い合わせください。サインイン認証情報を提供した担当者が管理者です。

## 管理者として Forecast へのアクセスを他のユーザーに許可したい

Forecast へのアクセスを他のユーザーに許可するには、アクセスを必要とする人またはアプリケーションの IAM エンティティ (ユーザーまたはロール) を作成する必要があります。ユーザーまたはアプリケーションは、そのエンティティの認証情報を使用して AWS にアクセスします。次に、Forecast の適切な許可を付与するポリシーを、そのエンティティにアタッチする必要があります

すぐに開始するには、IAM ユーザーガイドの「[IAM が委任した最初のユーザーおよびグループの作成](#)」を参照してください。

## 自分の 以外のユーザーに Forecast リソース AWS アカウント へのアクセスを許可したい

他のアカウントのユーザーや組織外の人が、リソースにアクセスするために使用できるロールを作成できます。ロールの引き受けを委託するユーザーを指定できます。リソースベースのポリシーまたはアクセスコントロールリスト (ACL) をサポートするサービスの場合、それらのポリシーを使用して、リソースへのアクセスを付与できます。

詳細については、以下を参照してください:

- Forecast がこれらの機能をサポートしているかどうかを確認するには、「[IAM を使用する Amazon Forecast の仕組み](#)」を参照してください。
- 所有 AWS アカウント している のリソースへのアクセスを提供する方法については、[IAM ユーザーガイドの「所有 AWS アカウント している別の の IAM ユーザーへのアクセスを提供する」](#)を参照してください。
- リソースへのアクセスをサードパーティー に提供する方法については AWS アカウント、IAM ユーザーガイドの「[サードパーティー AWS アカウント が所有する へのアクセス](#)を提供する」を参照してください。
- ID フェデレーションを介してアクセスを提供する方法については、IAM ユーザーガイドの[外部認証されたユーザーへのアクセスの提供 \(ID フェデレーション\)](#)を参照してください。
- クロスアカウントアクセスでのロールとリソースベースのポリシーの使用の違いについては、IAM ユーザーガイドの「[IAM でのクロスアカウントリソースアクセス](#)」を参照してください。

## Amazon Forecast でのログ記録とモニタリング

モニタリングは、Amazon Forecast アプリケーションの信頼性、可用性、パフォーマンスを維持する上で重要な部分です。Amazon Forecast の API コールをモニタリングするために、AWS CloudTrailを使用できます。Forecast のアセットとプロセスのステータスをモニタリングするには、Amazon を使用します CloudWatch。

### トピック

- [を使用した Forecast API コールのログ記録 AWS CloudTrail](#)
- [CloudWatch Amazon Forecast のメトリクス](#)

## を使用した Forecast API コールのログ記録 AWS CloudTrail

Amazon Forecast は と統合されています。これは AWS CloudTrail、Forecast のユーザー、ロール、または サービスによって実行されたアクションを記録する AWS サービスです。Forecast のすべての API コールをイベントとして CloudTrail キャプチャします。キャプチャされたコールには、Forecast コンソールからのコールと Forecast API オペレーションへのコードコールが含まれます。証跡を作成する場合は、Forecast の CloudTrail イベントなど、Amazon Simple Storage Service (Amazon S3) バケットへのイベントの継続的な配信を有効にすることができます。証跡を設定しない場合でも、CloudTrail コンソールのイベント履歴 で最新のイベントを表示できます。によって収集された情報を使用して CloudTrail、Forecast に対するリクエスト、リクエスト元の IP アドレス、リクエスト者、リクエスト日時などの詳細を確認できます。

の詳細については CloudTrail、[「AWS CloudTrail ユーザーガイド」](#)を参照してください。

## の予測情報 CloudTrail

CloudTrail AWS アカウントを作成すると、[がアカウントで有効になります](#)。Forecast でアクティビティが発生すると、そのアクティビティは CloudTrail イベント履歴の他の AWS サービスイベントとともにイベントに記録されます。AWS アカウントで最近のイベントを表示、検索、ダウンロードできます。詳細については、[「イベント履歴を使用した CloudTrail イベントの表示」](#)を参照してください。

Forecast のイベントなど、AWS アカウント内のイベントの継続的な記録については、証跡を作成します。証跡により CloudTrail はログファイルを Amazon S3 バケットに配信できます。デフォルトでは、コンソールで証跡を作成するときに、証跡がすべての AWS リージョンに適用されます。証跡は、AWS パーティション内のすべてのリージョンからのイベントをログに記録し、指定した Amazon S3 バケットにログファイルを配信します。さらに、CloudTrail ログで収集されたイベントデータをより詳細に分析し、それに基づいて行動するように他の AWS サービスを設定できます。詳細については、次を参照してください:

- [証跡の作成のための概要](#)
- [CloudTrail サポートされているサービスと統合](#)
- [の Amazon SNS 通知の設定 CloudTrail](#)
- [複数のリージョンからの CloudTrail ログファイルの受信と複数のアカウントからのログファイルの受信 CloudTrail](#)

すべての Forecast アクションは [によってログに記録 CloudTrail](#) され、[Amazon Forecast デベロッパーガイド](#) に記載されています。例えば、[および CreateForecast](#) アクションを呼び出す [CreateDataset](#) と、CloudTrail ログファイルにエントリが生成されます。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます:

- リクエストがルートまたは AWS Identity and Access Management (IAM) ユーザーの認証情報を使用して行われたかどうか。
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが別の AWS サービスによって行われたかどうか。

詳細については、[CloudTrail userIdentity Element](#) を参照してください。

## Forecast ログファイルエントリについて

証跡は、指定した Amazon S3 バケットにイベントをログファイルとして配信できるようにする設定です。CloudTrail ログファイルには 1 つ以上のログエントリが含まれます。イベントは任意のソースからの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストパラメータなどに関する情報が含まれます。CloudTrail ログファイルはパブリック API コールの順序付けられたスタックトレースではないため、特定の順序では表示されません。

次の例は、CreateDataset アクションを示す CloudTrail ログエントリを示しています。

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAIQ4PAJSMEEPNEXAMPLE",
    "arn": "arn:aws:iam::acct-id:user/userxyz",
    "accountId": "111111111111",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "userxyz"
  },
  "eventTime": "2018-11-21T23:53:06Z",
  "eventSource": "forecast.amazonaws.com",
  "eventName": "CreateDataset",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.168.0.1",
  "userAgent": "Boto3/1.7.82 Python/3.6.5 Linux/4.14.72-68.55.amzn1.x86_64
  Botocore/1.10.84",
  "requestParameters": {
    "domain": "CUSTOM",
    "datasetType": "TARGET_TIME_SERIES",
    "dataFormat": "CSV",
    "datasetName": "forecast_test_script_ds",
    "dataFrequency": "D",
    "timestampFormat": "yyyy-MM-dd",
    "schema": {
      "attributes": [
        {
          "attributeName": "item_id",
          "attributeType": "string"
        }
      ]
    }
  },
}
```

```
{
  {
    "attributeName": "timestamp",
    "attributeType": "timestamp"
  },
  {
    "attributeName": "target_value",
    "attributeType": "float"
  },
  {
    "attributeName": "visits",
    "attributeType": "float"
  },
  {
    "attributeName": "was_open",
    "attributeType": "float"
  },
  {
    "attributeName": "promotion_applied",
    "attributeType": "float"
  }
]
},
"responseElements": {
  "datasetName": "forecast_test_script_ds",
  "datasetArn": "arn:aws:forecast:us-west-2:acct-id:ds/forecast_test_script_ds"
},
"requestID": "EXAMPLE8-90ab-cdef-fedc-ba987EXAMPLE",
"eventID": "EXAMPLE8-90ab-cdef-fedc-ba987EXAMPLE",
"eventType": "AwsApiCall",
"recipientAccountId": "111111111111"
}
```

## CloudWatch Amazon Forecast のメトリクス

このセクションでは、Amazon Forecast で使用できる Amazon CloudWatch メトリクスについて説明します。

次の表に、Amazon Forecast のメトリクスを示します。



メトリクス	ディメンション	単位	統計	説明
DatasetSize		Kilobytes	Average、Sum、Min、Max	Amazon Forecast によってお客様のアカウントにインポートされたデータセットの合計サイズ。
DatasetSize	DatasetArn DatasetImportJobArn	Kilobytes	Average、Sum	<a href="#">CreateDatasetImportJob</a> オペレーションによってインポートされたデータセットのサイズ。
CreatePredictorEvaluationTime	PredictorArn	[秒]	Average、Sum	特定の予測子のトレーニング、推論、およびメトリクスにかかる時間。Amazon Forecast は、コンピューティングコストを c5.xlarge インスタンスに正規化して、トレーニングジョブによって消費された時間数に到達します。
CreateForecastEvaluationTime	ForecastArn	[秒]	Average、Sum	予測生成中のトレーニングと推論にかかる時間。Amazon Forecast は、コンピューティングコストを c5.xlarge インスタンスに正規化して、トレーニングジョブによって消費された時間数に到達します。
TimeSeriesForecastGenerated		カウント	Average、Sum、Min、Max	アカウント内のすべての予測子にわたって分位ごとに生成された一意の時系列予測の数。予測は最も近い 1,000 に切り上げられ、1,000 ごとに請求されます。
TimeSeriesForecastGenerated	PredictorArn	カウント	Average、Sum、Min、Max	アカウント内のすべての予測子にわたって分位ごとに生成された一意の時系列予測の数。予測は最も近い 1,000 に切り上げられ、1,000 ごとに請求されます。

メトリクス	ディメンション	単位	統計	説明
TimeSeriesForecastGenerated	Predictor Arn ForecastArn	カウント	Average、Sum、Min、Max	アカウント内のすべての予測子にわたって分位ごとに生成された一意の時系列予測の数。予測は最も近い 1,000 に切り上げられ、1,000 ごとに請求されます。
ForecastDataPointsGenerated	Predictor Arn ForecastArn	カウント	Average、Sum、Min、Max	アカウント内のすべての予測子にわたって予想ごとに生成された一意のデータポイントの数。予測は最も近い 1,000 に切り上げられ、1,000 ごとに請求されます。

## Amazon Forecast のコンプライアンス検証

サードパーティーの監査者は、複数のコンプライアンスプログラムの一環として Amazon Forecast のセキュリティと AWS コンプライアンスを評価します。このプログラムには、SOC、PCI、HIPAA などを含まれます。

特定のコンプライアンスプログラムの対象となる AWS サービスのリストについては、「[コンプライアンスプログラムAWSによる対象範囲内のサービスコンプライアンスプログラム](#)」を参照してください。一般的な情報については、[AWS「コンプライアンスプログラム」](#)を参照してください。

を使用して、サードパーティーの監査レポートをダウンロードできます AWS Artifact。詳細については、「[アーティファクトでの AWS レポートのダウンロード](#)」を参照してください。

Forecast を使用する際のお客様のコンプライアンス責任は、お客様のデータの機密性、貴社のコンプライアンス目的、適用される法律および規制によって決まります。では、コンプライアンスに役立つ以下のリソース AWS を提供しています。

- 「[セキュリティ&コンプライアンスクイックリファレンスガイド](#)」 - これらのデプロイガイドには、アーキテクチャ上の考慮事項の説明と、AWSでセキュリティとコンプライアンスに重点を置いたベースライン環境をデプロイするための手順が記載されています。
- [HIPAA セキュリティとコンプライアンスのアーキテクチャに関するホワイトペーパー](#) — このホワイトペーパーでは、企業が AWS を使用して HIPAA 準拠のアプリケーションを作成する方法について説明します。

- [AWS コンプライアンスリソース](#) – このワークブックとガイドのコレクションは、お客様の業界や地域に適用される場合があります。
- [「デベロッパーガイド」の「ルールによるリソースの評価」](#) – この AWS Config サービスは、リソース設定が社内プラクティス、業界ガイドライン、および規制にどの程度準拠しているかを評価します。AWS Config
- [AWS Security Hub](#) – この AWS サービスは、内のセキュリティ状態を包括的に把握 AWS し、セキュリティ業界標準とベストプラクティスへの準拠を確認するのに役立ちます。

## Amazon Forecast の回復力

AWS グローバルインフラストラクチャは、AWS リージョンとアベイラビリティゾーンを中心に構築されています。AWS リージョンは、低レイテンシー、高スループット、および高度に冗長なネットワークで接続された、物理的に分離および分離された複数のアベイラビリティゾーンを提供します。アベイラビリティゾーンでは、ゾーン間で中断することなく自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性が高く、フォールトトレラントで、スケーラブルです。

AWS リージョンとアベイラビリティゾーンの詳細については、[AWS 「グローバルインフラストラクチャ」](#)を参照してください。

## Amazon Forecast のインフラストラクチャセキュリティ

マネージドサービスである Amazon Forecast は、AWS グローバルネットワークセキュリティで保護されています。AWS セキュリティサービスと [ガインフラストラクチャ AWS](#) を保護する方法については、[AWS 「クラウドセキュリティ」](#)を参照してください。インフラストラクチャセキュリティのベストプラクティスを使用して AWS 環境を設計するには、「Security Pillar AWS Well-Architected Framework」の [「Infrastructure Protection」](#)を参照してください。

AWS が公開した API コールを使用して、ネットワーク経由で Forecast にアクセスします。クライアントは以下をサポートする必要があります：

- Transport Layer Security (TLS)。TLS 1.2 は必須で TLS 1.3 がお勧めです。
- DHE (楕円ディフィー・ヘルマン鍵共有) や ECDHE (楕円曲線ディフィー・ヘルマン鍵共有) などの完全前方秘匿性 (PFS) による暗号スイート。これらのモードは、Java 7 以降など、ほとんどの最新システムでサポートされています。

また、リクエストには、アクセスキー ID と、IAM プリンシパルに関連付けられているシークレットアクセスキーを使用して署名する必要があります。または、[AWS Security Token Service \(AWS STS\)](#) を使用して、一時的なセキュリティ認証情報を生成し、リクエストに署名することもできます。

## Forecast とインターフェイス VPC エンドポイント (AWS PrivateLink)

Amazon Virtual Private Cloud (Amazon VPC) を使用して AWS リソースをホストする場合、VPC と Amazon Forecast の間にプライベート接続を確立できます。この接続により、Amazon Forecast はパブリックインターネットを経由せずに、VPC のリソースと通信できます。

Amazon VPC は AWS のサービス、ユーザーが定義した仮想プライベートクラウド (VPC) または仮想ネットワークで AWS リソースを起動するために使用する です。VPC を使用することで、IP アドレス範囲、サブネット、ルートテーブル、ネットワークゲートウェイなどのネットワーク設定を制御できます。VPC エンドポイントでは、AWS ネットワークは VPC と 間のルーティングを処理します AWS のサービス。

VPC を Amazon Forecast に接続するには、Amazon Forecast のインターフェイス VPC エンドポイントを定義します。インターフェイスエンドポイントは、サポートされる AWS のサービスを宛先とするトラフィックのエントリポイントとなるプライベート IP アドレスを持つ Elastic Network Interface です。このエンドポイントは、インターネットゲートウェイ、ネットワークアドレス変換 (NAT) インスタンス、または VPN 接続を必要とせず、信頼性が高くスケーラブルな Amazon Forecast への接続を提供します。詳細については、「Amazon VPC ユーザーガイド」の「[Amazon VPC とは](#)」を参照してください。

インターフェイス VPC エンドポイントは によって有効になります AWS PrivateLink。この AWS テクノロジーにより、プライベート IP アドレスを持つ Elastic Network Interface AWS のサービス を使用して、間のプライベート通信が可能になります。

### Note

すべての Amazon Forecast Federal Information Processing Standard (FIPS) エンドポイントは、でサポートされています AWS PrivateLink。

## Forecast VPC エンドポイントの考慮事項

Forecast のインターフェイス VPC エンドポイントをセットアップする前に、「Amazon VPC ユーザーガイド」で「[インターフェイスエンドポイントのプロパティと制限](#)」を確認するようにしてください。

Forecast は、VPC からのすべての API アクションの呼び出しをサポートしています。

## Forecast 用のインターフェイス VPC エンドポイントの作成

Forecast サービスの VPC エンドポイントは、Amazon VPC コンソールまたは AWS Command Line Interface (CLI) を使用して作成できます。詳細については、Amazon VPC ユーザーガイドの[インターフェイスエンドポイントの作成](#)を参照してください。

Amazon Forecast で使用する 2 種類の VPC エンドポイントを作成できます。

- Amazon Forecast オペレーションで使用する VPC エンドポイント。ほとんどのユーザーにとって、これは VPC エンドポイントの最適なタイプです。
  - `com.amazonaws.region.forecast`
  - `com.amazonaws.region.forecastquery`
- 連邦情報処理標準 (FIPS) 刊行物 140-2 米国政府標準に準拠するエンドポイントを使用した、Amazon Forecast オペレーションの VPC エンドポイント (一部のリージョンで利用可能、「[Amazon Forecast エンドポイントとクォータ](#)」を参照)。
  - `com.amazonaws.region.forecast-fips`
  - `com.amazonaws.region.forecastquery-fips`

エンドポイントに対してプライベート DNS を有効にすると、リージョンのデフォルト DNS 名 (`forecast.us-east-1.amazonaws.com` など) を使用して、Forecast への API リクエストを実行できます。

詳細については、「Amazon VPC ユーザーガイド」の「[インターフェイスエンドポイントを介したサービスへのアクセス](#)」を参照してください。

## Forecast 用の VPC エンドポイントポリシーの作成

VPC エンドポイントには、Forecast へのアクセスを制御するエンドポイントポリシーをアタッチできます。このポリシーでは、以下の情報を指定します。

- アクションを実行できるプリンシパル。
- 実行可能なアクション。
- このアクションを実行できるリソース。

詳細については、「Amazon VPC ユーザーガイド」の「[VPC エンドポイントでサービスへのアクセスを制御する](#)」を参照してください。

例: すべての Forecast アクションと passRole アクションを許可する VPC エンドポイントポリシー

このポリシーは、エンドポイントに接続すると、すべての Forecast アクションと passRole アクションへのアクセスを許可します。

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "forecast:*",
        "iam:PassRole"
      ],
      "Resource": "*"
    }
  ]
}
```

例: Forecast ListDatasets アクションを許可する VPC エンドポイントポリシー

このポリシーは、エンドポイントにアタッチされると、リストされている Forecast ListDatasets アクションへのアクセスを許可します。

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "forecast:ListDatasets"
      ],
      "Resource": "*"
    }
  ]
}
```

```
]
}
```

# API リファレンス

このセクションでは、Amazon Forecast API オペレーションについて説明します。

トピック

- [アクション](#)
- [データ型](#)
- [共通エラー](#)
- [共通パラメータ](#)

## アクション

次のアクションは、Amazon Forecast Service でサポートされています:

- [CreateAutoPredictor](#)
- [CreateDataset](#)
- [CreateDatasetGroup](#)
- [CreateDatasetImportJob](#)
- [CreateExplainability](#)
- [CreateExplainabilityExport](#)
- [CreateForecast](#)
- [CreateForecastExportJob](#)
- [CreateMonitor](#)
- [CreatePredictor](#)
- [CreatePredictorBacktestExportJob](#)
- [CreateWhatIfAnalysis](#)
- [CreateWhatIfForecast](#)
- [CreateWhatIfForecastExport](#)
- [DeleteDataset](#)
- [DeleteDatasetGroup](#)
- [DeleteDatasetImportJob](#)
- [DeleteExplainability](#)



- [DeleteExplainabilityExport](#)
- [DeleteForecast](#)
- [DeleteForecastExportJob](#)
- [DeleteMonitor](#)
- [DeletePredictor](#)
- [DeletePredictorBacktestExportJob](#)
- [DeleteResourceTree](#)
- [DeleteWhatIfAnalysis](#)
- [DeleteWhatIfForecast](#)
- [DeleteWhatIfForecastExport](#)
- [DescribeAutoPredictor](#)
- [DescribeDataset](#)
- [DescribeDatasetGroup](#)
- [DescribeDatasetImportJob](#)
- [DescribeExplainability](#)
- [DescribeExplainabilityExport](#)
- [DescribeForecast](#)
- [DescribeForecastExportJob](#)
- [DescribeMonitor](#)
- [DescribePredictor](#)
- [DescribePredictorBacktestExportJob](#)
- [DescribeWhatIfAnalysis](#)
- [DescribeWhatIfForecast](#)
- [DescribeWhatIfForecastExport](#)
- [GetAccuracyMetrics](#)
- [ListDatasetGroups](#)
- [ListDatasetImportJobs](#)
- [ListDatasets](#)
- [ListExplainabilities](#)
- [ListExplainabilityExports](#)

- [ListForecastExportJobs](#)
- [ListForecasts](#)
- [ListMonitorEvaluations](#)
- [ListMonitors](#)
- [ListPredictorBacktestExportJobs](#)
- [ListPredictors](#)
- [ListTagsForResource](#)
- [ListWhatIfAnalyses](#)
- [ListWhatIfForecastExports](#)
- [ListWhatIfForecasts](#)
- [ResumeResource](#)
- [StopResource](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateDatasetGroup](#)

次のアクションは、Amazon Forecast Query Service でサポートされています:

- [QueryForecast](#)
- [QueryWhatIfForecast](#)

## Amazon Forecast サービス

次のアクションは、Amazon Forecast Service でサポートされています:

- [CreateAutoPredictor](#)
- [CreateDataset](#)
- [CreateDatasetGroup](#)
- [CreateDatasetImportJob](#)
- [CreateExplainability](#)
- [CreateExplainabilityExport](#)
- [CreateForecast](#)

- [CreateForecastExportJob](#)
- [CreateMonitor](#)
- [CreatePredictor](#)
- [CreatePredictorBacktestExportJob](#)
- [CreateWhatIfAnalysis](#)
- [CreateWhatIfForecast](#)
- [CreateWhatIfForecastExport](#)
- [DeleteDataset](#)
- [DeleteDatasetGroup](#)
- [DeleteDatasetImportJob](#)
- [DeleteExplainability](#)
- [DeleteExplainabilityExport](#)
- [DeleteForecast](#)
- [DeleteForecastExportJob](#)
- [DeleteMonitor](#)
- [DeletePredictor](#)
- [DeletePredictorBacktestExportJob](#)
- [DeleteResourceTree](#)
- [DeleteWhatIfAnalysis](#)
- [DeleteWhatIfForecast](#)
- [DeleteWhatIfForecastExport](#)
- [DescribeAutoPredictor](#)
- [DescribeDataset](#)
- [DescribeDatasetGroup](#)
- [DescribeDatasetImportJob](#)
- [DescribeExplainability](#)
- [DescribeExplainabilityExport](#)
- [DescribeForecast](#)
- [DescribeForecastExportJob](#)
- [DescribeMonitor](#)

- [DescribePredictor](#)
- [DescribePredictorBacktestExportJob](#)
- [DescribeWhatIfAnalysis](#)
- [DescribeWhatIfForecast](#)
- [DescribeWhatIfForecastExport](#)
- [GetAccuracyMetrics](#)
- [ListDatasetGroups](#)
- [ListDatasetImportJobs](#)
- [ListDatasets](#)
- [ListExplainabilities](#)
- [ListExplainabilityExports](#)
- [ListForecastExportJobs](#)
- [ListForecasts](#)
- [ListMonitorEvaluations](#)
- [ListMonitors](#)
- [ListPredictorBacktestExportJobs](#)
- [ListPredictors](#)
- [ListTagsForResource](#)
- [ListWhatIfAnalyses](#)
- [ListWhatIfForecastExports](#)
- [ListWhatIfForecasts](#)
- [ResumeResource](#)
- [StopResource](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateDatasetGroup](#)

## CreateAutoPredictor

サービス: Amazon Forecast Service

Amazon Forecast 予測子を作成します。

Amazon Forecast は を使用して予測子を作成します。これには AutoPredictor、データセット内の各時系列にアルゴリズムの最適な組み合わせを適用することが含まれます。新しい予測子を作成したり、既存の予測子をアップグレード/再トレーニングしたりするために [CreateAutoPredictor](#) を使用できます。

### 新しい予測子の作成

新しい予測子を作成するときは、次のパラメータが必要です。

- PredictorName - 予測子の一意の名前。
- DatasetGroupArn - 予測子のトレーニングに使用されるデータセットグループの ARN。
- ForecastFrequency - 予測の粒度 (毎時、毎日、毎週など)。
- ForecastHorizon - モデルが予測するタイムステップの数。予測期間は予測長とも呼ばれます。

新しい予測子を作成するときは、ReferencePredictorArn の値を指定しないでください。

### 予測子のアップグレードと再トレーニング

予測子を再トレーニングまたはアップグレードする場合は、次のパラメータが必要です。

- PredictorName - 予測子の一意の名前。
- ReferencePredictorArn - 再トレーニングまたはアップグレードする予測子の ARN。

予測子をアップグレードまたは再トレーニングするときは、ReferencePredictorArn と PredictorName の値のみを指定します。

### リクエストの構文

```
{
  "DataConfig": {
    "AdditionalDatasets": [
      {
        "Configuration": {
          "string" : [ "string" ]
        }
      },
    ],
  },
}
```

```
    "Name": "string"
  }
],
"AttributeConfigs": [
  {
    "AttributeName": "string",
    "Transformations": {
      "string": "string"
    }
  }
],
"DatasetGroupArn": "string"
},
"EncryptionConfig": {
  "KMSKeyArn": "string",
  "RoleArn": "string"
},
"ExplainPredictor": boolean,
"ForecastDimensions": [ "string" ],
"ForecastFrequency": "string",
"ForecastHorizon": number,
"ForecastTypes": [ "string" ],
"MonitorConfig": {
  "MonitorName": "string"
},
"OptimizationMetric": "string",
"PredictorName": "string",
"ReferencePredictorArn": "string",
"Tags": [
  {
    "Key": "string",
    "Value": "string"
  }
],
"TimeAlignmentBoundary": {
  "DayOfMonth": number,
  "DayOfWeek": "string",
  "Hour": number,
  "Month": "string"
}
}
```

## リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### [DataConfig](#)

データセットグループと追加のデータセットのデータ設定。

タイプ: [DataConfig](#) オブジェクト

必須: いいえ

### [EncryptionConfig](#)

Amazon Forecast がキーにアクセスするために引き受けることができる AWS Key Management Service (KMS) キーと AWS Identity and Access Management (IAM) ロール。このオプションのオブジェクトは、[CreateDataset](#) および [CreatePredictor](#) リクエストで指定できます。

タイプ: [EncryptionConfig](#) オブジェクト

必須: いいえ

### [ExplainPredictor](#)

予測子の Explainability リソースを作成します。

型: ブール値

必須: いいえ

### [ForecastDimensions](#)

生成された予測をグループ化する方法を指定するディメンション (フィールド) 名の配列。

例えば、すべての店舗の商品の販売予測を生成していて、データセットに store\_id フィールドが含まれている場合、各店舗の販売予測をグループ化するディメンションとして store\_id を指定します。

タイプ: 文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 10 項目です。

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: いいえ

## [ForecastFrequency](#)

予測内の予測の頻度。

有効な間隔は、整数の後に Y (年)、M (月)、W (週)、D (日)、H (時)、min (分) をつけたものです。例えば、「1D」は毎日を示し、「15min」は 15 分ごとを示します。次に多頻度で重複するような値は指定できません。つまり、たとえば 60 分の間隔は 1 時間に相当するため、指定することはできません。各頻度の有効な値は以下のとおりです。

- 分 - 1~59
- 時間 - 1~23
- 日 - 1~6
- 週 - 1~4
- 月 - 1~11
- 年 - 1

したがって、隔週の予測が必要な場合は、「2W」と指定します。または、四半期ごとの予測が必要な場合は、「3M」と指定します。

頻度は、TARGET\_TIME\_SERIES データセットの頻度以上である必要があります。

RELATED\_TIME\_SERIES データセットが提供される場合、頻度は RELATED\_TIME\_SERIES データセットの頻度と等しくなければなりません。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 5 です。

Pattern: `^Y|M|W|D|H|30min|15min|10min|5min|1min$`

必須: いいえ

## [ForecastHorizon](#)

モデルが予測するタイムステップの数。予測期間は予測長とも呼ばれます。

最大予測期間は、500 タイムステップまたは TARGET\_TIME\_SERIES のデータセットの長さの 1/4 のいずれかが小さい方です。既存のを再トレーニングする場合 AutoPredictor、最大予測期間は



500 タイムステップまたは TARGET\_TIME\_SERIES データセットの長さの 1/3 のいずれか小さい方です。

にアップグレードする場合、AutoPredictor または既存のを再トレーニングする場合 AutoPredictor、予測期間パラメータを更新することはできません。この要件は、データセットの時系列を長くすることで満たすことができます。

タイプ: 整数

必須: いいえ

### [ForecastTypes](#)

予測子のトレーニングに使用される予測タイプ。最大 5 つの予測タイプを指定できます。予測タイプは、0.01 以上の増分で 0.01 から 0.99 までの分位数にすることができます。mean で平均予測を指定することもできます。

タイプ: 文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 20 項目です。

長さの制限: 最小長は 2 です。最大長は 4 です。

Pattern: (^0?\.|d\d?\$|^mean\$)

必須: いいえ

### [MonitorConfig](#)

予測子モニタリングの構成の詳細。モニターの名前を指定します。

予測子モニタリングにより、予測子のパフォーマンスが時間の経過とともにどのように変化するかを確認できます。詳細については、「[Predictor Monitoring](#)」を参照してください。

タイプ: [MonitorConfig](#) オブジェクト

必須: いいえ

### [OptimizationMetric](#)

予測子を最適化するために使用される精度メトリクス。

型: 文字列

有効な値 : WAPE | RMSE | AverageWeightedQuantileLoss | MASE | MAPE

必須 : いいえ

### PredictorName

予測子の一意の名前

型: 文字列

長さの制限 : 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須 : はい

### ReferencePredictorArn

再トレーニングまたはアップグレードする予測子の ARN。このパラメータは、予測子を再トレーニングまたはアップグレードするときのみ使用されます。新しい予測子を作成するときは、このパラメータの値を指定しないでください。

予測子をアップグレードまたは再トレーニングするときは、ReferencePredictorArn と PredictorName の値のみを指定します。PredictorName の値は、一意の予測子名である必要があります。

型: 文字列

長さの制限 : 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

必須: いいえ

### Tags

予測子の分類と整理に役立つオプションのメタデータ。タグはそれぞれ、1つのキーとオプションの1つの値で設定されており、どちらもお客様側が定義します。タグのキーと値では、大文字と小文字が区別されます。

タグには以下の制限があります。

- タグキーは、リソースごとにそれぞれ一意である必要があります。また、各タグキーに設定できる値は1つのみです。

- リソースあたりのタグの最大数: 50。
- キーの最大長 128 文字 (Unicode) (UTF-8)。
- 値の最大長: 256 文字 (Unicode) (UTF-8)。
- 使用可能な文字: すべての文字と数字、UTF-8 で表現可能なスペース、および + - = 。 \_ : / @ 。タグ付けスキーマが他のサービスやリソースで使用されている場合、それらのサービスの文字制限も適用されます。
- キープレフィックスには、aws: または AWS: の大文字または小文字の組み合わせを含めることはできません。値にはこのプレフィックスを付けることができます。タグ値には aws がプレフィックスとして付されているが、キーには付されていない場合、Forecast は、それをユーザータグとみなし、タグ数の上限である 50 個を計算する際に算入します。aws のキープレフィックスのみを持つタグは、リソース制限あたりのタグに算入されません。このプレフィックスが含まれるタグキーを編集したり削除することはできません。

タイプ: [Tag](#) オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 200 項目です。

必須: いいえ

### [TimeAlignmentBoundary](#)

Forecast が、予想頻度に合わないすべてのデータを調整および集計するために使用する時間境界です。時間単位と時間境界をキーバリューペアとして指定します。時間境界の指定について詳しくは、「[Specifying a Time Boundary](#)」を参照してください。時間境界を指定しない場合、Forecast は[デフォルトの時間境界](#)のセットを使用します。

タイプ: [TimeAlignmentBoundary](#) オブジェクト

必須: いいえ

### レスポンスの構文

```
{
  "PredictorArn": "string"
}
```

### レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### PredictorArn

予測子の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

### エラー

#### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

#### LimitExceededException

アカウントあたりのリソース数の制限を超えました。

HTTP ステータスコード: 400

#### ResourceAlreadyExistsException

この名前のリソースは既に存在します。別の名前で再試行してください。

HTTP ステータスコード: 400

#### ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード: 400

#### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード: 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreateDataset

サービス: Amazon Forecast Service

Amazon Forecast データセットを作成します。提供するデータセットに関する情報は、Forecast がモデルトレーニングのためにデータを使用する方法を理解するのに役立ちます。これには以下が含まれます。

- *DataFrequency* - 履歴時系列データが収集される頻度。
- *Domain* および *DatasetType* - 各データセットには、関連付けられたデータセットドメインとドメイン内のタイプがあります。Amazon Forecast は、各ドメイン内の事前定義されたドメインとタイプのリストを提供します。一意の各データセットドメインとそのドメイン内のタイプについて、Amazon Forecast では、最小限の事前設定されたフィールドセットがデータに含まれている必要があります。
- *Schema* - スキーマは、フィールド名やデータ型など、データセット内のフィールドを指定します。

データセットを作成したら、それにトレーニングデータをインポートし、そのデータセットをデータセットグループに追加します。データセットグループを使用して予測子を作成します。詳細については、「[データセットのインポート](#)」を参照してください。

すべてのデータセットのリストを取得するには、[ListDatasets](#) オペレーションを使用します。

Forecast データセットの例については、「[Amazon Forecast サンプル GitHub リポジトリ](#)」を参照してください。

### Note

データセットの Status は、トレーニングデータをインポートする前に ACTIVE である必要があります。[DescribeDataset](#) オペレーションを使用してステータスを取得します。

### リクエストの構文

```
{
  "DataFrequency": "string",
  "DatasetName": "string",
  "DatasetType": "string",
  "Domain": "string",
  "EncryptionConfig": {
```

```
    "KMSKeyArn": "string",
    "RoleArn": "string"
  },
  "Schema": {
    "Attributes": [
      {
        "AttributeName": "string",
        "AttributeType": "string"
      }
    ]
  },
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

## リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### [DataFrequency](#)

データ収集の頻度。このパラメータは、RELATED\_TIME\_SERIES データセットに必須です。

有効な間隔は、整数の後に Y (年)、M (月)、W (週)、D (日)、H (時)、min (分) をつけたものです。例えば、「1D」は毎日を示し、「15min」は 15 分ごとを示します。次に多頻度で重複するような値は指定できません。つまり、たとえば 60 分の間隔は 1 時間に相当するため、指定することはできません。各頻度の有効な値は以下のとおりです。

- 分 - 1~59
- 時間 - 1~23
- 日 - 1~6
- 週 - 1~4
- 月 - 1~11
- 年 - 1

したがって、隔週の予測が必要な場合は、「2W」と指定します。または、四半期ごとの予測が必要な場合は、「3M」と指定します。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 5 です。

Pattern: `^Y|M|W|D|H|30min|15min|10min|5min|1min$`

必須: いいえ

### DatasetName

データセットの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: はい

### DatasetType

データセットのタイプ。有効な値は、選択した Domain によって異なります。

型: 文字列

有効な値: TARGET\_TIME\_SERIES | RELATED\_TIME\_SERIES | ITEM\_METADATA

必須: はい

### Domain

データセットに関連付けられているドメイン。データセットをデータセットグループに追加する場合、この値と [CreateDatasetGroup](#) オペレーションの Domain パラメータに指定された値が一致する必要があります。

Domain と DatasetType を選択すると、データセットにインポートするトレーニングデータに存在しなければならないフィールドが決まります。例えば、RETAIL ドメインを選択し、DatasetType として TARGET\_TIME\_SERIES を指定した場合、Amazon Forecast では、item\_id、timestamp、および demand フィールドがデータに存在している必要があります。詳細については、「[データセットのインポート](#)」を参照してください。

型: 文字列



有効な値 : RETAIL | CUSTOM | INVENTORY\_PLANNING | EC2\_CAPACITY | WORK\_FORCE  
| WEB\_TRAFFIC | METRICS

必須: はい

## [EncryptionConfig](#)

Amazon Forecast がキーにアクセスするために引き受けることができる AWS Key Management Service (KMS) キーと AWS Identity and Access Management (IAM) ロール。

タイプ : [EncryptionConfig](#) オブジェクト

必須: いいえ

## [Schema](#)

データセットのスキーマ。スキーマ属性とその順序は、データのフィールドと一致する必要があります。データセット Domain と選択した DatasetType によって、トレーニングデータの最小必須フィールドが決まります。特定のデータセットドメインとタイプの必須フィールドについては、「[データセットドメインとデータセットタイプ](#)」を参照してください。

型: [Schema](#) オブジェクト

必須: はい

## [Tags](#)

サービスに適用し、データセットの分類と整理に役立つオプションのメタデータ。タグはそれぞれ、1つのキーとオプションの1つの値で設定されており、どちらもお客様側が定義します。

タグには以下のベーシックな制限があります。

- リソースあたりのタグの最大数は 50 です。
- タグキーは、リソースごとにそれぞれ一意である必要があります。また、各タグキーに設定できる値は 1 つのみです。
- キーの最大長 - 128 文字 (Unicode) (UTF-8)。
- 値の最大長 - 256 文字 (Unicode) (UTF-8)。
- 複数の のサービス間およびリソース間でタグ付けスキーマを使用する場合、他のサービスにも許容される文字数に制限がある可能性があることに注意してください。一般的に使用が許可される文字は、UTF-8 で表現可能な文字、数字、スペース、および以下の文字です:  
+、-、=、..、\_、:、/、@。
- タグのキーと値では、大文字と小文字が区別されます。

- `aws:`、`AWS:`、またはキーのプレフィックスなどの大文字または小文字の組み合わせは使用しないでください。AWS 使用のために予約されています。このプレフィックスが含まれるタグキーを編集したり削除することはできません。値にはこのプレフィックスを付けることができます。タグ値には `aws` がプレフィックスとして付されているが、キーには付されていない場合、Forecast は、それをユーザータグとみなし、タグ数の上限である 50 個を計算する際に算入します。`aws` のキープレフィックスのみを持つタグは、リソース制限あたりのタグに算入されません。

タイプ: [Tag](#) オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 200 項目です。

必須: いいえ

## レスポンスの構文

```
{
  "DatasetArn": "string"
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### [DatasetArn](#)

データセットの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

## エラー

### `InvalidInputException`

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード : 400

LimitExceededException

アカウントあたりのリソース数の制限を超えました。

HTTP ステータスコード : 400

ResourceAlreadyExistsException

この名前のリソースは既に存在します。別の名前で再試行してください。

HTTP ステータスコード : 400

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreateDatasetGroup

サービス: Amazon Forecast Service

関連するデータセットのコレクションを保持するデータセットグループを作成します。データセットグループの作成時にデータセットグループに追加するか、[UpdateDatasetGroup](#)オペレーションを使用して後でデータセットグループに追加できます。

データセットグループを作成してデータセットを追加した後、予測子を作成するときにデータセットグループを使用します。データセットグループの詳細については、「[データセットグループ](#)」を参照してください。

すべてのデータセットグループのリストを取得するには、[ListDatasetGroups](#)オペレーションを使用します。

### Note

データセットグループを使用して予測子を作成する前に、データセットグループの Status が ACTIVE となっている必要があります。ステータスを取得するには、[DescribeDatasetGroup](#)オペレーションを使用します。

### リクエストの構文

```
{
  "DatasetArns": [ "string" ],
  "DatasetGroupName": "string",
  "Domain": "string",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

## DatasetArns

データセットグループに含めるデータセットの Amazon リソースネーム (ARN) の配列。

タイプ: 文字列の配列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

必須: いいえ

## DatasetGroupName

データセットグループの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: はい

## Domain

データセットグループに関連付けられているドメイン。データセットをデータセットグループに追加する場合、この値と [CreateDataset](#) オペレーションの Domain パラメータに指定された値が一致する必要があります。

Domain と DatasetType を選択すると、データセットにインポートするトレーニングデータに存在しなければならないフィールドが決まります。例えば、RETAIL ドメインを選択し、DatasetType として TARGET\_TIME\_SERIES を指定した場合、Amazon Forecast では、item\_id、timestamp、および demand フィールドがデータに存在している必要があります。データセットグループの詳細については、「[データセットグループ](#)」を参照してください。

型: 文字列

有効な値: RETAIL | CUSTOM | INVENTORY\_PLANNING | EC2\_CAPACITY | WORK\_FORCE  
| WEB\_TRAFFIC | METRICS

必須: はい

## Tags

サービスに適用し、データセットグループの分類と整理に役立つオプションのメタデータ。タグはそれぞれ、1つのキーとオプションの1つの値で設定されており、どちらもお客様側が定義します。

タグには以下のベーシックな制限があります。

- リソースあたりのタグの最大数は 50 です。
- タグキーは、リソースごとにそれぞれ一意である必要があります。また、各タグキーに設定できる値は 1 つのみです。
- キーの最大長 - 128 文字 (Unicode) (UTF-8)。
- 値の最大長 - 256 文字 (Unicode) (UTF-8)。
- 複数のサービス間およびリソース間でタグ付けスキーマを使用する場合、他のサービスにも許容される文字数に制限がある可能性があることに注意してください。一般的に使用が許可される文字は、UTF-8 で表現可能な文字、数字、スペース、および以下の文字です：  
+、-、=、.、\_、:、/、@。
- タグのキーと値では、大文字と小文字が区別されます。
- aws:、AWS:、またはキーのプレフィックスなどの大文字または小文字の組み合わせは使用しないでください。AWS 使用のために予約されています。このプレフィックスが含まれるタグキーを編集したり削除することはできません。値にはこのプレフィックスを付けることができます。タグ値には aws がプレフィックスとして付されているが、キーには付されていない場合、Forecast は、それをユーザータグとみなし、タグ数の上限である 50 個を計算する際に算入します。aws のキープレフィックスのみを持つタグは、リソース制限あたりのタグに算入されません。

タイプ: [Tag](#) オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 200 項目です。

必須: いいえ

### レスポンスの構文

```
{  
  "DatasetGroupArn": "string"  
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### DatasetGroupArn

データセットグループの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

### LimitExceededException

アカウントあたりのリソース数の制限を超えました。

HTTP ステータスコード: 400

### ResourceAlreadyExistsException

この名前のリソースは既に存在します。別の名前で再試行してください。

HTTP ステータスコード: 400

### ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード: 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)



## CreateDatasetImportJob

サービス: Amazon Forecast Service

トレーニングデータを Amazon Forecast のデータセットにインポートします。Amazon Simple Storage Service (Amazon S3) バケット内のトレーニングデータの場所と、データのインポート先とするデータセットの Amazon リソースネーム (ARN) を指定します。

Amazon Forecast がデータのコピーを作成し、内部 AWS システムで処理するため、Amazon Forecast がデータにアクセスするために引き受けることができる AWS Identity and Access Management (IAM) ロールを含む [DataSource](#) オブジェクトを指定する必要があります。詳細については、「[Set up permissions](#)」を参照してください。

トレーニングデータは CSV または Parquet 形式である必要があります。区切り文字はカンマ (,) にしてください。

特定のファイル、S3 バケット、または S3 バケット内のフォルダへのパスを指定できます。後者の 2 つのケースでは、Amazon Forecast は、上限である 10,000 個まですべてのファイルをインポートします。

データセットのインポートは集計されないため、最新のデータセットインポートは、予想子のトレーニング時や予測の生成時に使用されるものです。最新のデータセットインポートに、前回のインポート以降に収集された新しいデータだけでなく、モデル化するすべてのデータが含まれていることを確認します。

指定された基準でフィルタリングされたすべてのデータセットのインポートジョブのリストを取得するには、[ListDatasetImportJobs](#) オペレーションを使用します。

リクエストの構文

```
{
  "DatasetArn": "string",
  "DatasetImportJobName": "string",
  "DataSource": {
    "S3Config": {
      "KMSKeyArn": "string",
      "Path": "string",
      "RoleArn": "string"
    }
  },
  "Format": "string",
  "GeolocationFormat": "string",
  "ImportMode": "string",
```

```
"Tags": [  
  {  
    "Key": "string",  
    "Value": "string"  
  }  
],  
"TimestampFormat": "string",  
"TimeZone": "string",  
"UseGeolocationForTimeZone": boolean  
}
```

## リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### DatasetArn

データをインポートする Amazon Forecast データセットの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

### DatasetImportJobName

データセットのインポートジョブの名前。名前に現在のタイムスタンプを含めることをお勧めします (例: 20190721DatasetImport)。これは、ResourceAlreadyExistsException 例外の発生を回避するのに役立ちます。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: はい

### DataSource

インポートするトレーニングデータの場所と、Amazon Forecast がデータにアクセスするために引き受けることができる AWS Identity and Access Management (IAM) ロール。トレーニングデータは Amazon S3 バケットに保存されている必要があります。

暗号化を使用する場合は、に AWS Key Management Service (KMS) キーを含める DataSource 必要があり、IAM ロールは Amazon Forecast にキーへのアクセス許可を付与する必要があります。KMS キーと IAM ロールは、[CreateDataset](#) オペレーションの EncryptionConfig パラメータで指定されたものと一致する必要があります。

型: [DataSource](#) オブジェクト

必須: はい

### [Format](#)

インポートされたデータの形式は CSV または PARQUET のいずれかです。デフォルト値は「CSV」です。

型: 文字列

長さの制限: 最大長は 7 です。

Pattern: ^CSV|PARQUET\$

必須: いいえ

### [GeolocationFormat](#)

geolocation 属性の形式。geolocation 属性は、次の 2 つの方法のいずれかでフォーマットできます。

- LAT\_LONG - 10 進数のフォーマットの緯度と経度 (例: 47.61\_-122.33)。
- CC\_POSTALCODE (米国のみ) - 国コード (US) の後に 5 桁の郵便番号が続きます (例: US\_98121)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: ^[a-zA-Z0-9\_]+\$

必須: いいえ

### [ImportMode](#)

データセットのインポートジョブが FULL インポートか INCREMENTAL インポートかを指定します。FULL データセットのインポートは、既存のデータをすべて新しくインポートされたデータ

に置き換えます。INCREMENTAL インポートでは、インポートされたデータが既存のデータに追加されます。

型: 文字列

有効な値 : FULL | INCREMENTAL

必須 : いいえ

## Tags

サービスに適用し、データセットのインポートジョブの分類と整理に役立つオプションのメタデータ。タグはそれぞれ、1つのキーとオプションの1つの値で設定されており、どちらもお客様側が定義します。

タグには以下のベーシックな制限があります。

- リソースあたりのタグの最大数は 50 です。
- タグキーは、リソースごとにそれぞれ一意である必要があります。また、各タグキーに設定できる値は 1 つのみです。
- キーの最大長 - 128 文字 (Unicode) (UTF-8)。
- 値の最大長 - 256 文字 (Unicode) (UTF-8)。
- 複数の のサービス間およびリソース間でタグ付けスキーマを使用する場合、他のサービスにも許容される文字数に制限がある可能性があることに注意してください。一般的に使用が許可される文字は、UTF-8 で表現可能な文字、数字、スペース、および以下の文字です:  
+、-、=、.、\_、:、/、@。
- タグのキーと値では、大文字と小文字が区別されます。
- aws:、AWS:、またはキーのプレフィックスなどの大文字または小文字の組み合わせは使用しないでください。AWS 使用のために予約されています。このプレフィックスが含まれるタグキーを編集したり削除することはできません。値にはこのプレフィックスを付けることができます。タグ値には aws がプレフィックスとして付されているが、キーには付されていない場合、Forecast は、それをユーザータグとみなし、タグ数の上限である 50 個を計算する際に算入します。aws のキープレフィックスのみを持つタグは、リソース制限あたりのタグに算入されません。

タイプ : [Tag](#) オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 200 項目です。

必須: いいえ

## TimestampFormat

データセット内のタイムスタンプの形式。指定する形式は、データセットが作成されたときに指定された DataFrequency によって異なります。以下の形式がサポートされています

- 「yyyy-MM-dd」

次のデータ周波数の場合: Y、M、W、D

- 「yyyy-MM-dd HH:mm:ss」

次のデータ頻度の場合: H、30min、15min、および 1min。オプションで、Y、M、W、および D

フォーマットが指定されていない場合、Amazon Forecast はフォーマットが「yyyy-MM-dd HH:mm:ss」であると想定します。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `^[a-zA-Z0-9\-\:\.\,\'\s]+$`

必須: いいえ

## TimeZone

データセット内のすべての項目についての単一のタイムゾーン。このオプションは、すべてのタイムスタンプが単一のタイムゾーン内にあるデータセット、またはすべてのタイムスタンプが単一のタイムゾーンに正規化されているデータセットに最適です。

有効なタイムゾーン名の完全なリストについては、「[Joda-Time API](#)」を参照してください。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `^[a-zA-Z0-9\+\-\_\s]+$`

必須: いいえ

## UseGeolocationForTimeZone

geolocation 属性からタイムゾーンの情報自動的に取得します。このオプションは、複数のタイムゾーンのタイムスタンプを含み、それらのタイムスタンプが現地時間で表されるデータセットに最適です。

型: ブール値

必須: いいえ

## レスポンスの構文

```
{  
  "DatasetImportJobArn": "string"  
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### [DatasetImportJobArn](#)

データセットのインポートジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

### LimitExceededException

アカウントあたりのリソース数の制限を超えました。

HTTP ステータスコード: 400

### ResourceAlreadyExistsException

この名前のリソースは既に存在します。別の名前で再試行してください。

HTTP ステータスコード : 400

#### ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

#### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreateExplainability

サービス: Amazon Forecast Service

### Note

Explainability は、AutoPredictor ( [CreateAutoPredictor](#) ) から生成された Forecasts と Predictor でのみ使用できます。

Amazon Forecast Explainability を作成します。

Explainability は、データセットの属性が予測にどのように影響するかをよりよく理解するのに役立ちます。Amazon Forecast は、Impact スコアと呼ばれるメトリクスを使用して、各属性の相対的な影響を定量化し、それらが予測値を増加させるか減少させるかを決定します。

Forecast Explainability を有効にするには、関連する時系列、項目メタデータ、または祝祭日や Weather Index などの追加のデータセットの少なくとも 1 つを予測子に含める必要があります。

CreateExplainability は、予測子 ARN または予測 ARN のいずれかを受け入れます。データセット内のすべての時系列と時間ポイントの集計された Impact スコアを受け取るには、予測子 ARN を入力します。特定の時系列および時間ポイントの Impact スコアを受け取るには、予測 ARN を入力します。

CreateExplainability 予測子 ARN を使用する

### Note

予測子ごとに持つことができる Explainability リソースは 1 つだけです。 [CreateAutoPredictor](#) で ExplainPredictor を既に有効にしている場合、その予測子には既に Explainability リソースがあります。

Predictor ARN を入力する場合は、次のパラメータが必要です。

- ExplainabilityName - Explainability の一意の名前。
- ResourceArn - 予測子の Arn。
- TimePointGranularity - 「ALL」に設定する必要があります。
- TimeSeriesGranularity - 「ALL」に設定する必要があります。



次のパラメータには値を指定しないでください。

- DataSource - TimeSeriesGranularity が「SPECIFIC」の場合にのみ有効です。
- Schema - TimeSeriesGranularity が「SPECIFIC」の場合にのみ有効です。
- StartDateTime - TimePointGranularity が「SPECIFIC」の場合にのみ有効です。
- EndDateTime - TimePointGranularity が「SPECIFIC」の場合にのみ有効です。

CreateExplainability Forecast ARN を使用した

#### Note

最大 50 の時系列と 500 の時間ポイントを指定できます。

Predictor ARN を入力する場合は、次のパラメータが必要です。

- ExplainabilityName - Explainability の一意の名前。
- ResourceArn - 予測の Arn。
- TimePointGranularity - 「ALL」または「SPECIFIC」のいずれか。
- TimeSeriesGranularity - 「ALL」または「SPECIFIC」のいずれか。

TimeSeriesGranularity を「SPECIFIC」に設定する場合は、以下も指定する必要があります。

- DataSource - 時系列を指定する CSV ファイルの S3 の場所。
- Schema - スキーマは、データソースにリストされている属性と属性タイプを定義します。

TimePointGranularity を「SPECIFIC」に設定する場合は、以下も指定する必要があります。

- StartDateTime - 時間ポイントの範囲の最初のタイムスタンプ。
- EndDateTime - 時間ポイントの範囲の最後のタイムスタンプ。

リクエストの構文

```
{  
  "DataSource": {  
    "S3Config": {
```

```
    "KMSKeyArn": "string",
    "Path": "string",
    "RoleArn": "string"
  }
},
"EnableVisualization": boolean,
"EndTime": "string",
"ExplainabilityConfig": {
  "TimePointGranularity": "string",
  "TimeSeriesGranularity": "string"
},
"ExplainabilityName": "string",
"ResourceArn": "string",
"Schema": {
  "Attributes": [
    {
      "AttributeName": "string",
      "AttributeType": "string"
    }
  ]
},
"StartDate": "string",
"Tags": [
  {
    "Key": "string",
    "Value": "string"
  }
]
}
```

## リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### [DataSource](#)

データのソース、Amazon Forecast がデータにアクセスできるようにする AWS Identity and Access Management (IAM) ロール、およびオプションで AWS Key Management Service (KMS) キー。

タイプ : [DataSource](#) オブジェクト

必須: いいえ

## EnableVisualization

AWS コンソール内で表示可能な Explainability ビジュアライゼーションを作成します。

型: ブール値

必須: いいえ

## EndTime

TimePointGranularity が SPECIFIC に設定されている場合、Explainability の最後の時間ポイントを定義します。

次のタイムスタンプ形式を使用します: yyyy-MM-ddTHH:mm:ss (例: 2015-01-01T20:00:00)

型: 文字列

長さの制限: 最大長は 19。

Pattern: `^\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}$`

必須: いいえ

## ExplainabilityConfig

Explainability の時系列と時間ポイントの粒度を定義する構成設定。

型: [ExplainabilityConfig](#) オブジェクト

必須: はい

## ExplainabilityName

Explainability の一意の名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: はい

## ResourceArn

Explainability の作成に使用される予測子または予測の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限：最大長は 256 です。

パターン：`arn:([a-z\d-]+):forecast:.*:.*:.*+`

必須：はい

## Schema

データセットのフィールドを定義します。

タイプ：[Schema](#) オブジェクト

必須: いいえ

## StartTime

TimePointGranularity が SPECIFIC に設定されている場合、Explainability の最初のポイントを定義します。

次のタイムスタンプ形式を使用します: yyyy-MM-ddTHH:mm:ss (例: 2015-01-01T20:00:00)

型: 文字列

長さの制限: 最大長は 19。

Pattern: `^\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}$`

必須: いいえ

## Tags

リソースの分類と整理に役立つオプションのメタデータ。タグはそれぞれ、1つのキーとオプションの1つの値で設定されており、どちらもお客様側が定義します。タグのキーと値では、大文字と小文字が区別されます。

タグには以下の制限があります。

- タグキーは、リソースごとにそれぞれ一意である必要があります。また、各タグキーに設定できる値は1つのみです。
- リソースあたりのタグの最大数: 50。
- キーの最大長: 128 文字 (Unicode) (UTF-8)。
- 値の最大長: 256 文字 (Unicode) (UTF-8)。
- 使用可能な文字: すべての文字と数字、UTF-8 で表現可能なスペース、および + - = . \_ : / @。タグ付けスキーマが他のサービスやリソースで使用されている場合、それらのサービスの文字制限も適用されます。

- キープレフィックスには、aws: または AWS: の大文字または小文字の組み合わせを含めることはできません。値にはこのプレフィックスを付けることができます。タグ値には aws がプレフィックスとして付されているが、キーには付されていない場合、Forecast は、それをユーザータグとみなし、タグ数の上限である 50 個を計算する際に算入します。aws のキープレフィックスのみを持つタグは、リソース制限あたりのタグに算入されません。このプレフィックスが含まれるタグキーを編集したり削除することはできません。

タイプ: [Tag](#) オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 200 項目です。

必須: いいえ

## レスポンスの構文

```
{
  "ExplainabilityArn": "string"
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### [ExplainabilityArn](#)

Explainability の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード : 400

LimitExceededException

アカウントあたりのリソース数の制限を超えました。

HTTP ステータスコード : 400

ResourceAlreadyExistsException

この名前のリソースは既に存在します。別の名前で再試行してください。

HTTP ステータスコード : 400

ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)



## CreateExplainabilityExport

サービス: Amazon Forecast Service

[CreateExplainability](#) オペレーションによって作成された Explainability リソースをエクスポートします。エクスポートされたファイルは、Amazon Simple Storage Service (Amazon S3) バケットにエクスポートされます。

Amazon S3 バケットと、Amazon Forecast が Amazon S3 バケットにアクセスするために引き受けることができる AWS Identity and Access Management (IAM) ロールを含む [DataDestination](#) オブジェクトを指定する必要があります。Amazon S3 詳細については、「[Amazon Forecast のアクセス許可を設定する](#)」を参照してください。

### Note

Amazon S3 バケットのエクスポートにアクセスする前に、エクスポートジョブの Status が ACTIVE となっている必要があります。[DescribeExplainabilityExport](#) オペレーションを使用して、ステータスを取得します。

### リクエストの構文

```
{
  "Destination": {
    "S3Config": {
      "KMSKeyArn": "string",
      "Path": "string",
      "RoleArn": "string"
    }
  },
  "ExplainabilityArn": "string",
  "ExplainabilityExportName": "string",
  "Format": "string",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```



## リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### Destination

エクスポートジョブの宛先。S3 バス、Amazon Forecast がロケーションにアクセスできるようにする AWS Identity and Access Management (IAM) ロール、および AWS Key Management Service (KMS) キー (オプション) を指定します。

型: [DataDestination](#) オブジェクト

必須: はい

### ExplainabilityArn

エクスポートする Explainability の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

### ExplainabilityExportName

Explainability エクスポートの一意の名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: はい

### Format

エクスポートされたデータの形式は CSV または PARQUET のいずれかです。

型: 文字列

長さの制限: 最大長は 7 です。

Pattern: `^(CSV|PARQUET)$`

必須: いいえ

## Tags

リソースの分類と整理に役立つオプションのメタデータ。タグはそれぞれ、1つのキーとオプションの1つの値で設定されており、どちらもお客様側が定義します。タグのキーと値では、大文字と小文字が区別されます。

タグには以下の制限があります。

- タグキーは、リソースごとにそれぞれ一意である必要があります。また、各タグキーに設定できる値は1つのみです。
- リソースあたりのタグの最大数: 50。
- キーの最大長: 128 文字 (Unicode) (UTF-8)。
- 値の最大長: 256 文字 (Unicode) (UTF-8)。
- 使用可能な文字: すべての文字と数字、UTF-8 で表現可能なスペース、および + - = . \_ : / @。タグ付けスキーマが他のサービスやリソースで使用されている場合、それらのサービスの文字制限も適用されます。
- キープレフィックスには、aws: または AWS: の大文字または小文字の組み合わせを含めることはできません。値にはこのプレフィックスを付けることができます。タグ値には aws がプレフィックスとして付されているが、キーには付されていない場合、Forecast は、それをユーザータグとみなし、タグ数の上限である 50 個を計算する際に算入します。aws のキープレフィックスのみを持つタグは、リソース制限あたりのタグに算入されません。このプレフィックスが含まれるタグキーを編集したり削除することはできません。

タイプ: [Tag](#) オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 200 項目です。

必須: いいえ

## レスポンスの構文

```
{  
  "ExplainabilityExportArn": "string"  
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### [ExplainabilityExportArn](#)

エクスポートの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

### LimitExceededException

アカウントあたりのリソース数の制限を超えました。

HTTP ステータスコード: 400

### ResourceAlreadyExistsException

この名前のリソースは既に存在します。別の名前で再試行してください。

HTTP ステータスコード: 400

### ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード: 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード: 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreateForecast

サービス: Amazon Forecast Service

予測子のトレーニングに使用された TARGET\_TIME\_SERIES データセット内の各アイテムの予測を作成します。これは、推論として知られています。低レイテンシーで単一の項目の予測を取得するには、[QueryForecast](#) オペレーションを使用します。完全な予測を Amazon Simple Storage Service (Amazon S3) バケットにエクスポートするには、[CreateForecastExportJob](#) オペレーションを使用します。

予測の範囲は、[CreatePredictor](#) リクエストで指定した ForecastHorizon の値によって決まります。予測をクエリするときに、予測内の特定の日付範囲をリクエストできます。

すべての予測のリストを取得するには、[ListForecasts](#) オペレーションを使用します。

### Note

Amazon Forecast によって生成された予測は、予測子の作成に使用されたデータセットと同じタイムゾーンにあります。

詳細については、「[予測の生成](#)」を参照してください。

### Note

予測をクエリまたはエクスポートする前に、予測の Status が ACTIVE となっている必要があります。[DescribeForecast](#) オペレーションを使用して、ステータスを取得します。

デフォルトでは、予測子のトレーニングに使用されたデータセットグループ内の各項目 (item\_id) の予想が含まれます。ただし、TimeSeriesSelector オブジェクトを使用して時系列のサブセットに関する予想を生成することはできます。入力データセットに含まれていない指定の時系列では、予想の作成はスキップされます。予想エクスポートファイルには、これらの時系列や予測値は含まれません。

リクエストの構文

```
{
  "ForecastName": "string",
  "ForecastTypes": [ "string" ],
```

```
"PredictorArn": "string",
"Tags": [
  {
    "Key": "string",
    "Value": "string"
  }
],
"TimeSeriesSelector": {
  "TimeSeriesIdentifiers": {
    "DataSource": {
      "S3Config": {
        "KMSKeyArn": "string",
        "Path": "string",
        "RoleArn": "string"
      }
    },
    "Format": "string",
    "Schema": {
      "Attributes": [
        {
          "AttributeName": "string",
          "AttributeType": "string"
        }
      ]
    }
  }
}
```

## リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### ForecastName

予測の名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: はい

## ForecastTypes

確率的予測が生成される分位数。現在、予測ごとに最大 5 つの分位数を指定できます。使用可能な値には、0.01 to 0.99 (.01 の増分のみ) および mean が含まれます。分布が対称でない場合 (例えば、ベータおよび負の二項)、平均予測は中央値 (0.50) とは異なります。

デフォルトの分位数は、予測子の作成時に指定した分位数です。分位数を指定しなかった場合、デフォルト値は ["0.1", "0.5", "0.9"] です。

タイプ: 文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 20 項目です。

長さの制限: 最小長は 2 です。最大長は 4 です。

Pattern: (^0?\.?\d\d?\$|^mean\$)

必須: いいえ

## PredictorArn

予測の生成に使用する予測子の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: arn:([a-z\d-]+):forecast:.\*:.\*:.\*+

必須: はい

## Tags

サービスに適用し、予測の分類と整理に役立つオプションのメタデータ。タグはそれぞれ、1 つのキーとオプションの 1 つの値で設定されており、どちらもお客様側が定義します。

タグには以下のベーシックな制限があります。

- リソースあたりのタグの最大数は 50 です。
- タグキーは、リソースごとにそれぞれ一意である必要があります。また、各タグキーに設定できる値は 1 つのみです。
- キーの最大長 - 128 文字 (Unicode) (UTF-8)。
- 値の最大長 - 256 文字 (Unicode) (UTF-8)。

- 複数の のサービス間およびリソース間でタグ付けスキーマを使用する場合、他のサービスにも許容される文字数に制限がある可能性があることに注意してください。一般的に使用が許可される文字は、UTF-8 で表現可能な文字、数字、スペース、および以下の文字です：  
+、-、=、.、\_、:、/、@。
- タグのキーと値では、大文字と小文字が区別されます。
- aws:、AWS:、またはキーのプレフィックスなどの大文字または小文字の組み合わせは使用しないでください。AWS 使用のために予約されています。このプレフィックスが含まれるタグキーを編集したり削除することはできません。値にはこのプレフィックスを付けることができます。タグ値には aws がプレフィックスとして付されているが、キーには付されていない場合、Forecast は、それをユーザータグとみなし、タグ数の上限である 50 個を計算する際に算入します。aws のキープレフィックスのみを持つタグは、リソース制限あたりのタグに算入されません。

タイプ : [Tag](#) オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 200 項目です。

必須: いいえ

## [TimeSeriesSelector](#)

TimeSeriesIdentifiers オブジェクト内の予想を作成するために使用される時系列のセットを定義します。

TimeSeriesIdentifiers オブジェクトには次の情報が必要です。

- DataSource
- Format
- Schema

タイプ : [TimeSeriesSelector](#) オブジェクト

必須: いいえ

## レスポンスの構文

```
{  
  "ForecastArn": "string"  
}
```



## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### ForecastArn

予測の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

### LimitExceededException

アカウントあたりのリソース数の制限を超えました。

HTTP ステータスコード: 400

### ResourceAlreadyExistsException

この名前のリソースは既に存在します。別の名前で再試行してください。

HTTP ステータスコード: 400

### ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード: 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreateForecastExportJob

サービス: Amazon Forecast Service

[CreateForecast](#) オペレーションによって作成された予測を Amazon Simple Storage Service (Amazon S3) バケットにエクスポートします。予測ファイル名は、次の規則に一致します。

<ForecastExportJobName>\_<ExportTimestamp>\_<PartNumber>

ここで、<ExportTimestamp> コンポーネントは Java SimpleDateFormat (yyyy-MM-ddTHH -mm-ssZ) にあります。

Amazon Forecast が Amazon S3 バケットにアクセスするために引き受けることができる AWS Identity and Access Management (IAM) ロールを含む [DataDestination](#) オブジェクトを指定する必要があります。詳細については、「[Amazon Forecast のアクセス許可を設定する](#)」を参照してください。

詳細については、「[予測の生成](#)」を参照してください。

すべての予測エクスポートジョブのリストを取得するには、[ListForecastExportJobs](#) オペレーションを使用します。

### Note

Amazon S3 バケットの予測にアクセスする前に、予測エクスポートジョブの Status が ACTIVE となっている必要があります。[DescribeForecastExportJob](#) オペレーションを使用して、ステータスを取得します。

### リクエストの構文

```
{
  "Destination": {
    "S3Config": {
      "KMSKeyArn": "string",
      "Path": "string",
      "RoleArn": "string"
    }
  },
  "ForecastArn": "string",
  "ForecastExportJobName": "string",
  "Format": "string",
```

```
"Tags": [  
  {  
    "Key": "string",  
    "Value": "string"  
  }  
]  
}
```

## リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### Destination

予測を保存する場所と、Amazon Forecast がその場所にアクセスするために引き受けることができる AWS Identity and Access Management (IAM) ロール。予測は Amazon S3 バケットにエクスポートする必要があります。

暗号化を使用する場合は、に AWS Key Management Service (KMS) キーを含める Destination 必要があります。IAM ロールは、Amazon Forecast がキーにアクセスすることを許可する必要があります。

型: [DataDestination](#) オブジェクト

必須: はい

### ForecastArn

エクスポートする予測の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

### ForecastExportJobName

予測エクスポートジョブの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: はい

## Format

エクスポートされたデータの形式は CSV または PARQUET のいずれかです。デフォルト値は「CSV」です。

型: 文字列

長さの制限: 最大長は 7 です。

Pattern: `^CSV|PARQUET$`

必須: いいえ

## Tags

サービスに適用し、予測のエクスポートジョブの分類と整理に役立つオプションのメタデータ。タグはそれぞれ、1つのキーとオプションの1つの値で設定されており、どちらもお客様側が定義します。

タグには以下のベーシックな制限があります。

- リソースあたりのタグの最大数は 50 です。
- タグキーは、リソースごとにそれぞれ一意である必要があります。また、各タグキーに設定できる値は 1 つのみです。
- キーの最大長 - 128 文字 (Unicode) (UTF-8)。
- 値の最大長 - 256 文字 (Unicode) (UTF-8)。
- 複数の のサービス間およびリソース間でタグ付けスキーマを使用する場合、他のサービスにも許容される文字数に制限がある可能性があることに注意してください。一般的に使用が許可される文字は、UTF-8 で表現可能な文字、数字、スペース、および以下の文字です: `+、-、=、.、_、:、/、@`。
- タグのキーと値では、大文字と小文字が区別されます。
- `aws:`、`AWS:`、またはキーのプレフィックスなどの大文字または小文字の組み合わせは使用しないでください。AWS 使用のために予約されています。このプレフィックスが含まれるタグキーを編集したり削除することはできません。値にはこのプレフィックスを付けることができます。タグ値には `aws` がプレフィックスとして付されているが、キーには付されていない場合、Forecast は、それをユーザータグとみなし、タグ数の上限である 50 個を計算する際に算

入します。aws のキープレフィックスのみを持つタグは、リソース制限あたりのタグに算入されません。

タイプ: [Tag](#) オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 200 項目です。

必須: いいえ

## レスポンスの構文

```
{
  "ForecastExportJobArn": "string"
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### [ForecastExportJobArn](#)

エクスポートジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

### LimitExceededException

アカウントあたりのリソース数の制限を超えました。

HTTP ステータスコード : 400

ResourceAlreadyExistsException

この名前のリソースは既に存在します。別の名前で再試行してください。

HTTP ステータスコード : 400

ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreateMonitor

サービス: Amazon Forecast Service

既存の自動予測子の予測モニターリソースを作成します。予測子モニタリングにより、予測子のパフォーマンスが時間の経過とともにどのように変化するかを確認できます。詳細については、「[Predictor Monitoring](#)」を参照してください。

### リクエストの構文

```
{
  "MonitorName": "string",
  "ResourceArn": "string",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### MonitorName

モニターリソースの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: はい

#### ResourceArn

監視する予測子の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。



パターン : `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須 : はい

## Tags

モニターリソースに適用する [タグ](#) のリスト。

タイプ : [Tag](#) オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 200 項目です。

必須: いいえ

## レスポンスの構文

```
{
  "MonitorArn": "string"
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

## [MonitorArn](#)

モニターリソースの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限 : 最大長は 256 です。

パターン : `arn:([a-z\d-]+):forecast:.*:.*:.*`

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード : 400

## LimitExceededException

アカウントあたりのリソース数の制限を超えました。

HTTP ステータスコード : 400

## ResourceAlreadyExistsException

この名前のリソースは既に存在します。別の名前で再試行してください。

HTTP ステータスコード : 400

## ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

## ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreatePredictor

サービス: Amazon Forecast Service

### Note

このオペレーションにより、Amazon Forecast が提供するすべての予測子機能が含まれていないレガシー予測子が作成されます。Forecast のすべての側面と互換性のある予測子を作成するには、[CreateAutoPredictor](#) を使用します。

Amazon Forecast 予測子を作成します。

リクエストで、データセットグループを指定し、アルゴリズムを指定するか、AutoML を使用して Amazon Forecast にアルゴリズムを選択させます。アルゴリズムを指定すると、アルゴリズム固有のハイパーパラメータを上書きすることもできます。

Amazon Forecast は、アルゴリズムを使用して、指定されたデータセットグループ内のデータセットの最新バージョンを使用して予測子をトレーニングします。その後、[CreateForecast](#) オペレーションを使用して予測を生成できます。

評価メトリクスを表示するには、[GetAccuracyMetrics](#) オペレーションを使用します。

モデルトレーニングを改善するために、TARGET\_TIME\_SERIES データセットのデータフィールドに入力して集計する特徴量化の設定を指定できます。詳細については、「[FeaturizationConfig](#)」を参照してください。

RELATED\_TIME\_SERIES データセットについては、CreatePredictor は、データセットの作成時に DataFrequency が指定したものが ForecastFrequency に一致することを検証します。TARGET\_TIME\_SERIES データセットには、この制限はありません。Amazon Forecast は、区切り文字とタイムスタンプの形式も検証します。詳細については、「[データセットのインポート](#)」を参照してください。

デフォルトでは、予測子は 0.1 (P10)、0.5 (P50)、および 0.9 (P90) 分位数でトレーニングおよび評価されます。ForecastTypes を設定することにより、予測子をトレーニングおよび評価するためのカスタム予測タイプを選択できます。

### AutoML

Amazon Forecast で各アルゴリズムを評価し、objective function を最小化するアルゴリズムを選択する場合は、PerformAutoML を true に設定します。objective function は、予測タイプに対する重み付けされた損失の平均として定義されます。デフォルトでは、これらは

p10、p50、および p90 分位数のロスです。詳細については、「[EvaluationResult](#)」を参照してください。

AutoML が有効になっている場合、次のプロパティは許可されません。

- AlgorithmArn
- HPOConfig
- PerformHPO
- TrainingParameters

すべての予測子のリストを取得するには、[ListPredictors](#) オペレーションを使用します。

#### Note

予測子を使用して予測を作成する前に、予測子の Status が ACTIVE になっている必要があります。これは、トレーニングが完了したことを意味します。[DescribePredictor](#) オペレーションを使用して、ステータスを取得します。

## リクエストの構文

```
{
  "AlgorithmArn": "string",
  "AutoMLOverrideStrategy": "string",
  "EncryptionConfig": {
    "KMSKeyArn": "string",
    "RoleArn": "string"
  },
  "EvaluationParameters": {
    "BackTestWindowOffset": number,
    "NumberOfBacktestWindows": number
  },
  "FeaturizationConfig": {
    "Featurizations": [
      {
        "AttributeName": "string",
        "FeaturizationPipeline": [
          {
            "FeaturizationMethodName": "string",
            "FeaturizationMethodParameters": {
```

```

        "string" : "string"
      }
    }
  ]
},
"ForecastDimensions": [ "string" ],
"ForecastFrequency": "string"
},
"ForecastHorizon": number,
"ForecastTypes": [ "string" ],
"HPOConfig": {
  "ParameterRanges": {
    "CategoricalParameterRanges": [
      {
        "Name": "string",
        "Values": [ "string" ]
      }
    ],
    "ContinuousParameterRanges": [
      {
        "MaxValue": number,
        "MinValue": number,
        "Name": "string",
        "ScalingType": "string"
      }
    ],
    "IntegerParameterRanges": [
      {
        "MaxValue": number,
        "MinValue": number,
        "Name": "string",
        "ScalingType": "string"
      }
    ]
  }
},
"InputDataConfig": {
  "DatasetGroupArn": "string",
  "SupplementaryFeatures": [
    {
      "Name": "string",
      "Value": "string"
    }
  ]
}

```

```
    ]
  },
  "OptimizationMetric": "string",
  "PerformAutoML": boolean,
  "PerformHPO": boolean,
  "PredictorName": "string",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ],
  "TrainingParameters": {
    "string" : "string"
  }
}
```

## リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### AlgorithmArn

モデルトレーニングに使用するアルゴリズムの Amazon リソースネーム (ARN)。PerformAutoML が true に設定されていない場合は必須です。

サポートされているアルゴリズム:

- arn:aws:forecast:::algorithm/ARIMA
- arn:aws:forecast:::algorithm/CNN-QR
- arn:aws:forecast:::algorithm/Deep\_AR\_Plus
- arn:aws:forecast:::algorithm/ETS
- arn:aws:forecast:::algorithm/NPTS
- arn:aws:forecast:::algorithm/Prophet

型: 文字列

長さの制限: 最大長は 256 です。

パターン: arn:([a-z\d-]+):forecast:.\*:.\*:.\*+

必須: いいえ

## [AutoMLOverrideStrategy](#)

### Note

LatencyOptimized AutoML 上書き戦略は、プライベートベータでのみ使用できます。アクセス権限の詳細については、AWS サポートまたはアカウントマネージャーにお問い合わせください。

デフォルトの AutoML 戦略を上書きするために使用されます。これは、予測子の精度を最適化するために行われます。トレーニング時間を最小限に抑える AutoML 戦略を適用するには、LatencyOptimized を使用します。

このパラメータは、AutoML を使用してトレーニングした予測子についてのみ有効です。

型: 文字列

有効な値: LatencyOptimized | AccuracyOptimized

必須: いいえ

## [EncryptionConfig](#)

Amazon Forecast がキーにアクセスするために引き受けることができる AWS Key Management Service (KMS) キーと AWS Identity and Access Management (IAM) ロール。

タイプ: [EncryptionConfig](#) オブジェクト

必須: いいえ

## [EvaluationParameters](#)

指定されたアルゴリズムのデフォルトの評価パラメータを上書きするために使用されます。Amazon Forecast は、データセットをトレーニングデータとテストデータに分割することで予測子を評価します。評価パラメータは、分割の実行方法と反復回数を定義します。

タイプ: [EvaluationParameters](#) オブジェクト

必須: いいえ

## [FeaturizationConfig](#)

特徴量化の設定。

型: [FeaturizationConfig](#) オブジェクト

必須: はい

## [ForecastHorizon](#)

モデルが予測するようにトレーニングされている時間ステップの数を指定します。予測期間は予測長とも呼ばれます。

例えば、([CreateDataset](#) オペレーションの `DataFrequency` パラメータを使用して) 毎日のデータ収集用にデータセットを設定し、予測期間を 10 に設定すると、モデルは 10 日間の予測を返します。

最大予測期間は、500 タイムステップまたは `TARGET_TIME_SERIES` のデータセットの長さの 1/3 のいずれか小さい方です。

タイプ: 整数

必須: はい

## [ForecastTypes](#)

予測子のトレーニングに使用される予測タイプを指定します。最大 5 つの予測タイプを指定できます。予測タイプは、0.01 以上の増分で 0.01 から 0.99 までの分位数にすることができます。mean で平均予測を指定することもできます。

デフォルト値は ["0.10", "0.50", "0.9"] です。

タイプ: 文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 20 項目です。

長さの制限: 最小長は 2 です。最大長は 4 です。

Pattern: (^0?\.\d\d?|^mean\$)

必須: いいえ

## [HPOConfig](#)

アルゴリズム用にハイパーパラメータの上書き値を提供します。このパラメータを指定しない場合、Amazon Forecast はデフォルト値を使用します。個々のアルゴリズムは、ハイパーパラメータ最適化 (HPO) をサポートするハイパーパラメータを指定します。詳細については、「[Amazon Forecast アルゴリズム](#)」を参照してください。

HPOConfig オブジェクトを含めた場合は、`PerformHPO` を true に設定する必要があります。



タイプ: [HyperParameterTuningJobConfig](#) オブジェクト

必須: いいえ

### [InputDataConfig](#)

予測子のトレーニングに使用するデータを含むデータセットグループを記述します。

型: [InputDataConfig](#) オブジェクト

必須: はい

### [OptimizationMetric](#)

予測子を最適化するために使用される精度メトリクス。デフォルト値は `AverageWeightedQuantileLoss` です。

型: 文字列

有効な値: `WAPE` | `RMSE` | `AverageWeightedQuantileLoss` | `MASE` | `MAPE`

必須: いいえ

### [PerformAutoML](#)

AutoML を実行するかどうか。Amazon Forecast が AutoML を実行すると、提供するアルゴリズムが評価され、トレーニングデータセットに最適なアルゴリズムと設定が選択されます。

デフォルト値は、`false`です。この場合、アルゴリズムを指定する必要があります。

Amazon Forecast が AutoML を実行するように `PerformAutoML` を `true` に設定します。これは、トレーニングデータに適したアルゴリズムがわからない場合に適したオプションです。この場合、`PerformHPO` は `false` である必要があります。

型: ブール値

必須: いいえ

### [PerformHPO](#)

ハイパーパラメータの最適化 (HPO) を実行するかどうか。HPO は、トレーニングデータに最適なハイパーパラメータ値を見つけます。HPO を実行するプロセスは、ハイパーパラメータのチューニングジョブの実行として知られています。

デフォルト値は、`false`です。この場合、Amazon Forecast は、選択したアルゴリズムのデフォルトのハイパーパラメータ値を使用します。

デフォルト値を上書きするには、PerformHPO を true に設定し、オプションで [HyperParameterTuningJobConfig](#) オブジェクトを指定します。チューニングジョブは、最適化するメトリクス、チューニングに関係するハイパーパラメータ、およびチューニング可能な各ハイパーパラメータの有効範囲を指定します。この場合、アルゴリズムを指定する必要があり、PerformAutoML は false である必要があります。

次のアルゴリズムは HPO をサポートします。

- DeepAR+
- CNN-QR

型: ブール値

必須: いいえ

### PredictorName

予測子の名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: はい

### Tags

サービスに適用し、予測子の分類と整理に役立つオプションのメタデータ。タグはそれぞれ、1 つのキーとオプションの 1 つの値で設定されており、どちらもお客様側が定義します。

タグには以下のベーシックな制限があります。

- リソースあたりのタグの最大数は 50 です。
- タグキーは、リソースごとにそれぞれ一意である必要があります。また、各タグキーに設定できる値は 1 つのみです。
- キーの最大長 - 128 文字 (Unicode) (UTF-8)。
- 値の最大長 - 256 文字 (Unicode) (UTF-8)。
- 複数の のサービス間およびリソース間でタグ付けスキーマを使用する場合、他のサービスにも許容される文字数に制限がある可能性があることに注意してください。一般的に使用が許可される文字は、UTF-8 で表現可能な文字、数字、スペース、および以下の文字です: +, -, =, ., \_, :, /, @。

- タグのキーと値では、大文字と小文字が区別されます。
- `aws:`、`AWS:`、またはキーのプレフィックスなどの大文字または小文字の組み合わせは使用しないでください。AWS 使用のために予約されています。このプレフィックスが含まれるタグキーを編集したり削除することはできません。値にはこのプレフィックスを付けることができます。タグ値には `aws` がプレフィックスとして付されているが、キーには付されていない場合、Forecast は、それをユーザータグとみなし、タグ数の上限である 50 個を計算する際に算入します。`aws` のキープレフィックスのみを持つタグは、リソース制限あたりのタグに算入されません。

タイプ: [Tag](#) オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 200 項目です。

必須: いいえ

## [TrainingParameters](#)

モデルトレーニングのために上書きするハイパーパラメータ。上書きできるハイパーパラメータは、個々のアルゴリズムにリストされています。サポートされているアルゴリズムのリストについては、「[Amazon Forecast アルゴリズム](#)」を参照してください。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 100 項目です。

キーの長さの制限: 最大長は 256 です。

キーパターン: `^[a-zA-Z0-9\-\_\.\ \\/\[\]\,\\"\\\s]+$`

値の長さの制限: 最大長は 256 です。

値のパターン: `^[a-zA-Z0-9\-\_\.\ \\/\[\]\,\\"\\\s]+$`

必須: いいえ

## レスポンスの構文

```
{
  "PredictorArn": "string"
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### PredictorArn

予測子の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

### LimitExceededException

アカウントあたりのリソース数の制限を超えました。

HTTP ステータスコード: 400

### ResourceAlreadyExistsException

この名前のリソースは既に存在します。別の名前で再試行してください。

HTTP ステータスコード: 400

### ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード: 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreatePredictorBacktestExportJob

サービス: Amazon Forecast Service

[CreateAutoPredictor](#) または [CreatePredictor](#) オペレーションによって生成されたバックテスト予測および精度メトリクスをエクスポートします。CSV または Parquet ファイルを含む 2 つのフォルダが、指定した S3 バケットにエクスポートされます。

エクスポートファイル名は、次の規則に一致します。

<ExportJobName>\_<ExportTimestamp>\_<PartNumber>.csv

<ExportTimestamp> コンポーネントは Java SimpleDate 形式 (yyyy-MM-ddTHH -mm-ssZ) です。

Amazon S3 バケットと、Amazon Forecast が Amazon S3 バケットにアクセスするために引き受けられることができる AWS Identity and Access Management (IAM) Amazon S3 ロールを含む [DataDestination](#) オブジェクトを指定する必要があります。詳細については、「[Amazon Forecast のアクセス許可を設定する](#)」を参照してください。

### Note

Amazon S3 バケットのエクスポートにアクセスする前に、エクスポートジョブの Status が ACTIVE となっている必要があります。[DescribePredictorBacktestExportJob](#) オペレーションを使用して、ステータスを取得します。

### リクエストの構文

```
{
  "Destination": {
    "S3Config": {
      "KMSKeyArn": "string",
      "Path": "string",
      "RoleArn": "string"
    }
  },
  "Format": "string",
  "PredictorArn": "string",
  "PredictorBacktestExportJobName": "string",
  "Tags": [
    {
      "Key": "string",
```

```
    "Value": "string"  
  }  
]  
}
```

## リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### Destination

エクスポートジョブの宛先。S3 パス、Amazon Forecast がロケーションにアクセスできるようにする AWS Identity and Access Management (IAM) ロール、および AWS Key Management Service (KMS) キー (オプション) を指定します。

型: [DataDestination](#) オブジェクト

必須: はい

### Format

エクスポートされたデータの形式は CSV または PARQUET のいずれかです。デフォルト値は「CSV」です。

型: 文字列

長さの制限: 最大長は 7 です。

Pattern: ^CSV|PARQUET\$

必須: いいえ

### PredictorArn

エクスポートする予測子の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: arn:([a-z\d-]+):forecast:.\*:.\*:.\*

必須: はい

## PredictorBacktestExportJobName

バックテストエクスポートジョブの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: はい

## Tags

バックテストの分類と整理に役立つオプションのメタデータ。タグはそれぞれ、1つのキーとオプションの1つの値で設定されており、どちらもお客様側が定義します。タグのキーと値では、大文字と小文字が区別されます。

タグには以下の制限があります。

- タグキーは、リソースごとにそれぞれ一意である必要があります。また、各タグキーに設定できる値は1つのみです。
- リソースあたりのタグの最大数: 50。
- キーの最大長: 128 文字 (Unicode) (UTF-8)。
- 値の最大長: 256 文字 (Unicode) (UTF-8)。
- 使用可能な文字: すべての文字と数字、UTF-8 で表現可能なスペース、および + - = 。 \_ : / @ 。タグ付けスキーマが他のサービスやリソースで使用されている場合、それらのサービスの文字制限も適用されます。
- キープレフィックスには、aws: または AWS: の大文字または小文字の組み合わせを含めることはできません。値にはこのプレフィックスを付けることができます。タグ値には aws がプレフィックスとして付されているが、キーには付されていない場合、Forecast は、それをユーザータグとみなし、タグ数の上限である 50 個を計算する際に算入します。aws のキープレフィックスのみを持つタグは、リソース制限あたりのタグに算入されません。このプレフィックスが含まれるタグキーを編集したり削除することはできません。

タイプ: [Tag](#) オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 200 項目です。

必須: いいえ



## レスポンスの構文

```
{
  "PredictorBacktestExportJobArn": "string"
}
```

### レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

#### [PredictorBacktestExportJobArn](#)

エクスポートする予測子バックテストエクスポートジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

### エラー

#### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

#### LimitExceededException

アカウントあたりのリソース数の制限を超えました。

HTTP ステータスコード: 400

#### ResourceAlreadyExistsException

この名前のリソースは既に存在します。別の名前で再試行してください。

HTTP ステータスコード: 400

#### ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreateWhatIfAnalysis

サービス: Amazon Forecast Service

What-If 分析は、時系列に仮定の変更を加え、その変化によって生成された予想を、ベースラインの未変更時系列と比較するシナリオモデリング手法です。What-If 分析の目的は、ベースラインの時系列にさまざまな変更が加えられた場合に、予想がどのように変化するかを理解することであることを覚えておいてください。

例えば、ある衣料品小売業者が、新しいスタイル用にスペースを空けるためにシーズン終了時のセールを検討していると想像してみてください。ベースライン予想を作成したら、What-If 分析を使用して、さまざまな販売戦略が目標にどのように影響するかを調査できます。

すべてに 25% の値下げを付与するシナリオと、すべてに固定ドルの値下げを付与するシナリオを作成できます。セールが 1 週間続くシナリオと、セールが 1 か月続くシナリオを作成できます。What-if 分析では、さまざまなシナリオを互いに比較できます。

What-If 分析は、予想モデルが学習した内容と、評価中のシナリオで予想モデルがどのように動作するかを表示するためのものであることに注意してください。What-If 分析の結果をやみくもにビジネス上の意思決定に使用しないでください。例えば、予想が適切かどうかを判断するための参考資料がないような新しいシナリオでは、予想が正確でない可能性があります。

[TimeSeriesSelector](#) オブジェクトは what-if 分析に必要な項目を定義します。

### Note

What-If 分析を作成するには、データをカンマ区切り値 (CSV) 形式にする必要があります。

### リクエストの構文

```
{
  "ForecastArn": "string",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ],
  "TimeSeriesSelector": {
    "TimeSeriesIdentifiers": {
      "DataSource": {
```

```
    "S3Config": {
      "KMSKeyArn": "string",
      "Path": "string",
      "RoleArn": "string"
    },
    "Format": "string",
    "Schema": {
      "Attributes": [
        {
          "AttributeName": "string",
          "AttributeType": "string"
        }
      ]
    }
  },
  "WhatIfAnalysisName": "string"
}
```

## リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### ForecastArn

ベースライン予想の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

### Tags

What-If 予想に適用する [タグ](#) のリスト。

タイプ: [Tag](#) オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 200 項目です。

必須: いいえ

## [TimeSeriesSelector](#)

TimeSeriesIdentifiers オブジェクトの What-If 分析で使用される時系列のセットを定義します。What-If 分析は、このオブジェクトの時系列に対してのみ実行されます。

TimeSeriesIdentifiers オブジェクトには次の情報が必要です。

- DataSource
- Format
- Schema

タイプ: [TimeSeriesSelector](#) オブジェクト

必須: いいえ

## [WhatIfAnalysisName](#)

What-If 分析の名前。それぞれの名前は一意である必要があります。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: はい

## レスポンスの構文

```
{
  "WhatIfAnalysisArn": "string"
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

## [WhatIfAnalysisArn](#)

What-If 分析の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

### LimitExceededException

アカウントあたりのリソース数の制限を超えました。

HTTP ステータスコード: 400

### ResourceAlreadyExistsException

この名前のリソースは既に存在します。別の名前で再試行してください。

HTTP ステータスコード: 400

### ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード: 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード: 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)

- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreateWhatIfForecast

サービス: Amazon Forecast Service

What-If 予想は、ベースライン予測を修正したバージョンから作成された予想です。各 What-If 予想には、代替データセットまたは元のデータセットに対する一連の変換が組み込まれています。

### リクエストの構文

```
{
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ],
  "TimeSeriesReplacementsDataSource": {
    "Format": "string",
    "S3Config": {
      "KMSKeyArn": "string",
      "Path": "string",
      "RoleArn": "string"
    },
    "Schema": {
      "Attributes": [
        {
          "AttributeName": "string",
          "AttributeType": "string"
        }
      ]
    },
    "TimestampFormat": "string"
  },
  "TimeSeriesTransformations": [
    {
      "Action": {
        "AttributeName": "string",
        "Operation": "string",
        "Value": number
      },
      "TimeSeriesConditions": [
        {
          "AttributeName": "string",
          "AttributeValue": "string",

```



```
        "Condition": "string"
      }
    ]
  },
  "WhatIfAnalysisArn": "string",
  "WhatIfForecastName": "string"
}
```

## リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### Tags

What-If 予想に適用する [タグ](#) のリスト。

タイプ: [Tag](#) オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 200 項目です。

必須: いいえ

### TimeSeriesReplacementsDataSource

関連する時系列データセットで変更したい行を含む代替時系列データセット。代替時系列には、ベースライン関連の時系列に含まれるすべての行が含まれている必要はありません。What-If 予想に含めたい行 (メジャーとディメンションの組み合わせ) のみを含めてください。

このデータセットを元の時系列と統合して、What-If 分析に使用される変換済みデータセットを作成します。

このデータセットには、変更する項目 (item\_id や workforce\_type など)、関連するディメンション、タイムスタンプ列、および関連する時系列列の少なくとも 1 つが含まれている必要があります。このファイルには、同じ時系列のタイムスタンプが重複してはいけません。このファイルは CSV 形式である必要があります。

このデータセットに含まれていないタイムスタンプと item\_id は What-If 分析には含まれません。

タイプ: [TimeSeriesReplacementsDataSource](#) オブジェクト

必須: いいえ

## TimeSeriesTransformations

ベースライン時系列に適用される変換。各変換には 1 つのアクションと一連の条件が含まれます。アクションはすべての条件が満たされたときにのみ適用されます。条件を指定しない場合、アクションはすべての項目に適用されます。

タイプ : [TimeSeriesTransformation](#) オブジェクトの配列

配列メンバー : 最小数は 0 項目です。最大数は 30 項目です。

必須: いいえ

## WhatIfAnalysisArn

What-If 分析の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限 : 最大長は 256 です。

パターン : `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須 : はい

## WhatIfForecastName

What-If 予想の名前。名前は、各 What-If 分析内で一意である必要があります。

型: 文字列

長さの制限 : 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須 : はい

## レスポンスの構文

```
{
  "WhatIfForecastArn": "string"
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### WhatIfForecastArn

What-If 予想の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 300 です。

パターン : `arn:([a-z\d-]+):forecast:.*:.*:.*`

### エラー

#### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード : 400

#### LimitExceededException

アカウントあたりのリソース数の制限を超えました。

HTTP ステータスコード : 400

#### ResourceAlreadyExistsException

この名前のリソースは既に存在します。別の名前で再試行してください。

HTTP ステータスコード : 400

#### ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

#### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## CreateWhatIfForecastExport

サービス: Amazon Forecast Service

[CreateWhatIfForecast](#) オペレーションによって作成された予測を Amazon Simple Storage Service (Amazon S3) バケットにエクスポートします。予測ファイル名は、次の規則に一致します。

≈<ForecastExportJobName>\_<ExportTimestamp>\_<PartNumber>

<ExportTimestamp> コンポーネントは Java SimpleDateFormat (yyyy-MM-ddTHH -mm-ssZ) にあります。

Amazon Forecast が Amazon S3 バケットにアクセスするために引き受けることができる AWS Identity and Access Management (IAM) ロールを含む [DataDestination](#) オブジェクトを指定する必要があります。詳細については、「[Amazon Forecast のアクセス許可を設定する](#)」を参照してください。

詳細については、「[予測の生成](#)」を参照してください。

すべての What-If 予想エクスポートジョブのリストを取得するには、[ListWhatIfForecastExports](#) オペレーションを使用します。

### Note

Amazon S3 バケットの予測にアクセスする前に、予測エクスポートジョブの Status が ACTIVE となっている必要があります。[DescribeWhatIfForecastExport](#) オペレーションを使用して、ステータスを取得します。

### リクエストの構文

```
{
  "Destination": {
    "S3Config": {
      "KMSKeyArn": "string",
      "Path": "string",
      "RoleArn": "string"
    }
  },
  "Format": "string",
  "Tags": [
    {
```

```
    "Key": "string",
    "Value": "string"
  }
],
"WhatIfForecastArns": [ "string" ],
"WhatIfForecastExportName": "string"
}
```

## リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### Destination

予測を保存する場所と、Amazon Forecast がその場所にアクセスするために引き受けることができる AWS Identity and Access Management (IAM) ロール。予測は Amazon S3 バケットにエクスポートする必要があります。

暗号化を使用する場合は、に AWS Key Management Service (KMS) キーを含める Destination 必要があります。IAM ロールは、Amazon Forecast がキーにアクセスすることを許可する必要があります。

型: [DataDestination](#) オブジェクト

必須: はい

### Format

エクスポートされたデータの形式は CSV または PARQUET のいずれかです。

型: 文字列

長さの制限: 最大長は 7 です。

Pattern: ^CSV|PARQUET\$

必須: いいえ

### Tags

What-If 予想に適用する [タグ](#) のリスト。

タイプ: [Tag](#) オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 200 項目です。

必須: いいえ

### [WhatIfForecastArns](#)

エクスポートする What-If 予想 Amazon リソースネーム (ARN) のリスト。

タイプ: 文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 50 項目です。

長さの制限: 最大長は 300 です。

Pattern: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

必須: はい

### [WhatIfForecastExportName](#)

エクスポートする What-If 予想の名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: はい

### レスポンスの構文

```
{
  "WhatIfForecastExportArn": "string"
}
```

### レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### [WhatIfForecastExportArn](#)

What-If 予想の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 300 です。

パターン : `arn:([a-z\d-]+):forecast:.*:.*:.*`

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード : 400

### LimitExceededException

アカウントあたりのリソース数の制限を超えました。

HTTP ステータスコード : 400

### ResourceAlreadyExistsException

この名前のリソースは既に存在します。別の名前で再試行してください。

HTTP ステータスコード : 400

### ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)



- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteDataset

サービス: Amazon Forecast Service

[CreateDataset](#) オペレーションを使用して作成された Amazon Forecast データセットを削除します。ステータスが ACTIVE または CREATE\_FAILED のデータセットのみを削除できます。ステータスを取得するには、[DescribeDataset](#) オペレーションを使用します。

### Note

Forecast は、削除されたデータセットを含むデータセットグループを自動的に更新しません。データセットグループを更新するには、[UpdateDatasetGroup](#) オペレーションを使用し、削除されたデータセットの ARN を省略します。

### リクエストの構文

```
{
  "DatasetArn": "string"
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### DatasetArn

削除するデータセットの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

### レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード : 400

### ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteDatasetGroup

サービス: Amazon Forecast Service

[CreateDatasetGroup](#) オペレーションを使用して作成されたデータセットグループを削除します。削除できるのは、ステータスが ACTIVE、CREATE\_FAILED、または UPDATE\_FAILED のデータセットグループのみです。ステータスを取得するには、[DescribeDatasetGroup](#) オペレーションを使用します。

このオペレーションでは、データセットグループのみが削除され、グループ内のデータセットは削除されません。

### リクエストの構文

```
{
  "DatasetGroupArn": "string"
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### [DatasetGroupArn](#)

削除するデータセットグループの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

### レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

### エラー

#### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード : 400

#### ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

#### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteDatasetImportJob

サービス: Amazon Forecast Service

[CreateDatasetImportJob](#) オペレーションを使用して作成されたデータセットのインポートジョブを削除します。ステータスが ACTIVE または CREATE\_FAILED のデータセットのインポートジョブのみを削除できます。ステータスを取得するには、[DescribeDatasetImportJob](#) オペレーションを使用します。

リクエストの構文

```
{  
  "DatasetImportJobArn": "string"  
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### [DatasetImportJobArn](#)

削除するデータセットのインポートジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

## ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

## ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteExplainability

サービス: Amazon Forecast Service

Explainability リソースを削除します。

ステータスが ACTIVE または CREATE\_FAILED の予測子のみを削除できません。[DescribeExplainability](#) オペレーションを使用して、ステータスを取得します。

リクエストの構文

```
{
  "ExplainabilityArn": "string"
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### [ExplainabilityArn](#)

削除する Explainability リソースの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400



## ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

## ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteExplainabilityExport

サービス: Amazon Forecast Service

Explainability エクスポートを削除します。

リクエストの構文

```
{  
  "ExplainabilityExportArn": "string"  
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### ExplainabilityExportArn

削除する Explainability エクスポートの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteForecast

サービス: Amazon Forecast Service

[CreateForecast](#) オペレーションを使用して作成された予測を削除します。ステータスが ACTIVE または CREATE\_FAILED の予測のみを削除できます。[DescribeForecast](#) オペレーションを使用して、ステータスを取得します。

エクスポート中に予測を削除することはできません。予測を削除すると、予測のクエリを実行できなくなります。

### リクエストの構文

```
{
  "ForecastArn": "string"
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### [ForecastArn](#)

削除する予測の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

### レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

### エラー

#### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード : 400

#### ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

#### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteForecastExportJob

サービス: Amazon Forecast Service

[CreateForecastExportJob](#) オペレーションを使用して作成された予測エクスポートジョブを削除します。ステータスが ACTIVE または CREATE\_FAILED のエクスポートジョブのみを削除できます。[DescribeForecastExportJob](#) オペレーションを使用して、ステータスを取得します。

リクエストの構文

```
{
  "ForecastExportJobArn": "string"
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### [ForecastExportJobArn](#)

削除する予測エクスポートジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

## ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteMonitor

サービス: Amazon Forecast Service

モニターリソースを削除します。削除できるの

は、ACTIVE、ACTIVE\_STOPPED、CREATE\_FAILED、または CREATE\_STOPPED ステータスのモニターリソースだけです。

リクエストの構文

```
{  
  "MonitorArn": "string"  
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### MonitorArn

削除するモニターリソースの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

**InvalidInputException**

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

**ResourceInUseException**

指定されたリソースは使用中です。



HTTP ステータスコード : 400

## ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeletePredictor

サービス: Amazon Forecast Service

[DescribePredictor](#) または [CreatePredictor](#) オペレーションを使用して作成された予測子を削除します。ステータスが ACTIVE または CREATE\_FAILED の予測子のみを削除できます。[DescribePredictor](#) オペレーションを使用して、ステータスを取得します。

リクエストの構文

```
{
  "PredictorArn": "string"
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### [PredictorArn](#)

削除する予測子の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

`InvalidInputException`

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

`ResourceInUseException`

指定されたリソースは使用中です。

HTTP ステータスコード : 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeletePredictorBacktestExportJob

サービス: Amazon Forecast Service

予測子バックテストエクスポートジョブを削除します。

リクエストの構文

```
{  
  "PredictorBacktestExportJobArn": "string"  
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### PredictorBacktestExportJobArn

削除する予測子バックテストエクスポートジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteResourceTree

サービス: Amazon Forecast Service

リソースツリー全体を削除します。このオペレーションでは、親リソースとその子リソースを削除します。

子リソースは、別のリソースから作成されたリソースです。例えば、予測が予測子から生成される場合、予測は子リソースであり、予測子は親リソースです。

Amazon Forecast のリソースには、次の親-子リソース階層があります。

- [Dataset] (データセット): データセットのインポートジョブ
- [Dataset Group] (データセットグループ): 予測子、予測子バックテストエクスポートジョブ、予測、予測エクスポートジョブ
- [Predictor] (予測子): 予測子バックテストエクスポートジョブ、予測、予測エクスポートジョブ
- [Forecast] (予測): 予測エクスポートジョブ

### Note

DeleteResourceTree は、Amazon Forecast のリソースのみを削除し、Amazon S3 に保存されているデータセットやエクスポートされたファイルは削除しません。

### リクエストの構文

```
{  
  "ResourceArn": "string"  
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### ResourceArn

削除する親リソースの Amazon リソースネーム (ARN)。親リソースのすべての子リソースも削除されます。

型: 文字列

長さの制限：最大長は 256 です。

パターン：arn:([a-z\d-]+):forecast:.\*:.\*:.\*+

必須：はい

## レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード：400

### ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード：400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード：400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)

- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)



## DeleteWhatIfAnalysis

サービス: Amazon Forecast Service

[CreateWhatIfAnalysis](#) オペレーションを使用して作成された What-If 分析を削除します。削除できるのは、ステータスが ACTIVE または CREATE\_FAILED の What-If 分析だけです。[DescribeWhatIfAnalysis](#) オペレーションを使用して、ステータスを取得します。

予想のいずれかがエクスポートされている間は、What-If 分析を削除することはできません。

リクエストの構文

```
{  
  "WhatIfAnalysisArn": "string"  
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### [WhatIfAnalysisArn](#)

削除する What-If 分析の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

## ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

## ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteWhatIfForecast

サービス: Amazon Forecast Service

[CreateWhatIfForecast](#) オペレーションを使用して作成された What-If 予想を削除します。ステータスが ACTIVE または CREATE\_FAILED の What-If 予想のみを削除できます。[DescribeWhatIfForecast](#) オペレーションを使用して、ステータスを取得します。

エクスポート中に What-If 予想を削除することはできません。What-If 予想を削除すると、What-If 分析をクエリできなくなります。

### リクエストの構文

```
{
  "WhatIfForecastArn": "string"
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### [WhatIfForecastArn](#)

削除する What-If 予想の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 300 です。

Pattern: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

### レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

### エラー

#### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード : 400

#### ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

#### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DeleteWhatIfForecastExport

サービス: Amazon Forecast Service

[CreateWhatIfForecastExport](#) オペレーションを使用して作成された What-If 予想エクスポートを削除します。ステータスが ACTIVE または CREATE\_FAILED の What-If 予想エクスポートのみを削除できます。[DescribeWhatIfForecastExport](#) オペレーションを使用して、ステータスを取得します。

リクエストの構文

```
{
  "WhatIfForecastExportArn": "string"
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### [WhatIfForecastExportArn](#)

削除する What-If 予想エクスポートの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 300 です。

Pattern: arn:([a-z\d-]+):forecast:.\*:.\*:.\*

必須: はい

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeAutoPredictor

サービス: Amazon Forecast Service

CreateAutoPredictor オペレーションを使用して作成された予測子を記述します。

### リクエストの構文

```
{
  "PredictorArn": "string"
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### PredictorArn

予測子の Amazon リソース名前 (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

### レスポンスの構文

```
{
  "CreationTime": number,
  "DataConfig": {
    "AdditionalDatasets": [
      {
        "Configuration": {
          "string": [ "string" ]
        },
        "Name": "string"
      }
    ],
    "AttributeConfigs": [
      {
        "AttributeName": "string",
```

```
        "Transformations": {
            "string": "string"
        }
    ],
    "DatasetGroupArn": "string"
},
"DatasetImportJobArns": [ "string" ],
"EncryptionConfig": {
    "KMSKeyArn": "string",
    "RoleArn": "string"
},
"EstimatedTimeRemainingInMinutes": number,
"ExplainabilityInfo": {
    "ExplainabilityArn": "string",
    "Status": "string"
},
"ForecastDimensions": [ "string" ],
"ForecastFrequency": "string",
"ForecastHorizon": number,
"ForecastTypes": [ "string" ],
"LastModificationTime": number,
"Message": "string",
"MonitorInfo": {
    "MonitorArn": "string",
    "Status": "string"
},
"OptimizationMetric": "string",
"PredictorArn": "string",
"PredictorName": "string",
"ReferencePredictorSummary": {
    "Arn": "string",
    "State": "string"
},
"Status": "string",
"TimeAlignmentBoundary": {
    "DayOfMonth": number,
    "DayOfWeek": "string",
    "Hour": number,
    "Month": "string"
}
}
```



## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### CreationTime

CreateAutoPredictor リクエストのタイムスタンプ。

型: タイムスタンプ

### DataConfig

データセットグループと追加のデータセットのデータ設定。

タイプ: [DataConfig](#) オブジェクト

### DatasetImportJobArns

予測子のトレーニングデータをインポートするために使用されるデータセットのインポートジョブの ARN の配列。

タイプ: 文字列の配列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

### EncryptionConfig

Amazon Forecast がキーにアクセスするために引き受けることができる AWS Key Management Service (KMS) キーと AWS Identity and Access Management (IAM) ロール。このオプションのオブジェクトは、[CreateDataset](#) および [CreatePredictor](#) リクエストで指定できます。

タイプ: [EncryptionConfig](#) オブジェクト

### EstimatedTimeRemainingInMinutes

予測子トレーニングジョブが完了するまでの推定残り時間 (分)。

型: 長整数

### ExplainabilityInfo

Predictor Explainability のステータスと ARN を提供します。

タイプ: [ExplainabilityInfo](#) オブジェクト

### [ForecastDimensions](#)

時系列をグループ化するために使用される属性を指定するディメンション (フィールド) 名の配列。

タイプ: 文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 10 項目です。

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

### [ForecastFrequency](#)

予測内の予測の頻度。

有効な間隔は、Y (年)、M (月)、W (週)、D (日)、H (時間)、30min (30分)、15min (15分)、10min (10分)、5min (5分)、および 1min (1分) です。例えば、[Y] は毎年を示し、[5min] は 5 分ごとを示します。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 5 です。

Pattern: `^Y|M|W|D|H|30min|15min|10min|5min|1min$`

### [ForecastHorizon](#)

モデルが予測するタイムステップの数。予測期間は予測長とも呼ばれます。

タイプ: 整数

### [ForecastTypes](#)

予測子のトレーニング中に使用される予測タイプ。デフォルト値は ["0.1","0.5","0.9"] です。

タイプ: 文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 20 項目です。

長さの制限: 最小長は 2 です。最大長は 4 です。

Pattern: `(^0?\.\\d\\d?$|^mean$)`

## LastModificationTime

リソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。

- CREATE\_PENDING (CreationTime)
- CREATE\_IN\_PROGRESS - 現在のタイムスタンプ。
- CREATE\_STOPPING - 現在のタイムスタンプ。
- CREATE\_STOPPED - ジョブが停止した時点。
- ACTIVE または CREATE\_FAILED - ジョブが終了または失敗した時点。

型: タイムスタンプ

## Message

エラーが発生した場合は、エラーの原因を詳しく説明するメッセージ。

型: 文字列

## MonitorInfo

Amazon リソースネーム (ARN) とステータスを含む [MonitorInfo](#) オブジェクト。

タイプ: [MonitorInfo](#) オブジェクト

## OptimizationMetric

予測子を最適化するために使用される精度メトリクス。

型: 文字列

有効な値: WAPE | RMSE | AverageWeightedQuantileLoss | MASE | MAPE

## PredictorArn

予測子の Amazon リソースネーム (ARN)

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

## PredictorName

予測子の名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

### [ReferencePredictorSummary](#)

参照予測子の ARN と状態。このパラメータは、再トレーニング済みまたはアップグレード済みの予測子についてのみ有効です。

タイプ: [ReferencePredictorSummary](#) オブジェクト

### [Status](#)

予測子のステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED

型: 文字列

長さの制限: 最大長は 256 です。

### [TimeAlignmentBoundary](#)

Forecast がデータを集計するときに使用する時間境界。

型: [TimeAlignmentBoundary](#) オブジェクト

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeDataset

サービス: Amazon Forecast Service

[CreateDataset](#) オペレーションを使用して作成された Amazon Forecast データセットについて説明します。

このオペレーションは、CreateDataset リクエストで指定されたパラメータの一覧表示に加えて、次のデータセットのプロパティを含めます。

- CreationTime
- LastModificationTime
- Status

### リクエストの構文

```
{  
  "DatasetArn": "string"  
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### [DatasetArn](#)

データセットの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

### レスポンスの構文

```
{  
  "CreationTime": number,  
  "DataFrequency": "string",  
}
```

```
"DatasetArn": "string",
"DatasetName": "string",
"DatasetType": "string",
"Domain": "string",
"EncryptionConfig": {
  "KMSKeyArn": "string",
  "RoleArn": "string"
},
"LastModificationTime": number,
"Schema": {
  "Attributes": [
    {
      "AttributeName": "string",
      "AttributeType": "string"
    }
  ]
},
"Status": "string"
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### CreationTime

データセットが作成された時点。

型: タイムスタンプ

### DataFrequency

データ収集の頻度。

有効な間隔は、Y (年)、M (月)、W (週)、D (日)、H (時間)、30min (30分)、15min (15分)、10min (10分)、5min (5分)、および 1min (1分) です。例えば、[M] は毎月を示し、[30min] は 30 分ごとを示します。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 5 です。

Pattern: ^Y|M|W|D|H|30min|15min|10min|5min|1min\$

## DatasetArn

データセットの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限 : 最大長は 256 です。

パターン : `arn:([a-z\d-]+):forecast:.*:.*:.*+`

## DatasetName

データセットの名前。

型: 文字列

長さの制限 : 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

## DatasetType

データセットのタイプ。

型: 文字列

有効な値 : TARGET\_TIME\_SERIES | RELATED\_TIME\_SERIES | ITEM\_METADATA

## Domain

データセットに関連付けられているドメイン。

型: 文字列

有効な値 : RETAIL | CUSTOM | INVENTORY\_PLANNING | EC2\_CAPACITY | WORK\_FORCE  
| WEB\_TRAFFIC | METRICS

## EncryptionConfig

Amazon Forecast がキーにアクセスするために引き受けることができる AWS Key Management Service (KMS) キーと AWS Identity and Access Management (IAM) ロール。

タイプ : [EncryptionConfig](#) オブジェクト

## LastModificationTime

データセットを作成する場合、LastModificationTime は CreationTime と同じです。データがデータセットにインポートされている間、LastModificationTime は



DescribeDataset コールの現在の時刻です。[CreateDatasetImportJob](#) オペレーションが完了すると、LastModificationTimeはインポートジョブが完了または失敗した時点になります。

型: タイムスタンプ

### [Schema](#)

データセットフィールドを指定する SchemaAttribute オブジェクトの配列。各 SchemaAttribute がフィールドの名前とデータ型を指定します。

タイプ: [Schema](#) オブジェクト

### [Status](#)

データセットのステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED
- UPDATE\_PENDING, UPDATE\_IN\_PROGRESS, UPDATE\_FAILED

UPDATE 状態は、[CreateDatasetImportJob](#) オペレーションの呼び出しからデータがデータセットにインポートされる間に適用され、データセットのインポートジョブのステータスを反映します。例えば、インポートジョブのステータスが CREATE\_IN\_PROGRESS の場合、データセットのステータスは UPDATE\_IN\_PROGRESS です。

#### Note

データセットの Status は、トレーニングデータをインポートする前に ACTIVE である必要があります。

型: 文字列

長さの制限: 最大長は 256 です。

### エラー

#### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード : 400

ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeDatasetGroup

サービス: Amazon Forecast Service

[CreateDatasetGroup](#) オペレーションを使用して作成されたデータセットグループを記述します。

このオペレーションは、CreateDatasetGroup リクエストで提供されたパラメータの一覧表示に加えて、次のプロパティを含めます。

- DatasetArns - グループに属するデータセット。
- CreationTime
- LastModificationTime
- Status

### リクエストの構文

```
{
  "DatasetGroupArn": "string"
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### [DatasetGroupArn](#)

データセットグループの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

### レスポンスの構文

```
{
```

```
"CreationTime": number,  
"DatasetArns": [ "string" ],  
"DatasetGroupArn": "string",  
"DatasetGroupName": "string",  
"Domain": "string",  
"LastModificationTime": number,  
"Status": "string"  
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### CreationTime

データセットグループが作成された時点。

型: タイムスタンプ

### DatasetArns

データセットグループに含まれるデータセットの Amazon リソースネーム (ARN) の配列。

タイプ: 文字列の配列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

### DatasetGroupArn

データセットグループの ARN。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

### DatasetGroupName

データセットグループの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

### Domain

データセットグループに関連付けられているドメイン。

型: 文字列

有効な値: RETAIL | CUSTOM | INVENTORY\_PLANNING | EC2\_CAPACITY | WORK\_FORCE  
| WEB\_TRAFFIC | METRICS

### LastModificationTime

データセットグループが作成または [UpdateDatasetGroup](#) オペレーションの呼び出しから最後に更新された日時。データセットグループが更新されている間、LastModificationTime は DescribeDatasetGroup コールの現在の時刻です。

型: タイムスタンプ

### Status

データセットグループのステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED
- UPDATE\_PENDING, UPDATE\_IN\_PROGRESS, UPDATE\_FAILED

UPDATE 状態は、[UpdateDatasetGroup](#) オペレーションを呼び出すときに適用されます。

#### Note

データセットグループを使用して予測子を作成する前に、データセットグループの Status が ACTIVE となっている必要があります。

型: 文字列

長さの制限: 最大長は 256 です。

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード : 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeDatasetImportJob

サービス: Amazon Forecast Service

[CreateDatasetImportJob](#) オペレーションを使用して作成されたデータセットのインポートジョブについて説明します。

このオペレーションは、CreateDatasetImportJob リクエストで提供されたパラメータの一覧表示に加えて、次のプロパティを含めます。

- CreationTime
- LastModificationTime
- DataSize
- FieldStatistics
- Status
- Message - エラーが発生した場合は、エラーに関する情報。

リクエストの構文

```
{  
  "DatasetImportJobArn": "string"  
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### [DatasetImportJobArn](#)

データセットのインポートジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

## レスポンスの構文

```
{
  "CreationTime": number,
  "DatasetArn": "string",
  "DatasetImportJobArn": "string",
  "DatasetImportJobName": "string",
  "DataSize": number,
  "DataSource": {
    "S3Config": {
      "KMSKeyArn": "string",
      "Path": "string",
      "RoleArn": "string"
    }
  },
  "EstimatedTimeRemainingInMinutes": number,
  "FieldStatistics": {
    "string" : {
      "Avg": number,
      "Count": number,
      "CountDistinct": number,
      "CountDistinctLong": number,
      "CountLong": number,
      "CountNan": number,
      "CountNanLong": number,
      "CountNull": number,
      "CountNullLong": number,
      "Max": "string",
      "Min": "string",
      "Stddev": number
    }
  },
  "Format": "string",
  "GeolocationFormat": "string",
  "ImportMode": "string",
  "LastModificationTime": number,
  "Message": "string",
  "Status": "string",
  "TimestampFormat": "string",
  "TimeZone": "string",
  "UseGeolocationForTimeZone": boolean
}
```



## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### CreationTime

データセットのインポートジョブが作成された時点。

型: タイムスタンプ

### DatasetArn

トレーニングデータのインポート先であるデータセットの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

### DatasetImportJobArn

データセットのインポートジョブの ARN。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

### DatasetImportJobName

データセットのインポートジョブの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

### DataSize

インポートジョブが終了した後のデータセットのサイズ (GB)。

型: 倍精度

## [DataSource](#)

インポートするトレーニングデータの場所と、Amazon Forecast がデータにアクセスするために引き受けることができる AWS Identity and Access Management (IAM) ロール。

暗号化が使用されている場合、には AWS Key Management Service (KMS) キー DataSource が含まれます。

タイプ: [DataSource](#) オブジェクト

## [EstimatedTimeRemainingInMinutes](#)

データセットのインポートジョブが完了するまでの推定残り時間 (分)。

型: 長整数

## [FieldStatistics](#)

入力データの各フィールドに関する統計情報。

タイプ: [Statistics](#) オブジェクトマップへの文字列です。

キーの長さの制限: 最大長は 256 です。

キーパターン: `^[a-zA-Z0-9\_]+`

## [Format](#)

インポートされたデータの形式は CSV または PARQUET のいずれかです。

型: 文字列

長さの制限: 最大長は 7 です。

Pattern: `^CSV|PARQUET$`

## [GeolocationFormat](#)

geolocation 属性の形式。有効な値: "LAT\_LONG" および "CC\_POSTALCODE"。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `^[a-zA-Z0-9\_]+`

## ImportMode

データセットのインポートジョブのインポートモード。完全または増分。

型: 文字列

有効な値 : FULL | INCREMENTAL

## LastModificationTime

リソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。

- CREATE\_PENDING (CreationTime)
- CREATE\_IN\_PROGRESS - 現在のタイムスタンプ。
- CREATE\_STOPPING - 現在のタイムスタンプ。
- CREATE\_STOPPED - ジョブが停止した時点。
- ACTIVE または CREATE\_FAILED - ジョブが終了または失敗した時点。

型: タイムスタンプ

## Message

エラーが発生した場合は、エラーに関する情報メッセージ。

型: 文字列

## Status

データセットのインポートジョブのステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED

型: 文字列

長さの制限 : 最大長は 256 です。

## TimestampFormat

データセット内のタイムスタンプの形式。指定する形式は、データセットが作成されたときに指定された DataFrequency によって異なります。以下の形式がサポートされています

- 「yyyy-MM-dd」

次のデータ周波数の場合: Y、M、W、D

- 「yyyy-MM-dd HH:mm:ss」

次のデータ頻度の場合: H、30min、15min、および 1min。オプションで、Y、M、W、および D

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `^[a-zA-Z0-9\-\:\.\,\'\s]+$`

### [TimeZone](#)

データセット内のすべての項目に適用される単一のタイムゾーン

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `^[a-zA-Z0-9\/\+\-\_]+$`

### [UseGeolocationForTimeZone](#)

TimeZone が geolocation 属性から自動的に派生するかどうか。

型: ブール値

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード: 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeExplainability

サービス: Amazon Forecast Service

[CreateExplainability](#) オペレーションを使用して作成された Explainability リソースを記述します。

リクエストの構文

```
{
  "ExplainabilityArn": "string"
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### [ExplainabilityArn](#)

記述する Explainability の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

レスポンスの構文

```
{
  "CreationTime": number,
  "DataSource": {
    "S3Config": {
      "KMSKeyArn": "string",
      "Path": "string",
      "RoleArn": "string"
    }
  },
  "EnableVisualization": boolean,
  "EndTime": "string",
  "EstimatedTimeRemainingInMinutes": number,
  "ExplainabilityArn": "string",
}
```

```
"ExplainabilityConfig": {
  "TimePointGranularity": "string",
  "TimeSeriesGranularity": "string"
},
"ExplainabilityName": "string",
"LastModificationTime": number,
"Message": "string",
"ResourceArn": "string",
"Schema": {
  "Attributes": [
    {
      "AttributeName": "string",
      "AttributeType": "string"
    }
  ]
},
"StartDateTime": "string",
"Status": "string"
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### [CreationTime](#)

Explainability リソースが作成された時点。

型: タイムスタンプ

### [DataSource](#)

データのソース、Amazon Forecast がデータにアクセスできるようにする AWS Identity and Access Management (IAM) ロール、およびオプションで AWS Key Management Service (KMS) キー。

タイプ: [DataSource](#) オブジェクト

### [EnableVisualization](#)

Explainability リソースのためにビジュアライゼーションが有効になっているかどうか。

型: ブール値

## EndTime

TimePointGranularity が SPECIFIC に設定されている場合、Explainability の最後の時間ポイント。

型: 文字列

長さの制限: 最大長は 19。

Pattern: `^\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}$`

## EstimatedTimeRemainingInMinutes

[CreateExplainability](#) ジョブが完了するまでの推定残り時間 (分)。

型: 長整数

## ExplainabilityArn

Explainability の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限 : 最大長は 256 です。

パターン : `arn:([a-z\d-]+):forecast:.*:.*:.*`

## ExplainabilityConfig

Explainability の時系列と時間ポイントの粒度を定義する構成設定。

タイプ : [ExplainabilityConfig](#) オブジェクト

## ExplainabilityName

Explainability の名前。

型: 文字列

長さの制限 : 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

## LastModificationTime

リソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。



- CREATE\_PENDING (CreationTime)
- CREATE\_IN\_PROGRESS - 現在のタイムスタンプ。
- CREATE\_STOPPING - 現在のタイムスタンプ。
- CREATE\_STOPPED - ジョブが停止した時点。
- ACTIVE または CREATE\_FAILED - ジョブが終了または失敗した時点。

型: タイムスタンプ

### Message

エラーが発生した場合は、エラーに関するメッセージ。

型: 文字列

### ResourceArn

Explainability リソースの作成に使用される予測子または予測の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

### Schema

データセットのフィールドを定義します。

タイプ: [Schema](#) オブジェクト

### StartDateTime

TimePointGranularity が SPECIFIC に設定されている場合、Explainability の最初の時間ポイント。

型: 文字列

長さの制限: 最大長は 19。

Pattern: `^\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}$`

### Status

Explainability リソースのステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED

型: 文字列

長さの制限: 最大長は 256 です。

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード: 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)



## DescribeExplainabilityExport

サービス: Amazon Forecast Service

[CreateExplainabilityExport](#) オペレーションを使用して作成された Explainability エクスポートを記述します。

リクエストの構文

```
{
  "ExplainabilityExportArn": "string"
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### [ExplainabilityExportArn](#)

Explainability エクスポートの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

レスポンスの構文

```
{
  "CreationTime": number,
  "Destination": {
    "S3Config": {
      "KMSKeyArn": "string",
      "Path": "string",
      "RoleArn": "string"
    }
  },
  "ExplainabilityArn": "string",
  "ExplainabilityExportArn": "string",
  "ExplainabilityExportName": "string",
}
```

```
"Format": "string",
"LastModificationTime": number,
"Message": "string",
"Status": "string"
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### CreationTime

Explainability エクスポートが作成された時点。

型: タイムスタンプ

### Destination

エクスポートジョブの宛先。S3 バス、Amazon Forecast がロケーションにアクセスできるようにする AWS Identity and Access Management (IAM) ロール、および AWS Key Management Service (KMS) キー (オプション) を指定します。

タイプ: [DataDestination](#) オブジェクト

### ExplainabilityArn

Explainability エクスポートの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

### ExplainabilityExportArn

Explainability エクスポートの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

## ExplainabilityExportName

Explainability エクスポートの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

## Format

エクスポートされたデータの形式は CSV または PARQUET のいずれかです。

型: 文字列

長さの制限: 最大長は 7 です。

Pattern: `^CSV|PARQUET$`

## LastModificationTime

リソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。

- CREATE\_PENDING (CreationTime)
- CREATE\_IN\_PROGRESS - 現在のタイムスタンプ。
- CREATE\_STOPPING - 現在のタイムスタンプ。
- CREATE\_STOPPED - ジョブが停止した時点。
- ACTIVE または CREATE\_FAILED - ジョブが終了または失敗した時点。

型: タイムスタンプ

## Message

エクスポート中に発生したエラーに関する情報。

型: 文字列

## Status

Explainability エクスポートのステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED

- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED

型: 文字列

長さの制限: 最大長は 256 です。

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード: 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeForecast

サービス: Amazon Forecast Service

[CreateForecast](#) オペレーションを使用して作成された予測を記述します。

このオペレーションは、CreateForecast リクエストで提供されたプロパティの一覧表示に加えて、次のプロパティを一覧表示します。

- DatasetGroupArn - トレーニングデータを提供したデータセットグループ。
- CreationTime
- LastModificationTime
- Status
- Message - エラーが発生した場合は、エラーに関する情報。

### リクエストの構文

```
{  
  "ForecastArn": "string"  
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### [ForecastArn](#)

予測の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

### レスポンスの構文

```
{
```



```
"CreationTime": number,
"DatasetGroupArn": "string",
"EstimatedTimeRemainingInMinutes": number,
"ForecastArn": "string",
"ForecastName": "string",
"ForecastTypes": [ "string" ],
"LastModificationTime": number,
"Message": "string",
"PredictorArn": "string",
"Status": "string",
"TimeSeriesSelector": {
  "TimeSeriesIdentifiers": {
    "DataSource": {
      "S3Config": {
        "KMSKeyArn": "string",
        "Path": "string",
        "RoleArn": "string"
      }
    },
    "Format": "string",
    "Schema": {
      "Attributes": [
        {
          "AttributeName": "string",
          "AttributeType": "string"
        }
      ]
    }
  }
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### CreationTime

予測作成タスクが作成された時点。

型: タイムスタンプ

## DatasetGroupArn

予測子のトレーニングに使用されるデータを提供したデータセットグループの ARN。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

## EstimatedTimeRemainingInMinutes

予測ジョブが完了するまでの推定残り時間 (分)。

型: 長整数

## ForecastArn

リクエストで指定された予測の ARN。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

## ForecastName

予測の名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

## ForecastTypes

確率的予測が生成された分位数。

タイプ: 文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 20 項目です。

長さの制限: 最小長は 2 です。最大長は 4 です。

Pattern: (^0?\. \d\d? \$|^mean\$)

### LastModificationTime

リソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。

- CREATE\_PENDING (CreationTime)
- CREATE\_IN\_PROGRESS - 現在のタイムスタンプ。
- CREATE\_STOPPING - 現在のタイムスタンプ。
- CREATE\_STOPPED - ジョブが停止した時点。
- ACTIVE または CREATE\_FAILED - ジョブが終了または失敗した時点。

型: タイムスタンプ

### Message

エラーが発生した場合は、エラーに関する情報メッセージ。

型: 文字列

### PredictorArn

予測の生成に使用される予測子の ARN。

型: 文字列

長さの制限 : 最大長は 256 です。

パターン : arn:([a-z\d-]+):forecast:.\*:.\*:.\*+

### Status

予測のステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED

#### Note

予測をクエリまたはエクスポートする前に、予測の Status が ACTIVE となっている必要があります。

型: 文字列

長さの制限 : 最大長は 256 です。

パターン : `^[a-zA-Z0-9\_]+$`

### [TimeSeriesSelector](#)

予想に含める時系列。

型: [TimeSeriesSelector](#) オブジェクト

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード : 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)

- [AWS SDK for Ruby V3](#)

## DescribeForecastExportJob

サービス: Amazon Forecast Service

[CreateForecastExportJob](#) オペレーションを使用して作成された予測エクスポートジョブを記述します。

このオペレーションは、CreateForecastExportJob リクエストでユーザーによって提供されたプロパティの一覧表示に加えて、次のプロパティを一覧表示します。

- CreationTime
- LastModificationTime
- Status
- Message - エラーが発生した場合は、エラーに関する情報。

### リクエストの構文

```
{  
  "ForecastExportJobArn": "string"  
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### [ForecastExportJobArn](#)

予測エクスポートジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

### レスポンスの構文

```
{
```

```
"CreationTime": number,
"Destination": {
  "S3Config": {
    "KMSKeyArn": "string",
    "Path": "string",
    "RoleArn": "string"
  }
},
"ForecastArn": "string",
"ForecastExportJobArn": "string",
"ForecastExportJobName": "string",
"Format": "string",
"LastModificationTime": number,
"Message": "string",
"Status": "string"
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### CreationTime

予測エクスポートジョブが作成された時点。

型: タイムスタンプ

### Destination

予測がエクスポートされる Amazon Simple Storage Service (Amazon S3) バケットへのパス。

タイプ: [DataDestination](#) オブジェクト

### ForecastArn

エクスポートされた予測の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]):forecast:.*:.*:.*`

## ForecastExportJobArn

予測エクスポートジョブの ARN。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]):forecast:.*:.*:.*`

## ForecastExportJobName

予測エクスポートジョブの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

## Format

エクスポートされたデータの形式は CSV または PARQUET のいずれかです。

型: 文字列

長さの制限: 最大長は 7 です。

Pattern: `^CSV|PARQUET$`

## LastModificationTime

リソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。

- CREATE\_PENDING (CreationTime)
- CREATE\_IN\_PROGRESS - 現在のタイムスタンプ。
- CREATE\_STOPPING - 現在のタイムスタンプ。
- CREATE\_STOPPED - ジョブが停止した時点。
- ACTIVE または CREATE\_FAILED - ジョブが終了または失敗した時点。

型: タイムスタンプ

## Message

エラーが発生した場合は、エラーに関する情報メッセージ。



型: 文字列

## Status

予測のエクスポートジョブのステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED

### Note

S3 バケットの予測にアクセスする前に、予測エクスポートジョブの Status が ACTIVE となっている必要があります。

型: 文字列

長さの制限: 最大長は 256 です。

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード: 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)

- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeMonitor

サービス: Amazon Forecast Service

モニターリソースを説明します。このオペレーションは、[CreateMonitor](#) リクエストで提供されたプロパティの一覧表示に加えて、次のプロパティを一覧表示します。

- Baseline
- CreationTime
- LastEvaluationTime
- LastEvaluationState
- LastModificationTime
- Message
- Status

### リクエストの構文

```
{  
  "MonitorArn": "string"  
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### [MonitorArn](#)

説明するモニターリソースの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

### レスポンスの構文

```
{
```

```
"Baseline": {
  "PredictorBaseline": {
    "BaselineMetrics": [
      {
        "Name": "string",
        "Value": number
      }
    ]
  }
},
"CreationTime": number,
"EstimatedEvaluationTimeRemainingInMinutes": number,
"LastEvaluationState": "string",
"LastEvaluationTime": number,
"LastModificationTime": number,
"Message": "string",
"MonitorArn": "string",
"MonitorName": "string",
"ResourceArn": "string",
"Status": "string"
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### [Baseline](#)

比較のベースラインとして使用するメトリクス。自動予測子のモニタリング結果を解釈するときは、これらの値を使用してください。

タイプ: [Baseline](#) オブジェクト

### [CreationTime](#)

モニターリソースが作成されたときのタイムスタンプ。

型: タイムスタンプ

### [EstimatedEvaluationTimeRemainingInMinutes](#)

モニターリソースが現在の評価を終了するまでの推定残り時間 (分)。

型: 長整数

### LastEvaluationState

モニターの最新の評価の状態。

型: 文字列

長さの制限: 最大長は 256 です。

### LastEvaluationTime

モニターが完了した最新の評価のタイムスタンプ。

型: タイムスタンプ

### LastModificationTime

モニターに加えられた最新の変更のタイムスタンプ。

型: タイムスタンプ

### Message

モニターに関するエラーメッセージ (存在する場合)。

型: 文字列

### MonitorArn

説明されたモニターリソースの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

### MonitorName

モニターの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

## ResourceArn

モニタリングされる自動予測子の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限 : 最大長は 256 です。

パターン : `arn:([a-z\d-]+):forecast:.*:.*:.*`

## Status

モニターリソースのステータス。

型: 文字列

長さの制限 : 最大長は 256 です。

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード : 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)

- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribePredictor

サービス: Amazon Forecast Service

### Note

このオペレーションは、で作成されたレガシー予測子に対してのみ有効です CreatePredictor。レガシー予測子を使用していない場合は、[DescribeAutoPredictor](#) を使用します。

[CreatePredictor](#) オペレーションを使用して作成された予測子を記述します。

このオペレーションは、CreatePredictor リクエストで提供されたプロパティの一覧表示に加えて、次のプロパティを一覧表示します。

- DatasetImportJobArns - トレーニングデータのインポートに使用されるデータセットのインポートジョブ。
- AutoMLAlgorithmArns - 評価されたアルゴリズム (AutoML が実行される場合)。
- CreationTime
- LastModificationTime
- Status
- Message - エラーが発生した場合は、エラーに関する情報。

リクエストの構文

```
{  
  "PredictorArn": "string"  
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### [PredictorArn](#)

情報を取得する予測子の Amazon リソースネーム (ARN)。

型: 文字列



長さの制限：最大長は 256 です。

パターン：arn:([a-z\d-]+):forecast:.\*:.\*:.\*+

必須：はい

## レスポンスの構文

```
{
  "AlgorithmArn": "string",
  "AutoMLAlgorithmArns": [ "string" ],
  "AutoMLOverrideStrategy": "string",
  "CreationTime": number,
  "DatasetImportJobArns": [ "string" ],
  "EncryptionConfig": {
    "KMSKeyArn": "string",
    "RoleArn": "string"
  },
  "EstimatedTimeRemainingInMinutes": number,
  "EvaluationParameters": {
    "BackTestWindowOffset": number,
    "NumberOfBacktestWindows": number
  },
  "FeaturizationConfig": {
    "Featurizations": [
      {
        "AttributeName": "string",
        "FeaturizationPipeline": [
          {
            "FeaturizationMethodName": "string",
            "FeaturizationMethodParameters": {
              "string": "string"
            }
          }
        ]
      }
    ]
  },
  "ForecastDimensions": [ "string" ],
  "ForecastFrequency": "string"
},
"ForecastHorizon": number,
"ForecastTypes": [ "string" ],
"HPOConfig": {
```

```
"ParameterRanges": {
  "CategoricalParameterRanges": [
    {
      "Name": "string",
      "Values": [ "string" ]
    }
  ],
  "ContinuousParameterRanges": [
    {
      "MaxValue": number,
      "MinValue": number,
      "Name": "string",
      "ScalingType": "string"
    }
  ],
  "IntegerParameterRanges": [
    {
      "MaxValue": number,
      "MinValue": number,
      "Name": "string",
      "ScalingType": "string"
    }
  ]
},
"InputDataConfig": {
  "DatasetGroupArn": "string",
  "SupplementaryFeatures": [
    {
      "Name": "string",
      "Value": "string"
    }
  ]
},
"IsAutoPredictor": boolean,
"LastModificationTime": number,
"Message": "string",
"OptimizationMetric": "string",
"PerformAutoML": boolean,
"PerformHPO": boolean,
"PredictorArn": "string",
"PredictorExecutionDetails": {
  "PredictorExecutions": [
    {
```

```

    "AlgorithmArn": "string",
    "TestWindows": [
      {
        "Message": "string",
        "Status": "string",
        "TestWindowEnd": number,
        "TestWindowStart": number
      }
    ]
  }
]
},
"PredictorName": "string",
"Status": "string",
"TrainingParameters": {
  "string" : "string"
}
}

```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### [AlgorithmArn](#)

モデルトレーニングに使用されるアルゴリズムの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

### [AutoMLAlgorithmArns](#)

PerformAutoML が指定されている場合、選択したアルゴリズムの ARN。

タイプ: 文字列の配列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

## [AutoMLOverrideStrategy](#)

### Note

LatencyOptimized AutoML 上書き戦略は、プライベートベータでのみ使用できます。アクセス権限の詳細については、AWS サポートまたはアカウントマネージャーにお問い合わせください。

予測子のトレーニングに使用する AutoML 戦略。LatencyOptimized が指定されていない限り、AutoML 戦略は予測子の精度を最適化します。

このパラメータは、AutoML を使用してトレーニングした予測子についてのみ有効です。

型: 文字列

有効な値 : LatencyOptimized | AccuracyOptimized

## [CreationTime](#)

モデルトレーニングタスクが作成された時点。

型: タイムスタンプ

## [DatasetImportJobArns](#)

予測子のトレーニングデータをインポートするために使用されるデータセットのインポートジョブの ARN の配列。

タイプ : 文字列の配列

長さの制限 : 最大長は 256 です。

パターン : arn:([a-z\d-]+):forecast:.\*:.\*:.\*+

## [EncryptionConfig](#)

Amazon Forecast がキーにアクセスするために引き受けることができる AWS Key Management Service (KMS) キーと AWS Identity and Access Management (IAM) ロール。

タイプ : [EncryptionConfig](#) オブジェクト

## [EstimatedTimeRemainingInMinutes](#)

予測子トレーニングジョブが完了するまでの推定残り時間 (分)。

型: 長整数

### [EvaluationParameters](#)

指定されたアルゴリズムのデフォルトの評価パラメータを上書きするために使用されます。Amazon Forecast は、データセットをトレーニングデータとテストデータに分割することで予測子を評価します。評価パラメータは、分割の実行方法と反復回数を定義します。

タイプ: [EvaluationParameters](#) オブジェクト

### [FeaturizationConfig](#)

特徴量化の設定。

タイプ: [FeaturizationConfig](#) オブジェクト

### [ForecastHorizon](#)

予測のタイムステップの数。予測期間は予測長とも呼ばれます。

タイプ: 整数

### [ForecastTypes](#)

予測子のトレーニング中に使用される予測タイプ。デフォルト値は ["0.1", "0.5", "0.9"] です

タイプ: 文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 20 項目です。

長さの制限: 最小長は 2 です。最大長は 4 です。

Pattern: (^0?\.\d\d?\$|^mean\$)

### [HPOConfig](#)

アルゴリズム用のハイパーパラメータの上書き値。

タイプ: [HyperParameterTuningJobConfig](#) オブジェクト

### [InputDataConfig](#)

予測子のトレーニングに使用するデータを含むデータセットグループを記述します。

タイプ: [InputDataConfig](#) オブジェクト

## IsAutoPredictor

予測子が [CreateAutoPredictor](#) を使用して作成されたかどうか。

型: ブール値

## LastModificationTime

リソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。

- CREATE\_PENDING (CreationTime)
- CREATE\_IN\_PROGRESS - 現在のタイムスタンプ。
- CREATE\_STOPPING - 現在のタイムスタンプ。
- CREATE\_STOPPED - ジョブが停止した時点。
- ACTIVE または CREATE\_FAILED - ジョブが終了または失敗した時点。

型: タイムスタンプ

## Message

エラーが発生した場合は、エラーに関する情報メッセージ。

型: 文字列

## OptimizationMetric

予測子を最適化するために使用される精度メトリクス。

型: 文字列

有効な値 : WAPE | RMSE | AverageWeightedQuantileLoss | MASE | MAPE

## PerformAutoML

予測子が AutoML を実行するように設定されているかどうか。

型: ブール値

## PerformHPO

予測子がハイパーパラメータ最適化 (HPO) を実行するように設定されているかどうか。

型: ブール値

## PredictorArn

予測子の ARN。

型: 文字列

長さの制限 : 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

## PredictorExecutionDetails

予測子の精度を評価するために実行されたバックテストのステータスと結果の詳細。 [CreatePredictor](#) オペレーションを呼び出すときに実行するバックテストの数を指定します。

タイプ : [PredictorExecutionDetails](#) オブジェクト

## PredictorName

予測子の名前。

型: 文字列

長さの制限 : 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

## Status

予測子のステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED

### Note

予測子を使用して予測を作成する前に、予測子の Status が ACTIVE となっている必要があります。

型: 文字列

長さの制限：最大長は 256 です。

## TrainingParameters

モデルトレーニング中に選択されたデフォルトのトレーニングパラメータまたは上書き。AutoML を実行するか、CNN-QR または DeepAR+ で HPO を選択すると、選択したハイパーパラメータの最適化された値が返されます。詳細については、「[Amazon Forecast アルゴリズム](#)」を参照してください。

型: 文字列間のマッピング

マップエントリ: 最小数は 0 項目です。最大数は 100 項目です。

キーの長さの制限: 最大長は 256 です。

キーパターン: `^[a-zA-Z0-9\-\_\.\ \\/\[\]\,\ \\\]+$`

値の長さの制限: 最大長は 256 です。

値のパターン: `^[a-zA-Z0-9\-\_\.\ \\/\[\]\,\ \"\\\s]+$`

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード：400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード：400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)



- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribePredictorBacktestExportJob

サービス: Amazon Forecast Service

[CreatePredictorBacktestExportJob](#) オペレーションを使用して作成された予測子バックテストエクスポートジョブを記述します。

このオペレーションは、CreatePredictorBacktestExportJob リクエストでユーザーによって提供されたプロパティの一覧表示に加えて、次のプロパティを一覧表示します。

- CreationTime
- LastModificationTime
- Status
- Message (エラーが発生した場合)

### リクエストの構文

```
{
  "PredictorBacktestExportJobArn": "string"
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### [PredictorBacktestExportJobArn](#)

予測子バックテストエクスポートジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

### レスポンスの構文

```
{
```

```
"CreationTime": number,
"Destination": {
  "S3Config": {
    "KMSKeyArn": "string",
    "Path": "string",
    "RoleArn": "string"
  }
},
"Format": "string",
"LastModificationTime": number,
"Message": "string",
"PredictorArn": "string",
"PredictorBacktestExportJobArn": "string",
"PredictorBacktestExportJobName": "string",
"Status": "string"
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### CreationTime

予測子バックテストエクスポートジョブが作成された時点。

型: タイムスタンプ

### Destination

エクスポートジョブの宛先。S3 パス、Amazon Forecast がロケーションにアクセスできるようにする AWS Identity and Access Management (IAM) ロール、および AWS Key Management Service (KMS) キー (オプション) を指定します。

タイプ: [DataDestination](#) オブジェクト

### Format

エクスポートされたデータの形式は CSV または PARQUET のいずれかです。

型: 文字列

長さの制限: 最大長は 7 です。

Pattern: ^CSV|PARQUET\$

### LastModificationTime

リソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。

- CREATE\_PENDING (CreationTime)
- CREATE\_IN\_PROGRESS - 現在のタイムスタンプ。
- CREATE\_STOPPING - 現在のタイムスタンプ。
- CREATE\_STOPPED - ジョブが停止した時点。
- ACTIVE または CREATE\_FAILED - ジョブが終了または失敗した時点。

型: タイムスタンプ

### Message

バックテストエクスポート中に発生した可能性のあるエラーに関する情報。

型: 文字列

### PredictorArn

予測子の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限 : 最大長は 256 です。

パターン : arn:([a-z\d-]):forecast:.\*:.\*:.\*+

### PredictorBacktestExportJobArn

予測子バックテストエクスポートジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限 : 最大長は 256 です。

パターン : arn:([a-z\d-]):forecast:.\*:.\*:.\*+

### PredictorBacktestExportJobName

予測子バックテストエクスポートジョブの名前。

型: 文字列

長さの制限：最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

## Status

予測子バックテストエクスポートジョブのステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED

型: 文字列

長さの制限：最大長は 256 です。

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード：400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード：400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)

- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeWhatIfAnalysis

サービス: Amazon Forecast Service

[CreateWhatIfAnalysis](#) オペレーションを使用して作成された What-If 分析について説明します。

このオペレーションは、CreateWhatIfAnalysis リクエストで提供されたプロパティの一覧表示に加えて、次のプロパティを一覧表示します。

- CreationTime
- LastModificationTime
- Message - エラーが発生した場合は、エラーに関する情報。
- Status

### リクエストの構文

```
{  
  "WhatIfAnalysisArn": "string"  
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### [WhatIfAnalysisArn](#)

関心がある What-If 分析の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

### レスポンスの構文

```
{  
  "CreationTime": number,  
}
```

```
"EstimatedTimeRemainingInMinutes": number,
"ForecastArn": "string",
"LastModificationTime": number,
"Message": "string",
"Status": "string",
"TimeSeriesSelector": {
  "TimeSeriesIdentifiers": {
    "DataSource": {
      "S3Config": {
        "KMSKeyArn": "string",
        "Path": "string",
        "RoleArn": "string"
      }
    },
    "Format": "string",
    "Schema": {
      "Attributes": [
        {
          "AttributeName": "string",
          "AttributeType": "string"
        }
      ]
    }
  }
},
"WhatIfAnalysisArn": "string",
"WhatIfAnalysisName": "string"
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### CreationTime

What-If 分析が作成された日時。

型: タイムスタンプ

### EstimatedTimeRemainingInMinutes

What-If 分析が完了するまでのおおよその残り時間 (分単位)。



型: 長整数

### ForecastArn

What-If 予想の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

### LastModificationTime

リソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。

- CREATE\_PENDING (CreationTime)
- CREATE\_IN\_PROGRESS - 現在のタイムスタンプ。
- CREATE\_STOPPING - 現在のタイムスタンプ。
- CREATE\_STOPPED - ジョブが停止した時点。
- ACTIVE または CREATE\_FAILED - ジョブが終了または失敗した時点。

型: タイムスタンプ

### Message

エラーが発生した場合は、エラーに関する情報メッセージ。

型: 文字列

### Status

What-If 分析のステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED

#### Note

分析にアクセスするには、What-If 分析の Status が ACTIVE である必要があります。

型: 文字列

長さの制限 : 最大長は 256 です。

パターン : `^[a-zA-Z0-9\_]+$`

### [TimeSeriesSelector](#)

TimeSeriesIdentifiers オブジェクト内の予想を作成するために使用される時系列のセットを定義します。

TimeSeriesIdentifiers オブジェクトには次の情報が必要です。

- DataSource
- Format
- Schema

タイプ : [TimeSeriesSelector](#) オブジェクト

### [WhatIfAnalysisArn](#)

What-If 分析の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限 : 最大長は 256 です。

パターン : `arn:([a-z\d-]+):forecast:.*:.*:.*+`

### [WhatIfAnalysisName](#)

What-If 分析の名前。

型: 文字列

長さの制限 : 最小長は 1 です。最大長は 63 です。

パターン : `^[a-zA-Z][a-zA-Z0-9\_]*`

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード : 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeWhatIfForecast

サービス: Amazon Forecast Service

[CreateWhatIfForecast](#) オペレーションを使用して作成された What-If 予想を記述します。

このオペレーションは、CreateWhatIfForecast リクエストで提供されたプロパティの一覧表示に加えて、次のプロパティを一覧表示します。

- CreationTime
- LastModificationTime
- Message - エラーが発生した場合は、エラーに関する情報。
- Status

### リクエストの構文

```
{  
  "WhatIfForecastArn": "string"  
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### [WhatIfForecastArn](#)

関心がある What-If 予想の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 300 です。

Pattern: arn:([a-z\d-]+):forecast:.\*:.\*:.\*+

必須: はい

### レスポンスの構文

```
{  
  "CreationTime": number,  
  "EstimatedTimeRemainingInMinutes": number,  
  "ForecastTypes": [ "string" ],  
}
```

```
"LastModificationTime": number,
"Message": "string",
"Status": "string",
"TimeSeriesReplacementsDataSource": {
  "Format": "string",
  "S3Config": {
    "KMSKeyArn": "string",
    "Path": "string",
    "RoleArn": "string"
  },
  "Schema": {
    "Attributes": [
      {
        "AttributeName": "string",
        "AttributeType": "string"
      }
    ]
  },
  "TimestampFormat": "string"
},
"TimeSeriesTransformations": [
  {
    "Action": {
      "AttributeName": "string",
      "Operation": "string",
      "Value": number
    },
    "TimeSeriesConditions": [
      {
        "AttributeName": "string",
        "AttributeValue": "string",
        "Condition": "string"
      }
    ]
  }
],
"WhatIfAnalysisArn": "string",
"WhatIfForecastArn": "string",
"WhatIfForecastName": "string"
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### CreationTime

What-If 予想が作成された日時。

型: タイムスタンプ

### EstimatedTimeRemainingInMinutes

What-If 予想を完了するまでのおおよその残り時間 (分単位)。

型: 長整数

### ForecastTypes

確率的予測が生成される分位数。 [CreateWhatIfForecast](#) オペレーションでの What-If 予想ごとに最大 5 つの分位を指定できます。分位数を指定しなかった場合、デフォルト値は ["0.1", "0.5", "0.9"] です。

タイプ: 文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 20 項目です。

長さの制限: 最小長は 2 です。最大長は 4 です。

Pattern: (^0?\. \d\d? \$|^mean\$)

### LastModificationTime

リソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。

- CREATE\_PENDING (CreationTime)
- CREATE\_IN\_PROGRESS - 現在のタイムスタンプ。
- CREATE\_STOPPING - 現在のタイムスタンプ。
- CREATE\_STOPPED - ジョブが停止した時点。
- ACTIVE または CREATE\_FAILED - ジョブが終了または失敗した時点。

型: タイムスタンプ

### Message

エラーが発生した場合は、エラーに関する情報メッセージ。

型: 文字列

## Status

What-If 予想のステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED

### Note

予想にアクセスするには、What-If 予想の Status が ACTIVE である必要があります。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `^[a-zA-Z0-9\_]+$`

## TimeSeriesReplacementsDataSource

置換時系列を表す S3Config、Schema、Format 要素の配列。

タイプ: [TimeSeriesReplacementsDataSource](#) オブジェクト

## TimeSeriesTransformations

どの時系列にどの変換が適用されたかを記述する Action および TimeSeriesConditions 要素の配列。

タイプ: [TimeSeriesTransformation](#) オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 30 項目です。

## WhatIfAnalysisArn

この What-If 予想を含む分析の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]):forecast:.*:.*:.*`

## WhatIfForecastArn

What-If 予想の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 300 です。

Pattern: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

## WhatIfForecastName

What-If 予想の名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード: 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)



- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeWhatIfForecastExport

サービス: Amazon Forecast Service

[CreateWhatIfForecastExport](#) オペレーションを使用して作成された What-If 予想エクスポートを記述します。

このオペレーションは、`CreateWhatIfForecastExport` リクエストで提供されたプロパティの一覧表示に加えて、次のプロパティを一覧表示します。

- `CreationTime`
- `LastModificationTime`
- `Message` - エラーが発生した場合は、エラーに関する情報。
- `Status`

### リクエストの構文

```
{  
  "WhatIfForecastExportArn": "string"  
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### [WhatIfForecastExportArn](#)

関心がある What-If 予想エクスポートの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 300 です。

Pattern: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

### レスポンスの構文

```
{  
  "CreationTime": number,  
  ...  
}
```

```
"Destination": {
  "S3Config": {
    "KMSKeyArn": "string",
    "Path": "string",
    "RoleArn": "string"
  }
},
"EstimatedTimeRemainingInMinutes": number,
"Format": "string",
"LastModificationTime": number,
"Message": "string",
"Status": "string",
"WhatIfForecastArns": [ "string" ],
"WhatIfForecastExportArn": "string",
"WhatIfForecastExportName": "string"
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### CreationTime

What-If 予想エクスポートが作成された日時。

型: タイムスタンプ

### Destination

エクスポートジョブの宛先。S3 パス、Amazon Forecast がロケーションにアクセスできるようにする AWS Identity and Access Management (IAM) ロール、および AWS Key Management Service (KMS) キー (オプション) を指定します。

タイプ: [DataDestination](#) オブジェクト

### EstimatedTimeRemainingInMinutes

What-If 予想エクスポートを完了するまでのおおよその残り時間 (分単位)。

型: 長整数

### Format

エクスポートされたデータの形式は CSV または PARQUET のいずれかです。

型: 文字列

長さの制限: 最大長は 7 です。

Pattern: ^CSV|PARQUET\$

### LastModificationTime

リソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。

- CREATE\_PENDING (CreationTime)
- CREATE\_IN\_PROGRESS - 現在のタイムスタンプ。
- CREATE\_STOPPING - 現在のタイムスタンプ。
- CREATE\_STOPPED - ジョブが停止した時点。
- ACTIVE または CREATE\_FAILED - ジョブが終了または失敗した時点。

型: タイムスタンプ

### Message

エラーが発生した場合は、エラーに関する情報メッセージ。

型: 文字列

### Status

What-If 予想のステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED

#### Note

予想エクスポートにアクセスするには、What-If 予想エクスポートの Status が ACTIVE である必要があります。

型: 文字列

長さの制限: 最大長は 256 です。

## WhatIfForecastArns

このリソースにエクスポートされるすべての What-If 予想を表す Amazon リソースネーム (ARN) の配列。

タイプ: 文字列の配列

長さの制限: 最大長は 300 です。

Pattern: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

## WhatIfForecastExportArn

What-If 予想エクスポートの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 300 です。

Pattern: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

## WhatIfForecastExportName

What-If 予想エクスポートの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード: 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## GetAccuracyMetrics

サービス: Amazon Forecast Service

[CreatePredictor](#) オペレーションによってトレーニングされたモデルの精度に関するメトリクスを提供します。メトリクスを使用して、モデルのパフォーマンスを確認し、予測子を使用して予測を生成するかどうかを決定します。詳細については、「[Predictor Metrics](#)」(予測子メトリクス)を参照してください。

このオペレーションにより、評価された各バックテストウィンドウのメトリクスが生成されます。バックテストウィンドウの数 (NumberOfBacktestWindows) は、[EvaluationParameters](#) オブジェクトを使用して指定されます。このオブジェクトは、オプションで CreatePredictor リクエストに含まれます。NumberOfBacktestWindows が指定されていない場合、番号はデフォルトで 1 になります。

filling メソッドのパラメータは、どの項目がメトリクスに寄与するかを決定します。すべての項目が寄与するようにするには、zero を指定します。評価対象の範囲内の完全なデータを持つ項目のみが寄与するようにするには、nan を指定します。詳細については、「[FeaturizationMethod](#)」を参照してください。

### Note

精度メトリクスを取得する前に、予測子の Status が ACTIVE になっている必要があります。これは、トレーニングが完了したことを意味します。[DescribePredictor](#) オペレーションを使用して、ステータスを取得します。

### リクエストの構文

```
{
  "PredictorArn": "string"
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### [PredictorArn](#)

メトリクスを取得する予測子の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

## レスポンスの構文

```
{
  "AutoMLOverrideStrategy": "string",
  "IsAutoPredictor": boolean,
  "OptimizationMetric": "string",
  "PredictorEvaluationResults": [
    {
      "AlgorithmArn": "string",
      "TestWindows": [
        {
          "EvaluationType": "string",
          "ItemCount": number,
          "Metrics": {
            "AverageWeightedQuantileLoss": number,
            "ErrorMetrics": [
              {
                "ForecastType": "string",
                "MAPE": number,
                "MASE": number,
                "RMSE": number,
                "WAPE": number
              }
            ],
            "RMSE": number,
            "WeightedQuantileLosses": [
              {
                "LossValue": number,
                "Quantile": number
              }
            ]
          }
        },
        {
          "TestWindowEnd": number,
          "TestWindowStart": number
        }
      ]
    }
  ]
}
```



```
    ]  
  }  
]  
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### [AutoMLOverrideStrategy](#)

#### Note

LatencyOptimized AutoML 上書き戦略は、プライベートベータでのみ使用できます。アクセス権限の詳細については、AWS サポートまたはアカウントマネージャーにお問い合わせください。

予測子のトレーニングに使用する AutoML 戦略。LatencyOptimized が指定されていない限り、AutoML 戦略は予測子の精度を最適化します。

このパラメータは、AutoML を使用してトレーニングした予測子についてのみ有効です。

型: 文字列

有効な値 : LatencyOptimized | AccuracyOptimized

### [IsAutoPredictor](#)

予測子が [CreateAutoPredictor](#) を使用して作成されたかどうか。

型: ブール値

### [OptimizationMetric](#)

予測子を最適化するために使用される精度メトリクス。

型: 文字列

有効な値 : WAPE | RMSE | AverageWeightedQuantileLoss | MASE | MAPE

## [PredictorEvaluationResults](#)

予測子の評価から得られた結果の配列。

型: [EvaluationResult](#) オブジェクトの配列

### エラー

#### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード : 400

#### ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

#### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)

- [AWS SDK for Ruby V3](#)

## ListDatasetGroups

サービス: Amazon Forecast Service

[CreateDatasetGroup](#) オペレーションを使用して作成されたデータセットグループのリストを返します。このオペレーションは、各データセットグループについて、Amazon リソースネーム (ARN) を含むプロパティの概要を返します。[DescribeDatasetGroup](#) オペレーションでデータセットグループ ARN を使用して、プロパティの完全なセットを取得できます。

リクエストの構文

```
{
  "MaxResults": number,
  "NextToken": "string"
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### [MaxResults](#)

レスポンスで返される項目の数。

タイプ: 整数

有効範囲: 最小値は 1 です。最大値は 100 です。

必須: いいえ

### [NextToken](#)

前のリクエストの結果が切り捨てられた場合、レスポンスには NextToken が含まれます。次の結果セットを取得するには、次のリクエストでトークンを使用します。トークンの有効期間は 24 時間です。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

Pattern: .+

必須: いいえ

## レスポンスの構文

```
{
  "DatasetGroups": [
    {
      "CreationTime": number,
      "DatasetGroupArn": "string",
      "DatasetGroupName": "string",
      "LastModificationTime": number
    }
  ],
  "NextToken": "string"
}
```

### レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

#### DatasetGroups

各データセットグループのプロパティを要約するオブジェクトの配列。

型: DatasetGroupSummary オブジェクトの配列

#### NextToken

レスポンスが切り捨てられた場合、Amazon Forecast はこのトークンを返します。次の結果セットを取得するには、次のリクエストでトークンを使用します。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

パターン: .+

### エラー

#### InvalidNextTokenException

トークンが無効です。トークンの有効期間は 24 時間です。

HTTP ステータスコード : 400

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListDatasetImportJobs

サービス: Amazon Forecast Service

[CreateDatasetImportJob](#) オペレーションを使用して作成されたデータセットのインポートジョブのリストを返します。このオペレーションは、各インポートジョブについて、Amazon リソースネーム (ARN) を含むプロパティの概要を返します。[DescribeDatasetImportJob](#) オペレーションで ARN を使用して、プロパティの完全なセットを取得できます。[Filter](#) オブジェクトの配列を提供することにより、リストをフィルタリングできます。

### リクエストの構文

```
{
  "Filters": [
    {
      "Condition": "string",
      "Key": "string",
      "Value": "string"
    }
  ],
  "MaxResults": number,
  "NextToken": "string"
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### Filters

フィルターの配列。各フィルターについて、条件および match ステートメントを指定します。条件は IS または IS\_NOT のいずれかであり、それぞれについて、ステートメントに一致するデータセットをリストに含めるか除外するかを指定します。match ステートメントは、キーと値で構成されます。

#### フィルタープロパティ

- **Condition** - 適用する条件。有効な値は、IS および IS\_NOT です。ステートメントに一致するデータセットを含めるには、IS を指定します。一致するデータセットを除外するには、IS\_NOT を指定します。
- **Key** - フィルタリングするパラメータの名前。有効な値は、DatasetArn および Status です。

- Value - 照合する値。

例えば、ステータスが [ACTIVE] であるすべてのデータセットのインポートジョブを一覧表示するには、次のフィルターを指定します。

```
"Filters": [ { "Condition": "IS", "Key": "Status", "Value": "ACTIVE" } ]
```

型: [Filter](#) オブジェクトの配列

必須: いいえ

### [MaxResults](#)

レスポンスで返される項目の数。

タイプ: 整数

有効範囲: 最小値は 1 です。最大値は 100 です。

必須: いいえ

### [NextToken](#)

前のリクエストの結果が切り捨てられた場合、レスポンスには NextToken が含まれます。次の結果セットを取得するには、次のリクエストでトークンを使用します。トークンの有効期間は 24 時間です。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

Pattern: .+

必須: いいえ

### レスポンスの構文

```
{
  "DatasetImportJobs": [
    {
      "CreationTime": number,
      "DatasetImportJobArn": "string",
      "DatasetImportJobName": "string",
      "DataSource": {
```



```
    "S3Config": {
      "KMSKeyArn": "string",
      "Path": "string",
      "RoleArn": "string"
    },
    "ImportMode": "string",
    "LastModificationTime": number,
    "Message": "string",
    "Status": "string"
  },
  "NextToken": "string"
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### [DatasetImportJobs](#)

各データセットのインポートジョブのプロパティを要約するオブジェクトの配列。

型: [DatasetImportJobSummary](#) オブジェクトの配列

### [NextToken](#)

レスポンスが切り捨てられた場合、Amazon Forecast はこのトークンを返します。次の結果セットを取得するには、次のリクエストでトークンを使用します。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

パターン: .+

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード : 400

InvalidNextTokenException

トークンが無効です。トークンの有効期間は 24 時間です。

HTTP ステータスコード : 400

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListDatasets

サービス: Amazon Forecast Service

[CreateDataset](#) オペレーションを使用して作成されたデータセットのリストを返します。各データセットについて、Amazon リソースネーム (ARN) を含むプロパティの概要が返されます。プロパティの完全なセットを取得するには、[DescribeDataset](#) オペレーションで ARN を使用します。

リクエストの構文

```
{  
  "MaxResults": number,  
  "NextToken": "string"  
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### [MaxResults](#)

レスポンスで返される項目の数。

タイプ: 整数

有効範囲: 最小値は 1 です。最大値は 100 です。

必須: いいえ

### [NextToken](#)

前のリクエストの結果が切り捨てられた場合、レスポンスには NextToken が含まれます。次の結果セットを取得するには、次のリクエストでトークンを使用します。トークンの有効期間は 24 時間です。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

Pattern: .+

必須: いいえ

## レスポンスの構文

```
{
  "Datasets": [
    {
      "CreationTime": number,
      "DatasetArn": "string",
      "DatasetName": "string",
      "DatasetType": "string",
      "Domain": "string",
      "LastModificationTime": number
    }
  ],
  "NextToken": "string"
}
```

### レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

#### Datasets

各データセットのプロパティを要約するオブジェクトの配列。

型: [DatasetSummary](#) オブジェクトの配列

#### NextToken

レスポンスが切り捨てられた場合、Amazon Forecast はこのトークンを返します。次の結果セットを取得するには、次のリクエストでトークンを使用します。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

パターン: .+

### エラー

#### InvalidNextTokenException

トークンが無効です。トークンの有効期間は 24 時間です。

HTTP ステータスコード : 400

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListExplainabilities

サービス: Amazon Forecast Service

[CreateExplainability](#) オペレーションを使用して作成された Explainability リソースのリストを返します。このオペレーションは、各 Explainability に関する要約を返します。[Filter](#) オブジェクトの配列を使用してリストをフィルタリングできます。

特定の Explainability リソースのプロパティの完全なセットを取得するには、[DescribeExplainability](#) オペレーションで ARN を使用します。

### リクエストの構文

```
{
  "Filters": [
    {
      "Condition": "string",
      "Key": "string",
      "Value": "string"
    }
  ],
  "MaxResults": number,
  "NextToken": "string"
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### [Filters](#)

フィルターの配列。各フィルターについて、条件および match ステートメントを指定します。条件は IS または IS\_NOT のいずれかであり、ステートメントに一致するリソースをリストに含めるか除外するかを指定します。match ステートメントは、キーと値で構成されます。

#### フィルタープロパティ

- Condition - 適用する条件。有効な値は、IS および IS\_NOT です。
- Key - フィルタリングするパラメータの名前。有効な値は、ResourceArn および Status です。
- Value - 照合する値。

型: [Filter](#) オブジェクトの配列

必須: いいえ

### MaxResults

レスポンスで返された項目の数。

タイプ: 整数

有効範囲: 最小値は 1 です。最大値は 100 です。

必須: いいえ

### NextToken

前のリクエストの結果が切り捨てられた場合、レスポンスには `NextToken` が含まれます。次の結果セットを取得するには、次のリクエストでトークンを使用します。トークンの有効期間は 24 時間です。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

Pattern: `.+`

必須: いいえ

## レスポンスの構文

```
{
  "Explainabilities": [
    {
      "CreationTime": number,
      "ExplainabilityArn": "string",
      "ExplainabilityConfig": {
        "TimePointGranularity": "string",
        "TimeSeriesGranularity": "string"
      },
      "ExplainabilityName": "string",
      "LastModificationTime": number,
      "Message": "string",
      "ResourceArn": "string",
      "Status": "string"
    }
  ],
}
```

```
"NextToken": "string"  
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### Explainabilities

各 Explainability リソースのプロパティを要約するオブジェクトの配列。

型: [ExplainabilitySummary](#) オブジェクトの配列

### NextToken

レスポンスが切り捨てられた場合、このトークンを返します。次の結果セットを取得するには、次のリクエストでトークンを使用します。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

パターン: .+

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

### InvalidNextTokenException

トークンが無効です。トークンの有効期間は 24 時間です。

HTTP ステータスコード: 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。



- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListExplainabilityExports

サービス: Amazon Forecast Service

[CreateExplainabilityExport](#) オペレーションを使用して作成された Explainability エクスポートのリストを返します。このオペレーションは、各 Explainability のエクスポートに関する要約を返します。[Filter](#) オブジェクトの配列を使用してリストをフィルタリングできます。

特定の Explainability エクスポートのプロパティの完全なセットを取得するには、[DescribeExplainability](#) オペレーションで ARN を使用します。

### リクエストの構文

```
{
  "Filters": [
    {
      "Condition": "string",
      "Key": "string",
      "Value": "string"
    }
  ],
  "MaxResults": number,
  "NextToken": "string"
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### [Filters](#)

フィルターの配列。各フィルターについて、条件および match ステートメントを指定します。条件は IS または IS\_NOT のいずれかであり、ステートメントに一致するリソースをリストに含めるか除外するかを指定します。match ステートメントは、キーと値で構成されます。

#### フィルタープロパティ

- Condition - 適用する条件。有効な値は、IS および IS\_NOT です。
- Key - フィルタリングするパラメータの名前。有効な値は、ResourceArn および Status です。
- Value - 照合する値。

型: [Filter](#) オブジェクトの配列

必須: いいえ

### MaxResults

レスポンスで返される項目の数。

タイプ: 整数

有効範囲: 最小値は 1 です。最大値は 100 です。

必須: いいえ

### NextToken

前のリクエストの結果が切り捨てられた場合、レスポンスにはが含まれます NextToken。次の結果セットを取得するには、次のリクエストでトークンを使用します。トークンの有効期間は 24 時間です。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

Pattern: .+

必須: いいえ

## レスポンスの構文

```
{
  "ExplainabilityExports": [
    {
      "CreationTime": number,
      "Destination": {
        "S3Config": {
          "KMSKeyArn": "string",
          "Path": "string",
          "RoleArn": "string"
        }
      },
    },
    "ExplainabilityExportArn": "string",
    "ExplainabilityExportName": "string",
    "LastModificationTime": number,
    "Message": "string",
  ]
}
```

```
    "Status": "string"  
  }  
],  
"NextToken": "string"  
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### [ExplainabilityExports](#)

各 Explainability エクスポートのプロパティを要約するオブジェクトの配列。

型: [ExplainabilityExportSummary](#) オブジェクトの配列

### [NextToken](#)

レスポンスが切り捨てられた場合、このトークンを返します。次の結果セットを取得するには、次のリクエストでトークンを使用します。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

パターン: .+

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

### InvalidNextTokenException

トークンが無効です。トークンの有効期間は 24 時間です。

HTTP ステータスコード: 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListForecastExportJobs

サービス: Amazon Forecast Service

[CreateForecastExportJob](#) オペレーションを使用して作成された予測エクスポートジョブのリストを返します。このオペレーションは、各予測エクスポートジョブについて、Amazon リソースネーム (ARN) を含むプロパティの概要を返します。プロパティの完全なセットを取得するには、[DescribeForecastExportJob](#) オペレーションで ARN を使用します。[Filter](#) オブジェクトの配列を使用してリストをフィルタリングできます。

### リクエストの構文

```
{
  "Filters": [
    {
      "Condition": "string",
      "Key": "string",
      "Value": "string"
    }
  ],
  "MaxResults": number,
  "NextToken": "string"
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### Filters

フィルターの配列。各フィルターについて、条件および match ステートメントを指定します。条件は IS または IS\_NOT のいずれかであり、それぞれについて、ステートメントに一致する予測エクスポートジョブをリストに含めるか除外するかを指定します。match ステートメントは、キーと値で構成されます。

#### フィルタープロパティ

- **Condition** - 適用する条件。有効な値は、IS および IS\_NOT です。ステートメントに一致する予測エクスポートジョブを含めるには、IS を指定します。一致する予測エクスポートジョブを除外するには、IS\_NOT を指定します。
- **Key** - フィルタリングするパラメータの名前。有効な値は、ForecastArn および Status です。

- Value - 照合する値。

例えば、electricityforecast という名前の予測をエクスポートするすべてのジョブを一覧表示するには、次のフィルターを指定します。

```
"Filters": [ { "Condition": "IS", "Key": "ForecastArn", "Value":  
"arn:aws:forecast:us-west-2:<acct-id>:forecast/electricityforecast" } ]
```

型: [Filter](#) オブジェクトの配列

必須: いいえ

### [MaxResults](#)

レスポンスで返される項目の数。

タイプ: 整数

有効範囲: 最小値は 1 です。最大値は 100 です。

必須: いいえ

### [NextToken](#)

前のリクエストの結果が切り捨てられた場合、レスポンスには NextToken が含まれます。次の結果セットを取得するには、次のリクエストでトークンを使用します。トークンの有効期間は 24 時間です。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

Pattern: .+

必須: いいえ

### レスポンスの構文

```
{  
  "ForecastExportJobs": [  
    {  
      "CreationTime": number,  
      "Destination": {  
        "S3Config": {
```

```
        "KMSKeyArn": "string",
        "Path": "string",
        "RoleArn": "string"
    }
},
"ForecastExportJobArn": "string",
"ForecastExportJobName": "string",
"LastModificationTime": number,
"Message": "string",
"Status": "string"
}
],
"NextToken": "string"
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### [ForecastExportJobs](#)

各エクスポートジョブのプロパティを要約するオブジェクトの配列。

型: [ForecastExportJobSummary](#) オブジェクトの配列

### [NextToken](#)

レスポンスが切り捨てられた場合、Amazon Forecast はこのトークンを返します。次の結果セットを取得するには、次のリクエストでトークンを使用します。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

パターン: .+

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。



HTTP ステータスコード : 400

InvalidNextTokenException

トークンが無効です。トークンの有効期間は 24 時間です。

HTTP ステータスコード : 400

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListForecasts

サービス: Amazon Forecast Service

[CreateForecast](#) オペレーションを使用して作成された予測のリストを返します。このオペレーションは、各予測について、Amazon リソースネーム (ARN) を含むプロパティの概要を返します。プロパティの完全なセットを取得するには、[DescribeForecast](#) オペレーションで ARN を指定します。[Filter](#) オブジェクトの配列を使用してリストをフィルタリングできます。

リクエストの構文

```
{
  "Filters": [
    {
      "Condition": "string",
      "Key": "string",
      "Value": "string"
    }
  ],
  "MaxResults": number,
  "NextToken": "string"
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### Filters

フィルターの配列。各フィルターについて、条件および match ステートメントを指定します。条件は IS または IS\_NOT のいずれかであり、それぞれについて、ステートメントに一致する予測をリストに含めるか除外するかを指定します。match ステートメントは、キーと値で構成されます。

フィルタープロパティ

- Condition - 適用する条件。有効な値は、IS および IS\_NOT です。ステートメントに一致する予測を含めるには、IS を指定します。一致する予測を除外するには、IS\_NOT を指定します。
- Key - フィルタリングするパラメータの名前。有効な値は、DatasetGroupArn、PredictorArn、Status です。
- Value - 照合する値。

例えば、ステータスが [ACTIVE] ではないすべての予測を一覧表示するには、次のように指定します。

```
"Filters": [ { "Condition": "IS_NOT", "Key": "Status", "Value": "ACTIVE" } ]
```

型: [Filter](#) オブジェクトの配列

必須: いいえ

### [MaxResults](#)

レスポンスで返される項目の数。

タイプ: 整数

有効範囲: 最小値は 1 です。最大値は 100 です。

必須: いいえ

### [NextToken](#)

前のリクエストの結果が切り捨てられた場合、レスポンスには NextToken が含まれます。次の結果セットを取得するには、次のリクエストでトークンを使用します。トークンの有効期間は 24 時間です。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

Pattern: .+

必須: いいえ

### レスポンスの構文

```
{
  "Forecasts": [
    {
      "CreatedUsingAutoPredictor": boolean,
      "CreationTime": number,
      "DatasetGroupArn": "string",
      "ForecastArn": "string",
```

```
    "ForecastName": "string",
    "LastModificationTime": number,
    "Message": "string",
    "PredictorArn": "string",
    "Status": "string"
  }
],
"NextToken": "string"
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### Forecasts

各予測のプロパティを要約するオブジェクトの配列。

型: [ForecastSummary](#) オブジェクトの配列

### NextToken

レスポンスが切り捨てられた場合、Amazon Forecast はこのトークンを返します。次の結果セットを取得するには、次のリクエストでトークンを使用します。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

パターン: .+

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

### InvalidNextTokenException

トークンが無効です。トークンの有効期間は 24 時間です。

HTTP ステータスコード : 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListMonitorEvaluations

サービス: Amazon Forecast Service

さまざまな期間にモニターリソースによって収集されたモニタリング評価結果と予測イベントのリストを返します。

モニタリングの詳細については、「[予測子のモニタリング](#)」を参照してください。モニタリング結果の取得について詳しくは、「[Viewing Monitoring Results](#)」を参照してください。

### リクエストの構文

```
{
  "Filters": [
    {
      "Condition": "string",
      "Key": "string",
      "Value": "string"
    }
  ],
  "MaxResults": number,
  "MonitorArn": "string",
  "NextToken": "string"
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### Filters

フィルターの配列。各フィルターについて、条件および match ステートメントを指定します。条件は IS または IS\_NOT のいずれかであり、ステートメントに一致するリソースをリストに含めるか除外するかを指定します。match ステートメントは、キーと値で構成されます。

#### フィルタープロパティ

- Condition - 適用する条件。有効な値は、IS および IS\_NOT です。
- Key - フィルタリングするパラメータの名前。唯一の有効な値は EvaluationState です。
- Value - 照合する値。有効な値は SUCCESS または FAILURE のみです。

例えば、成功したモニタリング評価のみを一覧表示するには、次のように指定します。

```
"Filters": [ { "Condition": "IS", "Key": "EvaluationState", "Value": "SUCCESS" } ]
```

型: [Filter](#) オブジェクトの配列

必須: いいえ

### [MaxResults](#)

返されるモニタリング結果の最大数。

タイプ: 整数

有効範囲: 最小値は 1 です。最大値は 100 です。

必須: いいえ

### [MonitorArn](#)

結果を取得するモニターリソースの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

### [NextToken](#)

前のリクエストの結果が切り捨てられた場合、レスポンスには NextToken が含まれます。次の結果セットを取得するには、次のリクエストでトークンを使用します。トークンの有効期間は 24 時間です。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

Pattern: `.+`

必須: いいえ

### レスポンスの構文

```
{
```

```
"NextToken": "string",
"PredictorMonitorEvaluations": [
  {
    "EvaluationState": "string",
    "EvaluationTime": number,
    "Message": "string",
    "MetricResults": [
      {
        "MetricName": "string",
        "MetricValue": number
      }
    ],
    "MonitorArn": "string",
    "MonitorDataSource": {
      "DatasetImportJobArn": "string",
      "ForecastArn": "string",
      "PredictorArn": "string"
    },
    "NumItemsEvaluated": number,
    "PredictorEvent": {
      "Datetime": number,
      "Detail": "string"
    },
    "ResourceArn": "string",
    "WindowEndDatetime": number,
    "WindowStartDatetime": number
  }
]
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### NextToken

レスポンスが切り捨てられた場合、Amazon Forecast はこのトークンを返します。次の結果セットを取得するには、次のリクエストでトークンを使用します。トークンの有効期間は 24 時間です。

型: 文字列



長さの制限：最小長は 1 です。最大長は 3000 です。

Pattern: .+

## PredictorMonitorEvaluations

さまざまな期間にモニターリソースによって収集されたモニタリング結果と予測イベント。

モニタリングについて詳しくは、「[Viewing Monitoring Results](#)」を参照してください。モニタリング結果の取得について詳しくは、「[Viewing Monitoring Results](#)」を参照してください。

型: [PredictorMonitorEvaluation](#) オブジェクトの配列

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード：400

### InvalidNextTokenException

トークンが無効です。トークンの有効期間は 24 時間です。

HTTP ステータスコード：400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード：400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)

- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListMonitors

サービス: Amazon Forecast Service

[CreateMonitor](#) オペレーションと [CreateAutoPredictor](#) オペレーションで作成されたモニターのリストを返します。このオペレーションは、各モニターリソースについて、Amazon リソースネーム (ARN) を含むプロパティの概要を返します。[DescribeMonitor](#) 操作でモニターの ARN を指定することで、モニターリソースのプロパティの完全なセットを取得できます。

### リクエストの構文

```
{
  "Filters": [
    {
      "Condition": "string",
      "Key": "string",
      "Value": "string"
    }
  ],
  "MaxResults": number,
  "NextToken": "string"
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### [Filters](#)

フィルターの配列。各フィルターについて、条件および match ステートメントを指定します。条件は IS または IS\_NOT のいずれかであり、ステートメントに一致するリソースをリストに含めるか除外するかを指定します。match ステートメントは、キーと値で構成されます。

#### フィルタープロパティ

- Condition - 適用する条件。有効な値は、IS および IS\_NOT です。
- Key - フィルタリングするパラメータの名前。唯一の有効な値は Status です。
- Value - 照合する値。

例えば、ステータスが [ACTIVE] であるすべてのモニターを一覧表示するには、次のように指定します。

```
"Filters": [ { "Condition": "IS", "Key": "Status", "Value": "ACTIVE" } ]
```

型: [Filter](#) オブジェクトの配列

必須: いいえ

### [MaxResults](#)

レスポンスに含めるモニターの最大数。

タイプ: 整数

有効範囲: 最小値は 1 です。最大値は 100 です。

必須: いいえ

### [NextToken](#)

前のリクエストの結果が切り捨てられた場合、レスポンスには `NextToken` が含まれます。次の結果セットを取得するには、次のリクエストでトークンを使用します。トークンの有効期間は 24 時間です。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

Pattern: `.+`

必須: いいえ

## レスポンスの構文

```
{
  "Monitors": [
    {
      "CreationTime": number,
      "LastModificationTime": number,
      "MonitorArn": "string",
      "MonitorName": "string",
      "ResourceArn": "string",
      "Status": "string"
    }
  ],
  "NextToken": "string"
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### Monitors

各モニターのプロパティを要約するオブジェクトの配列。

型: [MonitorSummary](#) オブジェクトの配列

### NextToken

レスポンスが切り捨てられた場合、Amazon Forecast はこのトークンを返します。次の結果セットを取得するには、次のリクエストでトークンを使用します。

型: 文字列

長さの制限 : 最小長は 1 です。最大長は 3000 です。

パターン : .+

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード : 400

### InvalidNextTokenException

トークンが無効です。トークンの有効期間は 24 時間です。

HTTP ステータスコード : 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)

- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListPredictorBacktestExportJobs

サービス: Amazon Forecast Service

[CreatePredictorBacktestExportJob](#) オペレーションを使用して作成された予測子バックテストエクスポートジョブのリストを返します。このオペレーションは、各バックテストエクスポートジョブに関する要約を返します。[Filter](#) オブジェクトの配列を使用してリストをフィルタリングできます。

特定のバックテストエクスポートジョブのプロパティの完全なセットを取得するには、[DescribePredictorBacktestExportJob](#) オペレーションで ARN を使用します。

### リクエストの構文

```
{
  "Filters": [
    {
      "Condition": "string",
      "Key": "string",
      "Value": "string"
    }
  ],
  "MaxResults": number,
  "NextToken": "string"
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### [Filters](#)

フィルターの配列。各フィルターについて、条件および match ステートメントを指定します。条件は IS または IS\_NOT のいずれかであり、ステートメントに一致する予測子バックテストエクスポートジョブをリストに含めるか除外するかを指定します。match ステートメントは、キーと値で構成されます。

#### フィルタープロパティ

- **Condition** - 適用する条件。有効な値は、IS および IS\_NOT です。ステートメントに一致する予測子バックテストエクスポートジョブを含めるには、IS を指定します。一致する予測子バックテストエクスポートジョブを除外するには、IS\_NOT を指定します。
- **Key** - フィルタリングするパラメータの名前。有効な値は、PredictorArn および Status です。

- Value - 照合する値。

型: [Filter](#) オブジェクトの配列

必須: いいえ

### [MaxResults](#)

レスポンスで返される項目の数。

タイプ: 整数

有効範囲: 最小値は 1 です。最大値は 100 です。

必須: いいえ

### [NextToken](#)

前のリクエストの結果が切り捨てられた場合、レスポンスにはが含まれます NextToken。次の結果セットを取得するには、次のリクエストでトークンを使用します。トークンの有効期間は 24 時間です。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

Pattern: .+

必須: いいえ

## レスポンスの構文

```
{
  "NextToken": "string",
  "PredictorBacktestExportJobs": [
    {
      "CreationTime": number,
      "Destination": {
        "S3Config": {
          "KMSKeyArn": "string",
          "Path": "string",
          "RoleArn": "string"
        }
      }
    }
  ]
}
```



```
    },
    "LastModificationTime": number,
    "Message": "string",
    "PredictorBacktestExportJobArn": "string",
    "PredictorBacktestExportJobName": "string",
    "Status": "string"
  }
]
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### [NextToken](#)

レスポンスが切り捨てられた場合、このトークンを返します。次の結果セットを取得するには、次のリクエストでトークンを使用します。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

Pattern: .+

### [PredictorBacktestExportJobs](#)

各予測子バックテストエクスポートジョブのプロパティを要約するオブジェクトの配列。

型: [PredictorBacktestExportJobSummary](#) オブジェクトの配列

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

### InvalidNextTokenException

トークンが無効です。トークンの有効期間は 24 時間です。

HTTP ステータスコード : 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListPredictors

サービス: Amazon Forecast Service

[CreateAutoPredictor](#) または [CreatePredictor](#) オペレーションを使用して作成された予測子のリストを返します。このオペレーションは、各予測子について、Amazon リソースネーム (ARN) を含むプロパティの概要を返します。

[DescribeAutoPredictor](#) および [DescribePredictor](#) オペレーションで ARN を使用すると、プロパティの完全なセットを取得できます。[Filter](#) オブジェクトの配列を使用してリストをフィルタリングできます。

### リクエストの構文

```
{
  "Filters": [
    {
      "Condition": "string",
      "Key": "string",
      "Value": "string"
    }
  ],
  "MaxResults": number,
  "NextToken": "string"
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### [Filters](#)

フィルターの配列。各フィルターについて、条件および match ステートメントを指定します。条件は IS または IS\_NOT のいずれかであり、それぞれについて、ステートメントに一致する予測子をリストに含めるか除外するかを指定します。match ステートメントは、キーと値で構成されます。

#### フィルタープロパティ

- **Condition** - 適用する条件。有効な値は、IS および IS\_NOT です。ステートメントに一致する予測子を含めるには、IS を指定します。一致する予測子を除外するには、IS\_NOT を指定します。

- Key - フィルタリングするパラメータの名前。有効な値は、DatasetGroupArn および Status です。
- Value - 照合する値。

例えば、ステータスが [ACTIVE] であるすべての予測子を一覧表示するには、次のように指定します。

```
"Filters": [ { "Condition": "IS", "Key": "Status", "Value": "ACTIVE" } ]
```

型: [Filter](#) オブジェクトの配列

必須: いいえ

### [MaxResults](#)

レスポンスで返される項目の数。

タイプ: 整数

有効範囲: 最小値は 1 です。最大値は 100 です。

必須: いいえ

### [NextToken](#)

前のリクエストの結果が切り捨てられた場合、レスポンスには NextToken が含まれます。次の結果セットを取得するには、次のリクエストでトークンを使用します。トークンの有効期間は 24 時間です。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

Pattern: .+

必須: いいえ

### レスポンスの構文

```
{  
  "NextToken": "string",  
  "Predictors": [  
    {
```

```
    "CreationTime": number,
    "DatasetGroupArn": "string",
    "IsAutoPredictor": boolean,
    "LastModificationTime": number,
    "Message": "string",
    "PredictorArn": "string",
    "PredictorName": "string",
    "ReferencePredictorSummary": {
      "Arn": "string",
      "State": "string"
    },
    "Status": "string"
  }
]
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### [NextToken](#)

レスポンスが切り捨てられた場合、Amazon Forecast はこのトークンを返します。次の結果セットを取得するには、次のリクエストでトークンを使用します。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

Pattern: .+

### [Predictors](#)

各予測子のプロパティを要約するオブジェクトの配列。

型: [PredictorSummary](#) オブジェクトの配列

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード : 400

InvalidNextTokenException

トークンが無効です。トークンの有効期間は 24 時間です。

HTTP ステータスコード : 400

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListTagsForResource

サービス: Amazon Forecast Service

Amazon Forecast のリソースのタグを一覧表示します。

リクエストの構文

```
{  
  "ResourceArn": "string"  
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### ResourceArn

タグをリストするリソースを識別する Amazon リソース名前 (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

レスポンスの構文

```
{  
  "Tags": [  
    {  
      "Key": "string",  
      "Value": "string"  
    }  
  ]  
}
```

レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

## Tags

リソースのタグ。

タイプ : [Tag](#) オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 200 項目です。

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード : 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)





## ListWhatIfAnalyses

サービス: Amazon Forecast Service

[CreateWhatIfAnalysis](#) オペレーションを使用して作成された What-If 分析のリストを返します。このオペレーションは、各 What-If 分析について、Amazon リソースネーム (ARN) を含むプロパティの概要を返します。[DescribeWhatIfAnalysis](#) オペレーションで What-If 分析 ARN を使用すると、プロパティの完全なセットを取得できます。

リクエストの構文

```
{
  "Filters": [
    {
      "Condition": "string",
      "Key": "string",
      "Value": "string"
    }
  ],
  "MaxResults": number,
  "NextToken": "string"
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### Filters

フィルターの配列。各フィルターについて、条件および match ステートメントを指定します。条件は IS または IS\_NOT のいずれかであり、それぞれについて、ステートメントに一致する What-If 分析ジョブをリストに含めるか除外するかを指定します。match ステートメントは、キーと値で構成されます。

フィルタープロパティ

- Condition - 適用する条件。有効な値は、IS および IS\_NOT です。ステートメントに一致する What-If 分析を含めるには、IS を指定します。一致する What-If 分析ジョブを除外するには、IS\_NOT を指定します。
- Key - フィルタリングするパラメータの名前。有効な値は、WhatIfAnalysisArn および Status です。
- Value - 照合する値。

例えば、という名前の予測をエクスポートするすべてのジョブを一覧表示するにはelectricityWhatIf、次のフィルターを指定します。

```
"Filters": [ { "Condition": "IS", "Key": "WhatIfAnalysisArn", "Value": "arn:aws:forecast:us-west-2:<acct-id>:forecast/electricityWhatIf" } ]
```

型: [Filter](#) オブジェクトの配列

必須: いいえ

### [MaxResults](#)

レスポンスで返される項目の数。

タイプ: 整数

有効範囲: 最小値は 1 です。最大値は 100 です。

必須: いいえ

### [NextToken](#)

前のリクエストの結果が切り捨てられた場合、レスポンスには NextToken が含まれます。次の結果セットを取得するには、次のリクエストでトークンを使用します。トークンの有効期間は 24 時間です。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

Pattern: .+

必須: いいえ

### レスポンスの構文

```
{
  "NextToken": "string",
  "WhatIfAnalyses": [
    {
      "CreationTime": number,
      "ForecastArn": "string",
      "LastModificationTime": number,
```

```
    "Message": "string",
    "Status": "string",
    "WhatIfAnalysisArn": "string",
    "WhatIfAnalysisName": "string"
  }
]
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### [NextToken](#)

レスポンスが切り捨てられた場合、Forecast はこのトークンを返します。次の結果セットを取得するには、次のリクエストでトークンを使用します。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

Pattern: .+

### [WhatifAnalyses](#)

一致した分析を記述する `WhatIfAnalysisSummary` オブジェクトの配列。

型: [WhatIfAnalysisSummary](#) オブジェクトの配列

## エラー

### `InvalidInputException`

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

### `InvalidNextTokenException`

トークンが無効です。トークンの有効期間は 24 時間です。

HTTP ステータスコード: 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListWhatIfForecastExports

サービス: Amazon Forecast Service

[CreateWhatIfForecastExport](#) オペレーションを使用して作成された What-If 予想エクスポートのリストを返します。このオペレーションは、各 What-If 予想エクスポートについて、Amazon リソースネーム (ARN) を含むプロパティの概要を返します。[DescribeWhatIfForecastExport](#) オペレーションで What-If 予想エクスポート ARN を使用すると、プロパティの完全なセットを取得できます。

### リクエストの構文

```
{
  "Filters": [
    {
      "Condition": "string",
      "Key": "string",
      "Value": "string"
    }
  ],
  "MaxResults": number,
  "NextToken": "string"
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### Filters

フィルターの配列。各フィルターについて、条件および match ステートメントを指定します。条件は IS または IS\_NOT のいずれかであり、それぞれについて、ステートメントに一致する What-If 予想エクスポートジョブをリストに含めるか除外するかを指定します。match ステートメントは、キーと値で構成されます。

#### フィルタープロパティ

- Condition - 適用する条件。有効な値は、IS および IS\_NOT です。ステートメントに一致する予測エクスポートジョブを含めるには、IS を指定します。一致する予測エクスポートジョブを除外するには、IS\_NOT を指定します。
- Key - フィルタリングするパラメータの名前。有効な値は、WhatIfForecastExportArn および Status です。
- Value - 照合する値。

例えば、electricityWIFExport という名前の予想をエクスポートするすべてのジョブを一覧表示するには、次のフィルターを指定します。

```
"Filters": [ { "Condition": "IS", "Key": "WhatIfForecastExportArn",
"Value": "arn:aws:forecast:us-west-2:<acct-id>:forecast/
electricityWIFExport" } ]
```

型: [Filter](#) オブジェクトの配列

必須: いいえ

### [MaxResults](#)

レスポンスで返される項目の数。

タイプ: 整数

有効範囲: 最小値は 1 です。最大値は 100 です。

必須: いいえ

### [NextToken](#)

前のリクエストの結果が切り捨てられた場合、レスポンスには NextToken が含まれます。次の結果セットを取得するには、次のリクエストでトークンを使用します。トークンの有効期間は 24 時間です。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

Pattern: .+

必須: いいえ

### レスポンスの構文

```
{
  "NextToken": "string",
  "WhatIfForecastExports": [
    {
      "CreationTime": number,
      "Destination": {
        "S3Config": {
```

```
        "KMSKeyArn": "string",
        "Path": "string",
        "RoleArn": "string"
    }
},
"LastModificationTime": number,
"Message": "string",
"Status": "string",
"WhatIfForecastArns": [ "string" ],
"WhatIfForecastExportArn": "string",
"WhatIfForecastExportName": "string"
}
]
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### [NextToken](#)

レスポンスが切り捨てられた場合、Forecast はこのトークンを返します。次の結果セットを取得するには、次のリクエストでトークンを使用します。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

Pattern: .+

### [WhatIfForecastExports](#)

一致した予想エクスポートを記述する WhatIfForecastExports オブジェクトの配列。

型: [WhatIfForecastExportSummary](#) オブジェクトの配列

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。



HTTP ステータスコード : 400

InvalidNextTokenException

トークンが無効です。トークンの有効期間は 24 時間です。

HTTP ステータスコード : 400

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListWhatIfForecasts

サービス: Amazon Forecast Service

[CreateWhatIfForecast](#) オペレーションを使用して作成された What-If 予想のリストを返します。このオペレーションは、各 What-If 予想について、Amazon リソースネーム (ARN) を含むプロパティの概要を返します。[DescribeWhatIfForecast](#) オペレーションで What-If 予想 ARN を使用すると、プロパティの完全なセットを取得できます。

リクエストの構文

```
{
  "Filters": [
    {
      "Condition": "string",
      "Key": "string",
      "Value": "string"
    }
  ],
  "MaxResults": number,
  "NextToken": "string"
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### Filters

フィルターの配列。各フィルターについて、条件および match ステートメントを指定します。条件は IS または IS\_NOT のいずれかであり、それぞれについて、ステートメントに一致する What-If 予想エクスポートジョブをリストに含めるか除外するかを指定します。match ステートメントは、キーと値で構成されます。

フィルタープロパティ

- Condition - 適用する条件。有効な値は、IS および IS\_NOT です。ステートメントに一致する予測エクスポートジョブを含めるには、IS を指定します。一致する予測エクスポートジョブを除外するには、IS\_NOT を指定します。
- Key - フィルタリングするパラメータの名前。有効な値は、WhatIfForecastArn および Status です。
- Value - 照合する値。

たとえば、electricityWhatIfForecast という名前の予測をエクスポートするすべてのジョブを一覧表示するには、次のフィルターを指定します。

```
"Filters": [ { "Condition": "IS", "Key": "WhatIfForecastArn",  
"Value": "arn:aws:forecast:us-west-2:<acct-id>:forecast/  
electricityWhatIfForecast" } ]
```

型: [Filter](#) オブジェクトの配列

必須: いいえ

### [MaxResults](#)

レスポンスで返される項目の数。

タイプ: 整数

有効範囲: 最小値は 1 です。最大値は 100 です。

必須: いいえ

### [NextToken](#)

前のリクエストの結果が切り捨てられた場合、レスポンスには NextToken が含まれます。次の結果セットを取得するには、次のリクエストでトークンを使用します。トークンの有効期間は 24 時間です。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

Pattern: .+

必須: いいえ

## レスポンスの構文

```
{  
  "NextToken": "string",  
  "WhatIfForecasts": [  
    {  
      "CreationTime": number,  
      "LastModificationTime": number,
```

```
    "Message": "string",
    "Status": "string",
    "WhatIfAnalysisArn": "string",
    "WhatIfForecastArn": "string",
    "WhatIfForecastName": "string"
  }
]
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### [NextToken](#)

前のリクエストの結果が切り捨てられた場合、レスポンスには NextToken が含まれます。次の結果セットを取得するには、次のリクエストでトークンを使用します。トークンの有効期間は 24 時間です。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

Pattern: .+

### [WhatIfForecasts](#)

一致した予想を記述する WhatIfForecasts オブジェクトの配列。

型: [WhatIfForecastSummary](#) オブジェクトの配列

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

### InvalidNextTokenException

トークンが無効です。トークンの有効期間は 24 時間です。

HTTP ステータスコード : 400

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ResumeResource

サービス: Amazon Forecast Service

停止したモニターリソースを再開します。

リクエストの構文

```
{  
  "ResourceArn": "string"  
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### ResourceArn

再開するモニターリソースの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

エラー

`InvalidInputException`

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

`LimitExceededException`

アカウントあたりのリソース数の制限を超えました。

HTTP ステータスコード : 400

#### ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

#### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## StopResource

サービス: Amazon Forecast Service

リソースを停止します。

リソースは次の状態になります: CREATE\_STOPPING および CREATE\_STOPPED。リソースが停止されると、そのリソースを再開することはできません。

このオペレーションは、次のリソース (および対応する子リソース) に適用できます。

- データセットのインポートジョブ
- 予測子ジョブ
- 予測ジョブ
- 予測エクスポートジョブ
- 予測子バックテストエクスポートジョブ
- Explainability ジョブ
- Explainability エクスポートジョブ

### リクエストの構文

```
{  
  "ResourceArn": "string"  
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### ResourceArn

停止するリソースを識別する Amazon リソースネーム (ARN)。サポートされている ARN は、DatasetImportJobArn、PredictorArn、PredictorBacktestExportJobArn、ForecastArn および ExplainabilityExportArn です。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`



必須：はい

## レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード：400

### LimitExceededException

アカウントあたりのリソース数の制限を超えました。

HTTP ステータスコード：400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード：400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)

- [AWS SDK for Ruby V3](#)

## TagResource

サービス: Amazon Forecast Service

指定されたタグを、指定された `resourceArn` を持つリソースに関連付けます。リソース上の既存のタグがリクエストパラメータで指定されていない場合、それらは変更されません。リソースが削除されると、そのリソースに関連付けられているタグも削除されます。

### リクエストの構文

```
{
  "ResourceArn": "string",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### ResourceArn

タグをリストするリソースを識別する Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]):forecast:.*:.*:.*`

必須: はい

#### Tags

リソースに追加するタグ。タグは、キーバリューペアの配列です。

タグには以下のベーシックな制限があります。

- リソースあたりのタグの最大数は 50 です。

- タグキーは、リソースごとにそれぞれ一意である必要があります。また、各タグキーに設定できる値は 1 つのみです。
- キーの最大長 - 128 文字 (Unicode) (UTF-8)。
- 値の最大長 - 256 文字 (Unicode) (UTF-8)。
- 複数の のサービス間およびリソース間でタグ付けスキーマを使用する場合、他のサービスにも許容される文字数に制限がある可能性があることに注意してください。一般的に使用が許可される文字は、UTF-8 で表現可能な文字、数字、スペース、および以下の文字です：  
+、-、=、.、\_、:、/、@。
- タグのキーと値では、大文字と小文字が区別されます。
- aws:、AWS:、またはキーのプレフィックスなどの大文字または小文字の組み合わせは使用しないでください。AWS 使用のために予約されています。このプレフィックスが含まれるタグキーを編集したり削除することはできません。値にはこのプレフィックスを付けることができます。タグ値には aws がプレフィックスとして付されているが、キーには付されていない場合、Forecast は、それをユーザータグとみなし、タグ数の上限である 50 個を計算する際に算入します。aws のキープレフィックスのみを持つタグは、リソース制限あたりのタグに算入されません。

タイプ: [Tag](#) オブジェクトの配列

配列メンバー: 最小数は 0 項目です。最大数は 200 項目です。

必須: はい

## レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード: 400

### LimitExceededException

アカウントあたりのリソース数の制限を超えました。

HTTP ステータスコード: 400

## ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## UntagResource

サービス: Amazon Forecast Service

指定されたタグをリソースから削除します。

リクエストの構文

```
{
  "ResourceArn": "string",
  "TagKeys": [ "string" ]
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### ResourceArn

タグをリストするリソースを識別する Amazon リソース名前 (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

### TagKeys

削除するタグのキー。

タイプ: 文字列の配列

配列メンバー: 最小数は 0 項目です。最大数は 200 項目です。

長さの制限: 最小長は 1 です。最大長は 128 です。

パターン: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

必須: はい

レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード : 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## UpdateDatasetGroup

サービス: Amazon Forecast Service

データセットグループ内のデータセットを指定されたデータセットに置き換えます。

### Note

データセットグループを使用して予測子を作成する前に、データセットグループの Status が ACTIVE となっている必要があります。[DescribeDatasetGroup](#) オペレーションを使用してステータスを取得します。

### リクエストの構文

```
{  
  "DatasetArns": [ "string" ],  
  "DatasetGroupArn": "string"  
}
```

### リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

#### DatasetArns

データセットグループに追加するデータセットの Amazon リソースネーム (ARN) の配列。

タイプ: 文字列の配列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

#### DatasetGroupArn

データセットグループの ARN。

型: 文字列

長さの制限: 最大長は 256 です。



パターン : `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須 : はい

## レスポンス要素

アクションが成功した場合、サービスは空の HTTP 本文を持つ HTTP 200 レスポンスを返します。

## エラー

### InvalidInputException

無効な値または有効な範囲を超える値が含まれているため、リクエストを処理できません。

HTTP ステータスコード : 400

### ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

### ResourceNotFoundException

その Amazon リソースネーム (ARN) を持つリソースが見つかりません。ARN を確認して、もう一度お試しください。

HTTP ステータスコード : 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)

- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## Amazon Forecast クエリサービス

次のアクションは、Amazon Forecast Query Service でサポートされています:

- [QueryForecast](#)
- [QueryWhatIfForecast](#)

## QueryForecast

サービス: Amazon Forecast Query Service

指定された基準でフィルタリングされた、単一の項目の予測を取得します。

基準は、キーバリューペアです。キーは、TARGET\_TIME\_SERIES データセットからの `item_id` (もしくは同等の非タイムスタンプ、非ターゲットフィールド)、または `FeaturizationConfig` オブジェクトの一部として指定された予測ディメンションの 1 つです。

デフォルトでは、QueryForecast は、フィルタリングされた予測の完全な日付範囲を返します。特定の日付範囲をリクエストできます。

完全な予測を取得するには、[CreateForecastExportJob](#) オペレーションを使用します。

### Note

Amazon Forecast によって生成された予測は、予測子の作成に使用されたデータセットと同じタイムゾーンにあります。

## リクエストの構文

```
{
  "EndDate": "string",
  "Filters": {
    "string" : "string"
  },
  "ForecastArn": "string",
  "NextToken": "string",
  "StartDate": "string"
}
```

## リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### EndDate

予測の終了日。次の形式を使用して日付を指定します: `yyyy-MM-dd'T'HH:mm:ss` (ISO 8601 の形式)。例: `2015-01-01T20:00:00`。

タイプ: 文字列

必須: いいえ

## Filters

予測の取得時に適用するフィルター条件。例えば、電力使用量データセットの client\_21 についての予測を取得するには、次のように指定します。

```
{"item_id" : "client_21"}
```

完全な予測を取得するには、[CreateForecastExportJob](#) オペレーションを使用します。

型: 文字列間のマッピング

マップエントリ: 項目の最大数は 50 です。

キーの長さの制限: 最大長は 256 です。

キーパターン: `^[a-zA-Z0-9\_\-]+`

値の長さの制限: 最大長は 256 です。

必須: はい

## ForecastArn

クエリする予測の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

## NextToken

前のリクエストの結果が切り捨てられた場合、レスポンスには NextToken が含まれます。次の結果セットを取得するには、次のリクエストでトークンを使用します。トークンの有効期間は 24 時間です。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

必須: いいえ

## StartDate

予測の開始日。次の形式を使用して日付を指定します: yyyy-MM-dd'T'HH:mm:ss (ISO 8601 の形式)。例: 2015-01-01T08:00:00。

タイプ: 文字列

必須: いいえ

## レスポンスの構文

```
{
  "Forecast": {
    "Predictions": {
      "string": [
        {
          "Timestamp": "string",
          "Value": number
        }
      ]
    }
  }
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

## Forecast

予測。

型: Forecast オブジェクト

## エラー

### InvalidInputException

値が無効であるか、長すぎます。

HTTP ステータスコード : 400

InvalidNextTokenException

トークンが無効です。トークンの有効期間は 24 時間です。

HTTP ステータスコード : 400

LimitExceededException

1 秒あたりのリクエスト数の上限を超えています。

HTTP ステータスコード : 400

ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

ResourceNotFoundException

そのリソースを見つけることができません。入力した情報を確認して、再試行してください。

HTTP ステータスコード : 400

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## QueryWhatIfForecast

サービス: Amazon Forecast Query Service

What-If 予想を取得します。

リクエストの構文

```
{
  "EndDate": "string",
  "Filters": {
    "string" : "string"
  },
  "NextToken": "string",
  "StartDate": "string",
  "WhatIfForecastArn": "string"
}
```

リクエストパラメータ

リクエストは以下の JSON 形式のデータを受け入れます。

### EndDate

What-If 予想の終了日。次の形式を使用して日付を指定します: yyyy-MM-dd'T'HH:mm:ss (ISO 8601 の形式)。例: 2015-01-01T20:00:00。

タイプ: 文字列

必須: いいえ

### Filters

予測の取得時に適用するフィルター条件。例えば、電力使用量データセットの client\_21 についての予測を取得するには、次のように指定します。

```
{"item_id" : "client_21"}
```

What-If 予想をすべて取得するには、[CreateForecastExportJob](#) オペレーションを使用します。

型: 文字列間のマッピング

マップエントリ: 項目の最大数は 50 です。

キーの長さの制限: 最大長は 256 です。

キーパターン: `^[a-zA-Z0-9\_\-]+`

値の長さの制限: 最大長は 256 です。

必須: はい

### NextToken

前のリクエストの結果が切り捨てられた場合、レスポンスには NextToken が含まれます。次の結果セットを取得するには、次のリクエストでトークンを使用します。トークンの有効期間は 24 時間です。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 3000 です。

必須: いいえ

### StartDate

What-If 予想の開始日。次の形式を使用して日付を指定します: yyyy-MM-dd'T'HH:mm:ss (ISO 8601 の形式)。例: 2015-01-01T08:00:00。

タイプ: 文字列

必須: いいえ

### WhatIfForecastArn

クエリする What-If 予想の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 300 です。

Pattern: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

必須: はい

### レスポンスの構文

```
{
  "Forecast": {
    "Predictions": {
```



```
    "string" : [
      {
        "Timestamp": "string",
        "Value": number
      }
    ]
  }
}
```

## レスポンス要素

アクションが成功すると、サービスは HTTP 200 レスポンスを返します。

サービスから以下のデータが JSON 形式で返されます。

### Forecast

予測に関する情報を提供します。[QueryForecast](#) レスポンスの一部として返されます。

型: [Forecast](#) オブジェクト

## エラー

### InvalidInputException

値が無効であるか、長すぎます。

HTTP ステータスコード : 400

### InvalidNextTokenException

トークンが無効です。トークンの有効期間は 24 時間です。

HTTP ステータスコード : 400

### LimitExceededException

1 秒あたりのリクエスト数の上限を超えています。

HTTP ステータスコード : 400

### ResourceInUseException

指定されたリソースは使用中です。

HTTP ステータスコード : 400

ResourceNotFoundException

そのリソースを見つけることができません。入力した情報を確認して、再試行してください。

HTTP ステータスコード : 400

その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS コマンドラインインターフェイス](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## データ型

次のデータ型は、Amazon Forecast サービスでサポートされています:

- [Action](#)
- [AdditionalDataset](#)
- [AttributeConfig](#)
- [Baseline](#)
- [BaselineMetric](#)
- [CategoricalParameterRange](#)
- [ContinuousParameterRange](#)
- [DataConfig](#)

- [DataDestination](#)
- [DatasetGroupSummary](#)
- [DatasetImportJobSummary](#)
- [DatasetSummary](#)
- [DataSource](#)
- [EncryptionConfig](#)
- [ErrorMetric](#)
- [EvaluationParameters](#)
- [EvaluationResult](#)
- [ExplainabilityConfig](#)
- [ExplainabilityExportSummary](#)
- [ExplainabilityInfo](#)
- [ExplainabilitySummary](#)
- [Featurization](#)
- [FeaturizationConfig](#)
- [FeaturizationMethod](#)
- [Filter](#)
- [ForecastExportJobSummary](#)
- [ForecastSummary](#)
- [HyperParameterTuningJobConfig](#)
- [InputDataConfig](#)
- [IntegerParameterRange](#)
- [MetricResult](#)
- [Metrics](#)
- [MonitorConfig](#)
- [MonitorDataSource](#)
- [MonitorInfo](#)
- [MonitorSummary](#)
- [ParameterRanges](#)
- [PredictorBacktestExportJobSummary](#)

- [PredictorBaseline](#)
- [PredictorEvent](#)
- [PredictorExecution](#)
- [PredictorExecutionDetails](#)
- [PredictorMonitorEvaluation](#)
- [PredictorSummary](#)
- [ReferencePredictorSummary](#)
- [S3Config](#)
- [Schema](#)
- [SchemaAttribute](#)
- [Statistics](#)
- [SupplementaryFeature](#)
- [Tag](#)
- [TestWindowSummary](#)
- [TimeAlignmentBoundary](#)
- [TimeSeriesCondition](#)
- [TimeSeriesIdentifiers](#)
- [TimeSeriesReplacementsDataSource](#)
- [TimeSeriesSelector](#)
- [TimeSeriesTransformation](#)
- [WeightedQuantileLoss](#)
- [WhatIfAnalysisSummary](#)
- [WhatIfForecastExportSummary](#)
- [WhatIfForecastSummary](#)
- [WindowSummary](#)

次のデータ型は、Amazon Forecast Query Service でサポートされています:

- [DataPoint](#)
- [Forecast](#)

# Amazon Forecast サービス

次のデータ型は、Amazon Forecast サービスでサポートされています:

- [Action](#)
- [AdditionalDataset](#)
- [AttributeConfig](#)
- [Baseline](#)
- [BaselineMetric](#)
- [CategoricalParameterRange](#)
- [ContinuousParameterRange](#)
- [DataConfig](#)
- [DataDestination](#)
- [DatasetGroupSummary](#)
- [DatasetImportJobSummary](#)
- [DatasetSummary](#)
- [DataSource](#)
- [EncryptionConfig](#)
- [ErrorMetric](#)
- [EvaluationParameters](#)
- [EvaluationResult](#)
- [ExplainabilityConfig](#)
- [ExplainabilityExportSummary](#)
- [ExplainabilityInfo](#)
- [ExplainabilitySummary](#)
- [Featurization](#)
- [FeaturizationConfig](#)
- [FeaturizationMethod](#)
- [Filter](#)
- [ForecastExportJobSummary](#)
- [ForecastSummary](#)

- [HyperParameterTuningJobConfig](#)
- [InputDataConfig](#)
- [IntegerParameterRange](#)
- [MetricResult](#)
- [Metrics](#)
- [MonitorConfig](#)
- [MonitorDataSource](#)
- [MonitorInfo](#)
- [MonitorSummary](#)
- [ParameterRanges](#)
- [PredictorBacktestExportJobSummary](#)
- [PredictorBaseline](#)
- [PredictorEvent](#)
- [PredictorExecution](#)
- [PredictorExecutionDetails](#)
- [PredictorMonitorEvaluation](#)
- [PredictorSummary](#)
- [ReferencePredictorSummary](#)
- [S3Config](#)
- [Schema](#)
- [SchemaAttribute](#)
- [Statistics](#)
- [SupplementaryFeature](#)
- [Tag](#)
- [TestWindowSummary](#)
- [TimeAlignmentBoundary](#)
- [TimeSeriesCondition](#)
- [TimeSeriesIdentifiers](#)
- [TimeSeriesReplacementsDataSource](#)
- [TimeSeriesSelector](#)

- [TimeSeriesTransformation](#)
- [WeightedQuantileLoss](#)
- [WhatIfAnalysisSummary](#)
- [WhatIfForecastExportSummary](#)
- [WhatIfForecastSummary](#)
- [WindowSummary](#)

## Action

サービス: Amazon Forecast Service

What-If 予想の属性に対して行う変更を定義します。例えば、このオペレーションを使用して、すべての靴の 10% オフセールを調査する What-If 予想を作成できます。そのためには、"AttributeName": "shoes"、"Operation": "MULTIPLY"、"Value": "0.90" を指定します。このオペレーションを [CreateWhatIfForecast : TimeSeriesTransformations TimeSeriesCondition](#) オペレーション内の オペレーションと組み合わせて、変更される属性項目のサブセットを定義します。

### 内容

#### AttributeName

変更する関連時系列。この値は大文字と小文字が区別されません。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: はい

#### Operation

指定した属性に適用されるオペレーション。オペレーションは次のとおりです。

- ADD - AttributeName のすべての行に Value を加算します。
- SUBTRACT - AttributeName のすべての行から Value を減算します。
- MULTIPLY - AttributeName のすべての行に Value を乗算します。
- DIVIDE - AttributeName のすべての行を Value で除算します。

型: 文字列

有効な値: ADD | SUBTRACT | MULTIPLY | DIVIDE

必須: はい

#### Value

選択した Operation に適用される値。



型: 倍精度

必須: はい

以下の資料も参照してください。

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## AdditionalDataset

サービス: Amazon Forecast Service

追加のデータセットを記述します。このオブジェクトは [DataConfig](#) オブジェクトの一部です。Forecast は、Weather Index と Holidays の追加データセットをサポートしています。

### Weather Index

Amazon Forecast の Weather Index はビルトインデータセットであり、履歴気象情報および予測される気象情報をモデルに組み込みます。Weather Index は、過去 2 年にわたる気象データと最大 14 日間の気象予報データでデータセットを補完します。詳細については、「[Amazon Forecast Weather Index](#)」(Amazon Forecast の Weather Index) を参照してください。

### Holidays

Holidays は、特徴エンジニアリングされた祝祭日情報データセットをモデルに組み込む、ビルトインの特徴量化です。250 か国を超える祝祭日カレンダーをネイティブにサポートします。Amazon Forecast には、ホリデーカレンダーを生成するための [Holiday API ライブラリ](#) と [Jollyday API](#) の両方が組み込まれています。詳細については、「[Holidays Featurization](#)」(Holidays の特徴量化) を参照してください。

### 内容

#### Name

追加データセットの名前。有効な名前: "holiday" および "weather"。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: はい

#### Configuration

##### Weather Index

Weather Index を有効にするには、Configuration の値を指定しないでください。

##### Holidays

## Holidays

Holidays を有効にするには、CountryCode を次の 2 文字の国コードのいずれかに設定します。

- アフガニスタン - AF
- オーランド諸島 - AX
- アルバニア - AL
- アルジェリア - DZ
- アメリカ領サモア - AS
- アンドラ - AD
- アンゴラ - AO
- アンギラ - AI
- 南極 - AQ
- アンティグアバーブーダ - AG
- アルゼンチン - AR
- アルメニア - AM
- アルバ - AW
- オーストラリア - AU
- オーストリア - AT
- アゼルバイジャン - AZ
- バハマ - BS
- バーレーン - BH
- バングラデシュ - BD
- バルバドス - BB
- ベラルーシ - BY
- ベルギー - BE
- ベリーズ - BZ
- ベナン - BJ
- バーミューダ - BM
- ブータン - BT
- ボリビア - BO
- ボスニアヘルツェゴビナ - BA

- ボツワナ - BW
- ブーベ島 - BV
- ブラジル - BR
- 英領インド洋地域 - IO
- 英領バージン諸島 - VG
- ブルネイ・ダルサラーム - BN
- ブルガリア - BG
- ブルキナファソ - BF
- ブルンジ - BI
- カンボジア - KH
- カメルーン - CM
- カナダ - CA
- カーボベルデ - CV
- オランダカリブ領域 - BQ
- ケイマン諸島 - KY
- 中央アフリカ共和国 - CF
- チャド - TD
- チリ - CL
- 中国 - CN
- クリスマス島 - CX
- ココス (キーリング) - CC
- コロンビア - CO
- コモロ - KM
- クック諸島 - CK
- コスタリカ - CR
- クロアチア - HR
- キューバ - CU
- キュラソー - CW
- キプロス - CY
- チェコ - CZ

- コンゴ民主共和国 - CD
- デンマーク - DK
- ジブチ - DJ
- ドミニカ国 - DM
- ドミニカ共和国 - DO
- エクアドル - EC
- エジプト - EG
- エルサルバドル - SV
- 赤道ギニア - GQ
- エリトリア - ER
- エストニア - EE
- エスワティニ - SZ
- エチオピア - ET
- フォークランド諸島 - FK
- フェロー諸島 - FO
- フィジー - FJ
- フィンランド - FI
- フランス - FR
- フランス領ギアナ - GF
- フランス領ポリネシア - PF
- フランス領南方・南極地域 - TF
- ガボン - GA
- ガンビア - GM
- ジョージア - GE
- ドイツ - DE
- ガーナ - GH
- ジブラルタル - GI
- ギリシャ - GR
- グリーンランド - GL
- グレナダ - GD

- グアドループ - GP
- グアム - GU
- グアテマラ - GT
- ガーンジー - GG
- ギニア - GN
- ギニアビサウ - GW
- ガイアナ - GY
- ハイチ - HT
- ハード島と McDonald 諸島 - HM
- ホンジュラス - HN
- 香港 - HK
- ハンガリー - HU
- アイスランド - IS
- インド - IN
- インドネシア - ID
- イラン - IR
- イラク - IQ
- アイルランド - IE
- マン島 - IM
- イスラエル - IL
- イタリア - IT
- コートジボワール - CI
- ジャマイカ - JM
- 日本 - JP
- ジャージー - JE
- ヨルダン - JO
- カザフスタン - KZ
- ケニア - KE
- キリバス - KI
- コソボ - XK

- クウェート - KW
- キルギスタン - KG
- ラオス - LA
- ラトビア - LV
- レバノン - LB
- レソト - LS
- リベリア - LR
- リビア - LY
- リヒテンシュタイン - LI
- リトアニア - LT
- ルクセンブルグ - LU
- マカオ - MO
- マダガスカル - MG
- マラウイ - MW
- マレーシア - MY
- モルディブ - MV
- マリ - ML
- マルタ - MT
- マーシャル諸島共和国 - MH
- マルティニーク - MQ
- モーリタニア - MR
- モーリシャス - MU
- マヨット - YT
- メキシコ - MX
- ミクロネシア - FM
- モルドバ - MD
- モナコ - MC
- モンゴル国 - MN
- モンテネグロ - ME
- モントセラト島 - MS

- モロッコ - MA
- モザンビーク - MZ
- ミャンマー - MM
- ナミビア - NA
- ナウル - NR
- ネパール - NP
- オランダ - NL
- ニューカレドニア - NC
- ニュージーランド - NZ
- ニカラグア - NI
- ニジェール - NE
- ナイジェリア - NG
- ニウエ - NU
- ノーフォーク島 - NF
- 北朝鮮 - KP
- 北マケドニア - MK
- 北マリアナ諸島 - MP
- ノルウェー - NO
- オマーン - OM
- パキスタン - PK
- パラオ - PW
- パレスチナ - PS
- パナマ - PA
- パプアニューギニア - PG
- パラグアイ - PY
- ペルー - PE
- フィリピン - PH
- ピトケアン諸島 - PN
- ポーランド - PL
- ポルトガル - PT



- プエルトリコ - PR
- カタール - QA
- コンゴ共和国 - CG
- レユニオン - RE
- ルーマニア - RO
- ロシア連邦 - RU
- ルワンダ - RW
- サン・バルテルミー - BL
- セントヘレナ・アセンションおよびトリスタンダクーニャ - SH
- セントクリストファーネイビス - KN
- セントルシア - LC
- サン・マルタン島 - MF
- サンピエール・ミクロン - PM
- セントビンセントおよびグレナディーン諸島 - VC
- サモア - WS
- サンマリノ - SM
- サントメプリンシペ - ST
- サウジアラビア - SA
- セネガル - SN
- セルビア - RS
- セイシェル - SC
- シエラレオネ - SL
- シンガポール - SG
- シントマーテルテン - SX
- スロバキア - SK
- スロベニア - SI
- ソロモン諸島 - SB
- ソマリア - SO
- 南アフリカ - ZA
- サウスジョージア・サウスサンドウィッチ諸島 - GS

- 韓国 - KR
- 南スーダン - SS
- スペイン - ES
- スリランカ - LK
- スーダン - SD
- スリナム - SR
- スバルバル諸島およびヤンマイエン島 - SJ
- スウェーデン - SE
- スイス - CH
- シリア・アラブ共和国 - SY
- 台湾 - TW
- タジキスタン - TJ
- タンザニア - TZ
- タイ - TH
- 東ティモール - TL
- トーゴ - TG
- トケラウ - TK
- トンガ - TO
- トリニダード・トバゴ - TT
- チュニジア - TN
- トルコ - TR
- トルクメニスタン - TM
- タークスおよびカイコス諸島 - TC
- ツバル - TV
- ウガンダ - UG
- ウクライナ - UA
- アラブ首長国連邦 - AE
- 英国 - GB
- 国連 - UN
- 米国 - US

- 合衆国領有小離島 - UM
- アメリカ領バージン諸島 - VI
- ウルグアイ - UY
- ウズベキスタン - UZ
- バヌアツ - VU
- バチカン市国 - VA
- ベネズエラ - VE
- ベトナム - VN
- ウォリス・フツナ - WF
- 西サハラ - EH
- イエメン - YE
- ザンビア - ZM
- ジンバブエ - ZW

タイプ: 文字列マップの配列への文字列

キーの長さ制限: 最小長さは 1 です。最大長は 63 です。

キーパターン: `^[a-zA-Z][a-zA-Z0-9_]*`

配列メンバー: 最小数は 1 項目です。最大数は 20 項目です。

長さの制限: 最大長は 256 です。

パターン: `^[a-zA-Z0-9_\-\-]+`

必須: いいえ

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)



## AttributeConfig

サービス: Amazon Forecast Service

属性の変換に使用されるメソッドに関する情報を提供します。

RETAIL ドメインを使用する例を次に示します。

```
{  
  
  "AttributeName": "demand",  
  
  "Transformations": {"aggregation": "sum", "middlefill": "zero", "backfill":  
    "zero"}  
  
}
```

内容

### AttributeName

スキーマで指定されている属性の名前。Amazon Forecast は、ターゲット時系列のターゲットフィールドと関連する時系列データセットをサポートしています。例えば、RETAIL ドメインについては、ターゲットは demand です。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: はい

### Transformations

上書きパラメータのマップであるメソッドパラメータ (キーバリューペア)。これらのパラメータを指定して、デフォルト値を上書きします。関連する時系列の属性は、集計パラメータを受け入れません。

次のリストは、ターゲット時系列のデータセットの「filling」特徴量化メソッドのパラメータとその有効な値を示しています。デフォルト値は太字です。

- aggregation: 合計、avg、first、min、max
- frontfill: なし

- `middlefill`: ゼロ、nan (非数値)、value、median、mean、min、max
- `backfill`: ゼロ、nan、value、median、mean、min、max

次のリストは、関連する時系列の特徴量化メソッドのパラメータとその有効な値を示しています (デフォルトはありません)。

- `middlefill`: zero, value, median, mean, min, max
- `backfill`: zero, value, median, mean, min, max
- `futurefill`: zero, value, median, mean, min, max

filling メソッドを特定の値に設定するには、`fill` パラメータを `value` に設定し、対応する `_value` パラメータで値を定義します。例えば、`backfilling` を値 2 に設定するには、`"backfill": "value" および "backfill_value": "2"` を含めます。

型: 文字列間のマッピング

マップエントリ: 項目の最大数は 20 です。

キーの長さ制限: 最小長さは 1 です。最大長は 63 です。

キーパターン: `^[a-zA-Z][a-zA-Z0-9_]*`

値の長さの制限: 最大長は 256 です。

値のパターン: `^[a-zA-Z0-9_\-\-]+$`

必須: はい

以下の資料も参照してください。

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## Baseline

サービス: Amazon Forecast Service

比較のベースラインとして使用するメトリクス。自動予測子のモニタリング結果を解釈するときは、これらのメトリクスを使用してください。

内容

### PredictorBaseline

監視している予測子の初期[精度メトリクス](#)。予測子を使用してメトリクスが変化したときに、これらのメトリクスを比較するためのベースラインとして使用してください。

タイプ: [PredictorBaseline](#) オブジェクト

必須: いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## BaselineMetric

サービス: Amazon Forecast Service

モニタリング結果を評価する際に比較に使用できる個別のメトリクス。

内容

Name

メトリクスの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: いいえ

Value

メトリクスの値。

型: 倍精度浮動小数点数

必須: いいえ

その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)



## CategoricalParameterRange

サービス: Amazon Forecast Service

カテゴリハイパーパラメータとそのチューニング可能な値の範囲を指定します。このオブジェクトは [ParameterRanges](#) オブジェクトの一部です。

### 内容

#### Name

チューニングするカテゴリハイパーパラメータの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: はい

#### Values

ハイパーパラメータのチューニング可能なカテゴリのリスト。

タイプ: 文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 20 項目です。

長さの制限: 最大長は 256 です。

パターン: `^[a-zA-Z0-9_\-\-]+`

必須: はい

以下の資料も参照してください。

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)



## ContinuousParameterRange

サービス: Amazon Forecast Service

連続ハイパーパラメータとそのチューニング可能な値の範囲を指定します。このオブジェクトは [ParameterRanges](#) オブジェクトの一部です。

内容

### MaxValue

ハイパーパラメータのチューニング可能な最大値。

型: 倍精度

必須: はい

### MinValue

ハイパーパラメータのチューニング可能な最小値。

型: 倍精度

必須: はい

### Name

チューニングするハイパーパラメータの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: はい

### ScalingType

ハイパーパラメータチューニングがハイパーパラメータの範囲を検索するために使用するスケール。有効値:

Auto

ハイパーパラメータの最適なスケールを Amazon Forecast ハイパーパラメータの調整が選択します。

## [Linear] (線形)

ハイパーパラメータの調整では、線形スケールを使用してハイパーパラメータ範囲内の値を検索します。

### 対数

ハイパーパラメータの調整では、対数を使用してハイパーパラメータ範囲内の値を検索します。

対数スケールは、範囲内の値が 0 より大きい場合に使用できます。

### ReverseLogarithmic

ハイパーパラメータの調整では、逆対数スケールを使用してハイパーパラメータ範囲内の値を検索します。

逆対数スケールは、範囲全体が範囲  $0 \leq x < 1.0$  内に収まる場合にのみ使用できます。

ハイパーパラメータのスケールの選択については、「[Hyperparameter Scaling](#)」(ハイパーパラメータのスケールリング)を参照してください。次のいずれかの値になります。

型: 文字列

有効な値: Auto | Linear | Logarithmic | ReverseLogarithmic

必須: いいえ

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## DataConfig

サービス: Amazon Forecast Service

データセットグループと追加のデータセットのデータ設定。

内容

### DatasetGroupArn

予測子のトレーニングに使用されるデータセットグループの ARN。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

### AdditionalDatasets

Holidays や Weather Index などの追加のビルトインデータセット。

型: [AdditionalDataset](#) オブジェクトの配列

配列メンバー: 最小数は 1 項目です。最大数は 2 項目です。

必須: いいえ

### AttributeConfigs

データセットグループ内の属性の集計および filling オプション。

型: [AttributeConfig](#) オブジェクトの配列

配列メンバー: 最小数は 1 項目です。最大数は 50 項目です。

必須: いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## DataDestination

サービス: Amazon Forecast Service

エクスポートジョブの宛先。S3 パス、Amazon Forecast がロケーションにアクセスできるようにする AWS Identity and Access Management (IAM) ロール、および AWS Key Management Service (KMS) キー (オプション) を指定します。

内容

### S3Config

Amazon Simple Storage Service (Amazon S3) バケットへのパスと、そのバケットにアクセスするための認証情報。

型: [S3Config](#) オブジェクト

必須: はい

以下の資料も参照してください。

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## DatasetGroupSummary

サービス: Amazon Forecast Service

[ListDatasetGroups](#) オペレーションで使用されるデータセットグループプロパティの概要を提供します。プロパティの完全なセットを取得するには、[DescribeDatasetGroup](#) オペレーションを呼び出し、を指定します DatasetGroupArn。

内容

### CreationTime

データセットグループが作成された時点。

型: タイムスタンプ

必須: いいえ

### DatasetGroupArn

データセットグループの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: いいえ

### DatasetGroupName

データセットグループの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: いいえ

### LastModificationTime

データセットグループが作成または [UpdateDatasetGroup](#) オペレーションの呼び出しから最後に更新された日時。データセットグループが更新されている間、LastModificationTime は ListDatasetGroups コールの現在の時刻です。



型: タイムスタンプ

必須: いいえ

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## DatasetImportJobSummary

サービス: Amazon Forecast Service

[ListDatasetImportJobs](#) オペレーションで使用されるデータセットのインポートジョブプロパティの概要を提供します。プロパティの完全なセットを取得するには、[DescribeDatasetImportJob](#) オペレーションを呼び出し、`DatasetImportJobArn` を指定します。

### 内容

#### CreationTime

データセットのインポートジョブが作成された時点。

型: タイムスタンプ

必須: いいえ

#### DatasetImportJobArn

データセットのインポートジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: いいえ

#### DatasetImportJobName

データセットのインポートジョブの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: いいえ

#### DataSource

インポートするトレーニングデータの場所と、Amazon Forecast がデータにアクセスするために引き受けることができる AWS Identity and Access Management (IAM) ロール。トレーニングデータは Amazon S3 バケットに保存されている必要があります。

暗号化が使用されている場合、には AWS Key Management Service (KMS) キーDataSourceが含まれます。

タイプ: [DataSource](#) オブジェクト

必須: いいえ

#### ImportMode

データセットのインポートジョブのインポートモード。完全または増分。

型: 文字列

有効な値: FULL | INCREMENTAL

必須: いいえ

#### LastModificationTime

リソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。

- CREATE\_PENDING (CreationTime)
- CREATE\_IN\_PROGRESS - 現在のタイムスタンプ。
- CREATE\_STOPPING - 現在のタイムスタンプ。
- CREATE\_STOPPED - ジョブが停止した時点。
- ACTIVE または CREATE\_FAILED - ジョブが終了または失敗した時点。

型: タイムスタンプ

必須: いいえ

#### Message

エラーが発生した場合は、エラーに関する情報メッセージ。

タイプ: 文字列

必須: いいえ

#### Status

データセットのインポートジョブのステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED

- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED

型: 文字列

長さの制限: 最大長は 256 です。

必須: いいえ

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## DatasetSummary

サービス: Amazon Forecast Service

[ListDatasets](#) オペレーションで使用されるデータセットプロパティの概要を提供します。プロパティの完全なセットを取得するには、[DescribeDataset](#) オペレーションを呼び出し、`DatasetArn` を指定します。

### 内容

#### CreationTime

データセットが作成された時点。

型: タイムスタンプ

必須: いいえ

#### DatasetArn

データセットの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

必須: いいえ

#### DatasetName

データセットの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: いいえ

#### DatasetType

データセットのタイプ。

型: 文字列

有効な値 : TARGET\_TIME\_SERIES | RELATED\_TIME\_SERIES | ITEM\_METADATA

必須 : いいえ

#### Domain

データセットに関連付けられているドメイン。

型: 文字列

有効な値 : RETAIL | CUSTOM | INVENTORY\_PLANNING | EC2\_CAPACITY | WORK\_FORCE  
| WEB\_TRAFFIC | METRICS

必須 : いいえ

#### LastModificationTime

データセットを作成する場合、LastModificationTime は CreationTime と同じです。データがデータセットにインポートされている間、LastModificationTime は ListDatasets コールの現在の時刻です。[CreateDatasetImportJob](#) オペレーションが完了すると、LastModificationTimeはインポートジョブが完了または失敗した時点になります。

型: タイムスタンプ

必須 : いいえ

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## DataSource

サービス: Amazon Forecast Service

データのソース、Amazon Forecast がデータにアクセスできるようにする AWS Identity and Access Management (IAM) ロール、およびオプションで AWS Key Management Service (KMS) キー。

内容

### S3Config

Amazon Simple Storage Service (Amazon S3) バケットに保存されたデータへのパスと、そのデータにアクセスするための認証情報。

型: [S3Config](#) オブジェクト

必須: はい

以下の資料も参照してください。

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## EncryptionConfig

サービス: Amazon Forecast Service

Amazon Forecast がキーにアクセスするために引き受けることができる AWS Key Management Service (KMS) キーと AWS Identity and Access Management (IAM) ロール。このオプションのオブジェクトは、[CreateDataset](#) および [CreatePredictor](#) リクエストで指定できます。

内容

### KMSKeyArn

KMS キーの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:aws:kms:.*:key/.*`

必須: はい

### RoleArn

Amazon Forecast が AWS KMS キーにアクセスするために引き受けることができる IAM ロールの ARN。

AWS アカウント間でロールを渡すことはできません。アカウントにないロールを渡すと、`InvalidInputException` エラーが発生します。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

以下の資料も参照してください。

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)



- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ErrorMetric

サービス: Amazon Forecast Service

予測子のパフォーマンスを評価するための詳細なエラーメトリクスを提供します。このオブジェクトは [Metrics](#) オブジェクトの一部です。

内容

### ForecastType

WAPE、MAPE、MASE、および RMSE のコンピューティングに使用される Forecast のタイプ。

型: 文字列

長さの制限: 最小長は 2 です。最大長は 4 です。

Pattern:  $(^0?\.\d\d?|\^mean\$)$

必須: いいえ

### MAPE

平均絶対パーセント誤差 (MAPE)

型: 倍精度浮動小数点数

必須: いいえ

### MASE

平均絶対スケーリング誤差 (MASE)

型: 倍精度浮動小数点数

必須: いいえ

### RMSE

root-mean-square エラー (RMSE)。

型: 倍精度浮動小数点数

必須: いいえ

### WAPE

重み付き絶対誤差率 (WAPE)。

型: 倍精度浮動小数点数

必須: いいえ

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## EvaluationParameters

サービス: Amazon Forecast Service

データセットをトレーニングデータとテストデータに分割する方法、および実行する反復回数を定義するパラメータ。これらのパラメータは事前定義されたアルゴリズムで指定されていますが、[CreatePredictor](#) リクエストで上書きできます。

内容

### BackTestWindowOffset

モデルのトレーニングとテスト (評価) のためにデータを分割するデータセットの末尾からのポイント。値をデータポイントの数として指定します。デフォルトは、予測期間の値です。BackTestWindowOffset は、過去の仮想予測開始日を再現するために使用できます。この値は、予測期間以上で、TARGET\_TIME\_SERIES データセットの長さの半分未満でなければなりません。

$\text{ForecastHorizon} \leq \text{BackTestWindowOffset} < 1/2 * \text{TARGET\_TIME\_SERIES}$  データセットの長さ

タイプ: 整数

必須: いいえ

### NumberOfBacktestWindows

入力データを分割する回数。デフォルトは 1 です。有効な値の範囲は 1~5 です。

タイプ: 整数

必須: いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## EvaluationResult

サービス: Amazon Forecast Service

アルゴリズムを評価した結果。[GetAccuracyMetrics](#) レスポンスの一部として返されます。

内容

### AlgorithmArn

評価されたアルゴリズムの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

必須: いいえ

### TestWindows

アルゴリズムの評価に使用されるテストウィンドウの配列。[EvaluationParameters](#) オブジェクトからの `NumberOfBacktestWindows` は、配列内のウィンドウの数を決定します。

型: [WindowSummary](#) オブジェクトの配列

必須: いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ExplainabilityConfig

サービス: Amazon Forecast Service

ExplainabilityConfig データ型は、に含まれる時系列と時点の数を定義します [CreateExplainability](#)。

ResourceArn の予測 ARN を指定する場合は、TimePointGranularity と TimeSeriesGranularity の両方を「ALL」に設定する必要があります。Predictor Explainability を作成するとき、Amazon Forecast は、すべての時系列と時間ポイントを考慮します。

ResourceArn の予測 ARN を指定する場合は、TimePointGranularity および TimeSeriesGranularity を「ALL」または「Specific」のいずれかに設定できます。

### 内容

#### TimePointGranularity

予測期間内のすべての時間ポイントについて Explainability を作成するには、ALL を使用します。予測期間内の特定の時間ポイントについて Explainability を作成するには、SPECIFIC を使用します。

[CreateExplainability](#) オペレーション内の StartDateTime および EndDateTime パラメータを使用して時間ポイントを指定します。

型: 文字列

有効な値 : ALL | SPECIFIC

必須: はい

#### TimeSeriesGranularity

データセット内のすべての時系列について Explainability を作成するには、ALL を使用します。データセット内の特定の時系列について Explainability を作成するには、SPECIFIC を使用します。

CSV または Parquet ファイルを Amazon S3 バケットにアップロードして時系列を指定し、[DataDestination](#) データ型内のロケーションを設定します。

型: 文字列

有効な値 : ALL | SPECIFIC

必須 : はい

以下の資料も参照してください。

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ExplainabilityExportSummary

サービス: Amazon Forecast Service

[ListExplainabilityExports](#) オペレーションで使用される Explainability エクスポートのプロパティの概要を提供します。プロパティの完全なセットを取得するには、[DescribeExplainabilityExport](#) オペレーションを呼び出して、ExplainabilityExportArn を指定します。

内容

### CreationTime

Explainability が作成された時点。

型: タイムスタンプ

必須: いいえ

### Destination

エクスポートジョブの宛先。S3 パス、Amazon Forecast がロケーションにアクセスできるようにする AWS Identity and Access Management (IAM) ロール、および AWS Key Management Service (KMS) キー (オプション) を指定します。

タイプ: [DataDestination](#) オブジェクト

必須: いいえ

### ExplainabilityExportArn

Explainability エクスポートの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: いいえ

### ExplainabilityExportName

Explainability エクスポートの名前

型: 文字列



長さの制限：最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: いいえ

## LastModificationTime

リソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。

- CREATE\_PENDING (CreationTime)
- CREATE\_IN\_PROGRESS - 現在のタイムスタンプ。
- CREATE\_STOPPING - 現在のタイムスタンプ。
- CREATE\_STOPPED - ジョブが停止した時点。
- ACTIVE または CREATE\_FAILED - ジョブが終了または失敗した時点。

型: タイムスタンプ

必須: いいえ

## Message

Explainability エクスポート中に発生した可能性のあるエラーに関する情報。

タイプ: 文字列

必須: いいえ

## Status

Explainability エクスポートのステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED

型: 文字列

長さの制限：最大長は 256 です。

必須： いいえ

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ExplainabilityInfo

サービス: Amazon Forecast Service

Explainability リソースに関する情報を提供します。

内容

### ExplainabilityArn

Explainability の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限 : 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

必須: いいえ

### Status

Explainability のステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED

型: 文字列

長さの制限 : 最大長は 256 です。

必須 : いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)



## ExplainabilitySummary

サービス: Amazon Forecast Service

[ListExplainabilities](#) オペレーションで使用される Explainability のプロパティの概要を提供します。プロパティの完全なセットを取得するには、[DescribeExplainability](#) オペレーションを呼び出して、リストされた ExplainabilityArn を指定します。

内容

### CreationTime

Explainability が作成された時点。

型: タイムスタンプ

必須: いいえ

### ExplainabilityArn

Explainability の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: いいえ

### ExplainabilityConfig

Explainability の時系列と時間ポイントの粒度を定義する構成設定。

タイプ: [ExplainabilityConfig](#) オブジェクト

必須: いいえ

### ExplainabilityName

Explainability の名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: いいえ

#### LastModificationTime

リソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。

- CREATE\_PENDING (CreationTime)
- CREATE\_IN\_PROGRESS - 現在のタイムスタンプ。
- CREATE\_STOPPING - 現在のタイムスタンプ。
- CREATE\_STOPPED - ジョブが停止した時点。
- ACTIVE または CREATE\_FAILED - ジョブが終了または失敗した時点。

型: タイムスタンプ

必須: いいえ

#### Message

Explainability の作成プロセス中に発生した可能性のあるエラーに関する情報。

タイプ: 文字列

必須: いいえ

#### ResourceArn

Explainability の作成に使用される予測子または予測の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: いいえ

#### Status

Explainability のステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED

- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED

型: 文字列

長さの制限: 最大長は 256 です。

必須: いいえ

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## Featurization

サービス: Amazon Forecast Service

### Note

このオブジェクトは [CreatePredictor](#) オペレーションに属しています。 [CreateAutoPredictor](#) で予測子を作成した場合は、「[AttributeConfig](#)」を参照してください。

データセットフィールドの特徴量化 (変換) 情報を提供します。このオブジェクトは [FeaturizationConfig](#) オブジェクトの一部です。

例:

```
{  
  "AttributeName": "demand",  
  "FeaturizationPipeline": [  
    {  
      "FeaturizationMethodName": "filling",  
      "FeaturizationMethodParameters": {"aggregation": "avg", "backfill": "nan"}  
    }  
  ]  
}
```

内容

### AttributeName

特徴量化するデータフィールドを指定するスキーマ属性の名前。Amazon Forecast は、TARGET\_TIME\_SERIES および RELATED\_TIME\_SERIES のデータセットのターゲットフィールドをサポートしています。例えば、RETAIL ドメインについては、ターゲットは demand です。CUSTOM ドメインについては、ターゲットは target\_value です。詳細については、「[欠落した値の処理](#)」を参照してください。

タイプ: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。



Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須 : はい

## FeaturizationPipeline

特徴変換メソッドを指定する 1 つの `FeaturizationMethod` オブジェクトの配列。

型: [FeaturizationMethod](#) オブジェクトの配列

配列メンバー: 定数は 1 項目です。

必須 : いいえ

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## FeaturizationConfig

サービス: Amazon Forecast Service

### Note

このオブジェクトは [CreatePredictor](#) オペレーションに属しています。[CreateAutoPredictor](#) で予測子を作成した場合は、「[AttributeConfig](#)」を参照してください。

[CreatePredictor](#) オペレーションでは、指定されたアルゴリズムが、指定されたデータセットグループを使用してモデルをトレーニングします。オプションで、モデルをトレーニングする前にデータフィールドを変更するようにオペレーションに指示できます。これらの変更は、特徴量化と呼ばれます。

FeaturizationConfig オブジェクトを使用して特徴量化を定義します。特徴量化するフィールドごとに 1 つずつ、変換の配列を指定します。その後、FeaturizationConfig オブジェクトを CreatePredictor リクエストに含めます。Amazon Forecast は、モデルトレーニングの前に TARGET\_TIME\_SERIES および RELATED\_TIME\_SERIES データセットに特徴量化を適用します。

複数の特徴量化設定を作成できます。例えば、異なる特徴量化の設定を指定して、CreatePredictor オペレーションを 2 回呼び出すことができます。

### 内容

#### ForecastFrequency

予測内の予測の頻度。

有効な間隔は、整数の後に Y (年)、M (月)、W (週)、D (日)、H (時)、min (分) をつけたものです。例えば、「1D」は毎日を示し、「15min」は 15 分ごとを示します。次に多頻度で重複するような値は指定できません。つまり、たとえば 60 分の間隔は 1 時間に相当するため、指定することはできません。各頻度の有効な値は以下のとおりです。

- 分 - 1~59
- 時間 - 1~23
- 日 - 1~6
- 週 - 1~4
- 月 - 1~11
- 年 - 1

したがって、隔週の予測が必要な場合は、「2W」と指定します。または、四半期ごとの予測が必要な場合は、「3M」と指定します。

頻度は、TARGET\_TIME\_SERIES データセットの頻度以上である必要があります。

RELATED\_TIME\_SERIES データセットが提供される場合、頻度は TARGET\_TIME\_SERIES データセットの頻度と等しくなければなりません。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 5 です。

Pattern: ^Y|M|W|D|H|30min|15min|10min|5min|1min\$

必須: はい

## Featurizations

データセットのフィールドの特徴量化 (変換) 情報の配列。

型: [Featurization](#) オブジェクトの配列

配列メンバー: 最小数は 1 項目です。最大数は 50 項目です。

必須: いいえ

## ForecastDimensions

生成された予測をグループ化する方法を指定するディメンション (フィールド) 名の配列。

例えば、すべての店舗での商品販売の予測を生成していて、データセットに store\_id フィールドが含まれているとします。店舗ごとの各商品の売上予測が必要な場合は、ディメンションとして store\_id を指定します。

TARGET\_TIME\_SERIES データセットで指定されたすべての予測ディメンションは、CreatePredictor リクエストで指定する必要はありません。RELATED\_TIME\_SERIES データセットで指定されたすべての予測ディメンションは、CreatePredictor リクエストで指定する必要があります。

タイプ: 文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 10 項目です。

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## FeaturizationMethod

サービス: Amazon Forecast Service

データセットフィールドを特徴量化 (変換) するメソッドに関する情報を提供します。メソッドは [Featurization](#) オブジェクトの `FeaturizationPipeline` の一部です。

FeaturizationMethod オブジェクトを指定する方法の例を次に示します。

```
{  
  "FeaturizationMethodName": "filling",  
  "FeaturizationMethodParameters": {"aggregation": "sum", "middlefill":  
  "zero", "backfill": "zero"}  
}
```

内容

### FeaturizationMethodName

メソッドの名前。「filling」メソッドは、サポートされている唯一のメソッドです。

型: 文字列

有効な値: filling

必須: はい

### FeaturizationMethodParameters

上書きパラメータのマップであるメソッドパラメータ (キーバリューペア)。これらのパラメータを指定して、デフォルト値を上書きします。関連する時系列の属性は、集計パラメータを受け入れません。

次のリストは、ターゲット時系列のデータセットの「filling」特徴量化メソッドのパラメータとその有効な値を示しています。太字はデフォルト値を示します。

- aggregation: 合計、avg、first、min、max
- frontfill: なし
- middlefill: **ゼロ**、nan (非数値)、value、median、mean、min、max
- backfill: **ゼロ**、nan、value、median、mean、min、max

次のリストは、関連する時系列の特徴量化メソッドのパラメータとその有効な値を示しています (デフォルトはありません)。

- `middlefill`: zero, value, median, mean, min, max
- `backfill`: zero, value, median, mean, min, max
- `futurefill`: zero, value, median, mean, min, max

filling メソッドを特定の値に設定するには、`fill` パラメータを `value` に設定し、対応する `_value` パラメータで値を定義します。例えば、`backfilling` を値 2 に設定するには、`"backfill"`: `"value"` および `"backfill_value": "2"` を含めます。

型: 文字列間のマッピング

マップエントリ: 項目の最大数は 20 です。

キーの長さの制限: 最大長は 256 です。

キーパターン: `^[a-zA-Z0-9\-\_\.\[\]\,\\"\\\s]+$`

値の長さの制限: 最大長は 256 です。

値のパターン: `^[a-zA-Z0-9\-\_\.\[\]\,\\"\\\s]+$`

必須: いいえ

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## Filter

サービス: Amazon Forecast Service

オブジェクトのサブセットを選択するためのフィルターを記述します。各フィルターは、条件と match ステートメントで構成されます。条件は IS または IS\_NOT のいずれかであり、それぞれについて、ステートメントに一致するオブジェクトを含めるか除外するかを指定します。match ステートメントは、キーと値で構成されます。

### 内容

#### Condition

適用する条件。ステートメントに一致するオブジェクトを含めるには、IS を指定します。一致するオブジェクトを除外するには、IS\_NOT を指定します。

型: 文字列

有効な値: IS | IS\_NOT

必須: はい

#### Key

フィルタリングするパラメータの名前。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `^[a-zA-Z0-9\_]+$`

必須: はい

#### Value

照合する値。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

以下の資料も参照してください。

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)



## ForecastExportJobSummary

サービス: Amazon Forecast Service

[ListForecastExportJobs](#) オペレーションで使用される予測エクスポートジョブのプロパティの概要を提供します。プロパティの完全なセットを取得するには、[DescribeForecastExportJob](#) オペレーションを呼び出して、リストされた ForecastExportJobArn を指定します。

### 内容

#### CreationTime

予測エクスポートジョブが作成された時点。

型: タイムスタンプ

必須: いいえ

#### Destination

予測がエクスポートされる Amazon Simple Storage Service (Amazon S3) バケットへのパス。

タイプ: [DataDestination](#) オブジェクト

必須: いいえ

#### ForecastExportJobArn

予測エクスポートジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: いいえ

#### ForecastExportJobName

予測エクスポートジョブの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: いいえ

### LastModificationTime

リソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。

- CREATE\_PENDING (CreationTime)
- CREATE\_IN\_PROGRESS - 現在のタイムスタンプ。
- CREATE\_STOPPING - 現在のタイムスタンプ。
- CREATE\_STOPPED - ジョブが停止した時点。
- ACTIVE または CREATE\_FAILED - ジョブが終了または失敗した時点。

型: タイムスタンプ

必須: いいえ

### Message

エラーが発生した場合は、エラーに関する情報メッセージ。

タイプ: 文字列

必須: いいえ

### Status

予測のエクスポートジョブのステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED

#### Note

S3 バケットの予測にアクセスする前に、予測エクスポートジョブの Status が ACTIVE となっている必要があります。

型: 文字列

長さの制限：最大長は 256 です。

必須：いいえ

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ForecastSummary

サービス: Amazon Forecast Service

[ListForecasts](#) オペレーションで使用される予測のプロパティの概要を提供します。プロパティの完全なセットを取得するには、[DescribeForecast](#) オペレーションを呼び出して、概要にリストされている ForecastArn を指定します。

内容

### CreatedUsingAutoPredictor

Forecast が から作成されたかどうか AutoPredictor。

型: ブール値

必須: いいえ

### CreationTime

予測作成タスクが作成された時点。

型: タイムスタンプ

必須: いいえ

### DatasetGroupArn

予測子のトレーニングに使用されるデータを提供したデータセットグループの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `^[a-zA-Z0-9\_]+$`

必須: いいえ

### ForecastArn

予測の ARN。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

必須: いいえ

### ForecastName

予測の名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: いいえ

### LastModificationTime

リソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。

- CREATE\_PENDING (CreationTime)
- CREATE\_IN\_PROGRESS - 現在のタイムスタンプ。
- CREATE\_STOPPING - 現在のタイムスタンプ。
- CREATE\_STOPPED - ジョブが停止した時点。
- ACTIVE または CREATE\_FAILED - ジョブが終了または失敗した時点。

型: タイムスタンプ

必須: いいえ

### Message

エラーが発生した場合は、エラーに関する情報メッセージ。

タイプ: 文字列

必須: いいえ

### PredictorArn

予測の生成に使用される予測子の ARN。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `^[a-zA-Z0-9\_]+$`

必須: いいえ

## Status

予測のステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED

### Note

予測をクエリまたはエクスポートする前に、予測の Status が ACTIVE となっている必要があります。

型: 文字列

長さの制限: 最大長は 256 です。

必須: いいえ

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## HyperParameterTuningJobConfig

サービス: Amazon Forecast Service

ハイパーパラメータのチューニングジョブの設定情報。このオブジェクトは [CreatePredictor](#) リクエストで指定します。

ハイパーパラメータは、モデルトレーニングのプロセスを管理するパラメータです。トレーニング中に決定されるモデルパラメータとは異なり、トレーニング開始前にハイパーパラメータを設定します。ハイパーパラメータの値は、モデルパラメータ用に選択される値に影響します。

ハイパーパラメータのチューニングジョブでは、Amazon Forecast は、指定されたメトリクスを最適化するハイパーパラメータ値のセットを選択します。Forecast は、ハイパーパラメータ値の範囲で多くのトレーニングジョブを実行することでこれを実現します。最適な値のセットは、アルゴリズム、トレーニングデータ、および指定されたメトリクスの目的によって異なります。

内容

### ParameterRanges

ハイパーパラメータの有効な値の範囲を指定します。

タイプ: [ParameterRanges](#) オブジェクト

必須: いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## InputDataConfig

サービス: Amazon Forecast Service

### Note

このオブジェクトは [CreatePredictor](#) オペレーションに属しています。 [CreateAutoPredictor](#) で予測子を作成した場合は、「[DataConfig](#)」を参照してください。

予測子のトレーニングに使用されるデータ。データには、データセットグループと補足的な特徴が含まれます。このオブジェクトは [CreatePredictor](#) リクエストで指定します。

### 内容

#### DatasetGroupArn

データセットグループの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

#### SupplementaryFeatures

補足的な特徴の配列。サポートされている特徴は祝祭日カレンダーのみです。

型: [SupplementaryFeature](#) オブジェクトの配列

配列メンバー: 最小数は 1 項目です。最大数は 2 項目です。

必須: いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)



- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## IntegerParameterRange

サービス: Amazon Forecast Service

整数のハイパーパラメータとそのチューニング可能な値の範囲を指定します。このオブジェクトは [ParameterRanges](#) オブジェクトの一部です。

内容

### MaxValue

ハイパーパラメータのチューニング可能な最大値。

タイプ: 整数

必須: はい

### MinValue

ハイパーパラメータのチューニング可能な最小値。

タイプ: 整数

必須: はい

### Name

チューニングするハイパーパラメータの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: はい

### ScalingType

ハイパーパラメータチューニングがハイパーパラメータの範囲を検索するために使用するスケール。有効値:

Auto

ハイパーパラメータの最適なスケールを Amazon Forecast ハイパーパラメータの調整が選択します。

## [Linear] (線形)

ハイパーパラメータの調整では、線形スケールを使用してハイパーパラメータ範囲内の値を検索します。

### 対数

ハイパーパラメータの調整では、対数を使用してハイパーパラメータ範囲内の値を検索します。

対数スケールは、範囲内の値が 0 より大きい場合に使用できます。

### ReverseLogarithmic

IntegerParameterRange をサポートしていません。

逆対数スケールは、範囲全体が範囲  $0 \leq x < 1.0$  内に収まる場合にのみ使用できます。

ハイパーパラメータのスケールの選択については、「[Hyperparameter Scaling](#)」(ハイパーパラメータのスケールリング)を参照してください。次のいずれかの値になります。

型: 文字列

有効な値: Auto | Linear | Logarithmic | ReverseLogarithmic

必須: いいえ

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## MetricResult

サービス: Amazon Forecast Service

予測子の使用量を監視する際に Forecast が計算した個別のメトリクス。このメトリクスの値を [Baseline](#) 内のメトリクスの値と比較して、予測子のパフォーマンスがどのように変化しているかを確認できます。

Forecast によって生成されるメトリクスの詳細については、「[Evaluating Predictor Accuracy](#)」を参照してください。

内容

### MetricName

メトリクスの名前。

型: 文字列

長さの制限: 最大長は 256 です。

必須: いいえ

### MetricValue

メトリクスの値。

型: 倍精度浮動小数点数

必須: いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## Metrics

サービス: Amazon Forecast Service

予測子のパフォーマンスを評価するために使用されるメトリクスを提供します。このオブジェクトは [WindowSummary](#) オブジェクトの一部です。

内容

### AverageWeightedQuantileLoss

すべての重み付き分位損失の平均値。

型: 倍精度浮動小数点数

必須: いいえ

### ErrorMetrics

各予測タイプの詳細なエラーメトリクスを提供します。メトリクスには、二乗平均平方根誤差 (RMSE)、平均絶対パーセント誤差 (MAPE)、平均絶対スケーリング誤差 (MASE)、および重み付き絶対誤差率 (WAPE) が含まれます。

型: [ErrorMetric](#) オブジェクトの配列

必須: いいえ

### RMSE

このメンバーは非推奨になりました。

root-mean-square エラー (RMSE)。

型: 倍精度浮動小数点数

必須: いいえ

### WeightedQuantileLosses

重み付き分位損失の配列。分位数は、確率分布を等しい確率の領域に分割します。この場合の分布は損失関数です。

型: [WeightedQuantileLoss](#) オブジェクトの配列

必須: いいえ

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## MonitorConfig

サービス: Amazon Forecast Service

予測子モニターの構成の詳細。

内容

### MonitorName

モニターリソースの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: はい

以下の資料も参照してください。

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## MonitorDataSource

サービス: Amazon Forecast Service

モニターが評価中に使用したデータのソース。

内容

### DatasetImportJobArn

モニター評価を開始したデータのインポートに使用されるデータセットのインポートジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

必須: いいえ

### ForecastArn

評価中にモニターが使用する予想の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

必須: いいえ

### PredictorArn

モニタリングする予測子リソースの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

必須: いいえ



## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## MonitorInfo

サービス: Amazon Forecast Service

モニターリソースに関する情報を提供します。

内容

### MonitorArn

モニターリソースの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

必須: いいえ

### Status

モニターのステータス。含まれる状態は次のとおりです。

- ACTIVE
- ACTIVE\_STOPPING, ACTIVE\_STOPPED
- UPDATE\_IN\_PROGRESS
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED

型: 文字列

長さの制限: 最大長は 256 です。

必須: いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)

- [AWS SDK for Ruby V3](#)

## MonitorSummary

サービス: Amazon Forecast Service

[ListMonitors](#) オペレーションで使用されるモニターのプロパティの概要を提供します。プロパティの完全なセットを取得するには、[DescribeMonitor](#) オペレーションを呼び出して、リストされた `MonitorArn` を指定します。

内容

### CreationTime

モニターリソースが作成された時点。

型: タイムスタンプ

必須: いいえ

### LastModificationTime

モニターリソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。

- `CREATE_PENDING` (`CreationTime`)
- `CREATE_IN_PROGRESS` - 現在のタイムスタンプ。
- `STOPPED` - リソースが停止したとき。
- `ACTIVE` または `CREATE_FAILED` - モニターの作成が終了または失敗したとき。

型: タイムスタンプ

必須: いいえ

### MonitorArn

モニターリソースの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: いいえ

## MonitorName

モニターリソースの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: いいえ

## ResourceArn

モニタリングされる予測子の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: いいえ

## Status

モニターのステータス。含まれる状態は次のとおりです。

- ACTIVE
- ACTIVE\_STOPPING, ACTIVE\_STOPPED
- UPDATE\_IN\_PROGRESS
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED

型: 文字列

長さの制限: 最大長は 256 です。

必須: いいえ

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ParameterRanges

サービス: Amazon Forecast Service

カテゴリ、連続、整数のハイパーパラメータと、それらのチューニング可能な値の範囲を指定します。チューニング可能な値の範囲によって、ハイパーパラメータチューニングジョブが指定されたハイパーパラメータ用に選択できる値が決まります。このオブジェクトは [HyperParameterTuningJobConfig](#) オブジェクトの一部です。

内容

### CategoricalParameterRanges

各カテゴリハイパーパラメータのチューニング可能な範囲を指定します。

型: [CategoricalParameterRange](#) オブジェクトの配列

配列メンバー: 最小数は 1 項目です。最大数は 20 項目です。

必須: いいえ

### ContinuousParameterRanges

各連続ハイパーパラメータのチューニング可能な範囲を指定します。

型: [ContinuousParameterRange](#) オブジェクトの配列

配列メンバー: 最小数は 1 項目です。最大数は 20 項目です。

必須: いいえ

### IntegerParameterRanges

各整数ハイパーパラメータのチューニング可能な範囲を指定します。

型: [IntegerParameterRange](#) オブジェクトの配列

配列メンバー: 最小数は 1 項目です。最大数は 20 項目です。

必須: いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)



## PredictorBacktestExportJobSummary

サービス: Amazon Forecast Service

[ListPredictorBacktestExportJobs](#) オペレーションで使用される予測子バックテストのエクスポートジョブのプロパティの概要を提供します。プロパティの完全なセットを取得するには、[DescribePredictorBacktestExportJob](#) オペレーションを呼び出して、リストされた `PredictorBacktestExportJobArn` を指定します。

### 内容

#### CreationTime

予測子バックテストエクスポートジョブが作成された時点。

型: タイムスタンプ

必須: いいえ

#### Destination

エクスポートジョブの宛先。S3 バス、Amazon Forecast がロケーションにアクセスできるようにする AWS Identity and Access Management (IAM) ロール、および AWS Key Management Service (KMS) キー (オプション) を指定します。

タイプ: [DataDestination](#) オブジェクト

必須: いいえ

#### LastModificationTime

リソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。

- CREATE\_PENDING (CreationTime)
- CREATE\_IN\_PROGRESS - 現在のタイムスタンプ。
- CREATE\_STOPPING - 現在のタイムスタンプ。
- CREATE\_STOPPED - ジョブが停止した時点。
- ACTIVE または CREATE\_FAILED - ジョブが終了または失敗した時点。

型: タイムスタンプ

必須: いいえ

## Message

バックテストエクスポート中に発生した可能性のあるエラーに関する情報。

タイプ: 文字列

必須: いいえ

## PredictorBacktestExportJobArn

予測子バックテストエクスポートジョブの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

必須: いいえ

## PredictorBacktestExportJobName

予測子バックテストエクスポートジョブの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: いいえ

## Status

予測子バックテストエクスポートジョブのステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED

型: 文字列

長さの制限: 最大長は 256 です。

必須：いいえ

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## PredictorBaseline

サービス: Amazon Forecast Service

比較のベースラインとして使用するメトリクス。自動予測子のモニタリング結果を解釈するときは、これらのメトリクスを使用してください。

内容

### BaselineMetrics

予測子の当初の[精度メトリクス](#)。予測子を使用してメトリクスが変化したときに、これらのメトリクスを比較するためのベースラインとして使用してください。

型: [BaselineMetric](#) オブジェクトの配列

必須: いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## PredictorEvent

サービス: Amazon Forecast Service

再トレーニングなどの予測子イベントに関する詳細を提供します。

内容

Datetime

イベントが発生したときのタイムスタンプ。

型: タイムスタンプ

必須: いいえ

Detail

イベントのタイプ。例えば Retrain です。再トレーニングイベントは、予測子が再トレーニングされた時点を示します。Datetime の前のモニターの結果はすべて、前の予測子のものです。新しいメトリクスはすべて新しく再トレーニングされた予測子のものです。

型: 文字列

長さの制限: 最大長は 256 です。

必須: いいえ

その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## PredictorExecution

サービス: Amazon Forecast Service

バックテストの実行に使用されるアルゴリズムとそれらのテストのステータス。

内容

### AlgorithmArn

予測子のテストに使用されるアルゴリズムの ARN。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: いいえ

### TestWindows

アルゴリズムの評価に使用されるテストウィンドウの配列。[EvaluationParameters](#) オブジェクトからの `NumberOfBacktestWindows` は、配列内のウィンドウの数を決定します。

型: [TestWindowSummary](#) オブジェクトの配列

必須: いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## PredictorExecutionDetails

サービス: Amazon Forecast Service

予測子の精度を評価するために実行されたバックテストの詳細が含まれています。テストは精度の高い順に返され、最も正確なバックテストが最初に表示されます。[CreatePredictor](#) オペレーションを呼び出すときに実行するバックテストの数を指定します。

内容

### PredictorExecutions

特定のアルゴリズムに対する予測子の精度を評価するために実行されるバックテストの配列。[EvaluationParameters](#) オブジェクトからの `NumberOfBacktestWindows` は、配列内のウィンドウの数を決定します。

型: [PredictorExecution](#) オブジェクトの配列

配列メンバー: 最小数は 1 項目です。最大数は 5 項目です。

必須: いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## PredictorMonitorEvaluation

サービス: Amazon Forecast Service

モニター評価の結果について説明します。

内容

### EvaluationState

モニター評価の状態。状態は、SUCCESS または FAILURE にできます。

型: 文字列

長さの制限: 最大長は 256 です。

必須: いいえ

### EvaluationTime

モニター評価が開始された日時を示すタイムスタンプ。

型: タイムスタンプ

必須: いいえ

### Message

モニター評価中に発生した可能性のあるエラーに関する情報。

タイプ: 文字列

必須: いいえ

### MetricResults

予測子を監視する際に Forecast が計算したメトリクスのリスト。リスト内の各メトリクスの値を [Baseline](#) 内のメトリクスの値と比較して、予測子のパフォーマンスがどのように変化しているかを確認できます。

型: [MetricResult](#) オブジェクトの配列

必須: いいえ

### MonitorArn

モニターリソースの Amazon リソースネーム (ARN)。



型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

必須: いいえ

#### MonitorDataSource

モニターリソースが評価中に使用したデータのソース。

タイプ: [MonitorDataSource](#) オブジェクト

必須: いいえ

#### NumItemsEvaluated

評価中に考慮された項目の数。

型: Long

必須: いいえ

#### PredictorEvent

再トレーニングなどの予測子イベントに関する詳細を提供します。

タイプ: [PredictorEvent](#) オブジェクト

必須: いいえ

#### ResourceArn

モニタリングするリソースの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

必須: いいえ

#### WindowEndDatetime

モニターの評価に使用される期間の終わりを知らせるタイムスタンプ。

型: タイムスタンプ

必須: いいえ

WindowStartDatetime

モニターの評価に使用される期間の開始を知らせるタイムスタンプ。

型: タイムスタンプ

必須: いいえ

その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## PredictorSummary

サービス: Amazon Forecast Service

[ListPredictors](#) オペレーションで使用される予測子のプロパティの概要を提供します。プロパティの完全なセットを取得するには、[DescribePredictor](#) オペレーションを呼び出して、リストされた PredictorArn を指定します。

内容

### CreationTime

モデルトレーニングタスクが作成された時点。

型: タイムスタンプ

必須: いいえ

### DatasetGroupArn

予測子のトレーニングに使用されるデータを含むデータセットグループの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: いいえ

### IsAutoPredictor

AutoPredictor 予測子の作成に が使用されたかどうか。

型: ブール値

必須: いいえ

### LastModificationTime

リソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。

- CREATE\_PENDING (CreationTime)
- CREATE\_IN\_PROGRESS - 現在のタイムスタンプ。
- CREATE\_STOPPING - 現在のタイムスタンプ。

- CREATE\_STOPPED - ジョブが停止した時点。
- ACTIVE または CREATE\_FAILED - ジョブが終了または失敗した時点。

型: タイムスタンプ

必須: いいえ

## Message

エラーが発生した場合は、エラーに関する情報メッセージ。

タイプ: 文字列

必須: いいえ

## PredictorArn

予測子の ARN。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

必須: いいえ

## PredictorName

予測子の名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: いいえ

## ReferencePredictorSummary

予測子が再トレーニングまたはアップグレードされた場合に使用される参照予測子の概要。

タイプ: [ReferencePredictorSummary](#) オブジェクト

必須: いいえ

## Status

予測子のステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED

### Note

予測子を使用して予測を作成する前に、予測子の Status が ACTIVE となっている必要があります。

型: 文字列

長さの制限: 最大長は 256 です。

必須: いいえ

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## ReferencePredictorSummary

サービス: Amazon Forecast Service

予測子を再トレーニングまたはアップグレードするときに使用される参照予測子の概要を提供します。

内容

Arn

参照予測子の ARN。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: いいえ

State

参照予測子が Active または Deleted であるかどうか。

型: 文字列

有効な値: Active | Deleted

必須: いいえ

その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## S3Config

サービス: Amazon Forecast Service

Amazon Simple Storage Service (Amazon S3) バケット内のファイルへのパス、および Amazon Forecast がファイルにアクセスするために引き受けることができる AWS Identity and Access Management (IAM) ロール (複数可)。オプションで、には AWS Key Management Service (KMS) キーが含まれます。このオブジェクトは、[CreateDatasetImportJob](#) リクエストで送信された [DataSource](#) オブジェクトの一部であり、かつ、[DataDestination](#) オブジェクトの一部です。

内容

Path

Amazon Simple Storage Service (Amazon S3) バケットまたは Amazon S3 バケット内のファイルへのパス。

型: 文字列

長さの制限: 最小長は 7 です。最大長は 4096 です。

Pattern: `^s3://[a-z0-9].+$`

必須: はい

RoleArn

Amazon Forecast が Amazon S3 バケットまたはファイルにアクセスするために引き受けることができる AWS Identity and Access Management (IAM) ロールの ARN。Amazon S3 KMSKeyArn キーに値を指定する場合、ロールはキーへのアクセスを許可する必要があります。

AWS アカウント間でロールを渡すことはできません。アカウントにないロールを渡すと、`InvalidInputException` エラーが発生します。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: はい

KMSKeyArn

(KMS) キーの Amazon リソースネーム AWS Key Management Service (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:aws:kms:.*:key/.*`

必須: いいえ

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)



## Schema

サービス: Amazon Forecast Service

データセットのフィールドを定義します。

内容

### Attributes

データセット内の各フィールドの名前とタイプを指定する属性の配列。

型: [SchemaAttribute](#) オブジェクトの配列

配列メンバー: 最小数は 1 項目です。最大数は 100 項目です。

必須: いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SchemaAttribute

サービス: Amazon Forecast Service

データセットフィールドを定義するスキーマの属性。データセットのすべてのフィールドにスキーマ属性が必要です。[Schema](#) オブジェクトには SchemaAttribute オブジェクト配列が含まれます。

内容

### AttributeName

データセットフィールドの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: いいえ

### AttributeType

フィールドのデータ型。

関連する時系列データセットでは、日付、item\_id、予測ディメンション属性以外のすべての属性が数値型 (整数/浮動小数点数) である必要があります。

型: 文字列

有効な値: `string | integer | float | timestamp | geolocation`

必須: いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## Statistics

サービス: Amazon Forecast Service

[CreateDatasetImportJob](#) オペレーションを使用して、Amazon Forecast データセットにインポートされた各データフィールドの統計を提供します。

内容

Avg

数値フィールドについては、フィールド内の平均値。

型: 倍精度浮動小数点数

必須: いいえ

Count

フィールド内の値の数。レスポンス値が -1 の場合は、「CountLong」を参照してください。

タイプ: 整数

必須: いいえ

CountDistinct

フィールド内の異なる値の数。レスポンス値が -1 の場合は、「CountDistinctLong」を参照してください。

タイプ: 整数

必須: いいえ

CountDistinctLong

フィールド内の個別の値の数。値が 2,147,483,647 より大きい場合、CountDistinct の代わりに CountDistinctLong が使用されます。

型: Long

必須: いいえ

CountLong

フィールド内の値の数。値が 2,147,483,647 より大きい場合、Count の代わりに CountLong が使用されます。

型: Long

必須: いいえ

### CountNan

フィールド内の NAN (非数値) の値の数。レスポンス値が -1 の場合は、「CountNanLong」を参照してください。

タイプ: 整数

必須: いいえ

### CountNanLong

フィールド内の NAN (非数値) の値の数。値が 2,147,483,647 より大きい場合、CountNan の代わりに CountNanLong が使用されます。

型: Long

必須: いいえ

### CountNull

フィールド内の null 値の数。レスポンス値が -1 の場合は、「CountNullLong」を参照してください。

タイプ: 整数

必須: いいえ

### CountNullLong

フィールド内の null 値の数。値が 2,147,483,647 より大きい場合、CountNull の代わりに CountNullLong が使用されます。

型: Long

必須: いいえ

### Max

数値フィールドについては、フィールド内の最大値。

型: 文字列

長さの制限：最大長は 256 です。

パターン: `^[a-zA-Z0-9\_]+$`

必須: いいえ

#### Min

数値フィールドについては、フィールド内の最小値。

型: 文字列

長さの制限：最大長は 256 です。

パターン: `^[a-zA-Z0-9\_]+$`

必須: いいえ

#### Stddev

数値フィールドについては、標準偏差。

型: 倍精度浮動小数点数

必須: いいえ

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## SupplementaryFeature

サービス: Amazon Forecast Service

### Note

このオブジェクトは [CreatePredictor](#) オペレーションに属しています。 [CreateAutoPredictor](#) で予測子を作成した場合は、「[AdditionalDataset](#)」を参照してください。

データセットグループの補足的な特徴を記述します。このオブジェクトは [InputDataConfig](#) オブジェクトの一部です。Forecast は、Weather Index と Holidays のビルトインの特徴量化をサポートしています。

### Weather Index

Amazon Forecast の Weather Index はビルトインの特徴量化であり、履歴気象情報および予測される気象情報をモデルに組み込みます。Weather Index は、2 年を超える期間にわたる履歴気象データと最大 14 日間の気象予報データでデータセットを補完します。詳細については、「[Amazon Forecast Weather Index](#)」(Amazon Forecast の Weather Index) を参照してください。

### Holidays

Holidays は、特徴エンジニアリングされた祝祭日情報データセットをモデルに組み込む、ビルトインの特徴量化です。250 か国を超える祝祭日カレンダーをネイティブにサポートします。Amazon Forecast には、ホリデーカレンダーを生成するための [Holiday API ライブラリ](#) と [Jollyday API](#) の両方が組み込まれています。詳細については、「[Holidays Featurization](#)」(Holidays の特徴量化) を参照してください。

### 内容

#### Name

機能の名前。有効な値: "holiday" および "weather"。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: はい

## Value

### Weather Index

Weather Index を有効にするには、値を "true" に設定します。

### Holidays

Holidays を有効にするには、次の 2 文字の国コードで国を指定します。

- アフガニスタン - AF
- オーランド諸島 - AX
- アルバニア - AL
- アルジェリア - DZ
- アメリカ領サモア - AS
- アンドラ - AD
- アンゴラ - AO
- アンギラ - AI
- 南極 - AQ
- アンティグアバーブーダ - AG
- アルゼンチン - AR
- アルメニア - AM
- アルバ - AW
- オーストラリア - AU
- オーストリア - AT
- アゼルバイジャン - AZ
- バハマ - BS
- バーレーン - BH
- バングラデシュ - BD
- バルバドス - BB
- ベラルーシ - BY
- ベルギー - BE
- ベリーズ - BZ
- ベナン - BJ

- バーミューダ - BM
- ブータン - BT
- ボリビア -BO
- ボスニアヘルツェゴビナ - BA
- ボツワナ - BW
- ブーベ島 - BV
- ブラジル - BR
- 英領インド洋地域 - IO
- 英領バージン諸島 - VG
- ブルネイ・ダルサラーム - BN
- ブルガリア - BG
- ブルキナファソ - BF
- ブルンジ - BI
- カンボジア - KH
- カメルーン - CM
- カナダ - CA
- カーボベルデ - CV
- オランダカリブ領域 - BQ
- ケイマン諸島 - KY
- 中央アフリカ共和国 - CF
- チャド - TD
- チリ - CL
- 中国 - CN
- クリスマス島 - CX
- ココス (キーリング) - CC
- コロンビア - CO
- コモロ - KM
- クック諸島 -CK
- コスタリカ - CR
- クロアチア - HR



- キューバ - CU
- キュラソー - CW
- キプロス - CY
- チェコ - CZ
- コンゴ民主共和国 - CD
- デンマーク - DK
- ジブチ - DJ
- ドミニカ国 - DM
- ドミニカ共和国 - DO
- エクアドル - EC
- エジプト - EG
- エルサルバドル - SV
- 赤道ギニア - GQ
- エリトリア - ER
- エストニア - EE
- エスワティニ - SZ
- エチオピア - ET
- フォークランド諸島 - FK
- フェロー諸島 - FO
- フィジー - FJ
- フィンランド - FI
- フランス - FR
- フランス領ギアナ - GF
- フランス領ポリネシア - PF
- フランス領南方・南極地域 - TF
- ガボン - GA
- ガンビア - GM
- ジョージア - GE
- ドイツ - DE
- ガーナ - GH

- ジブラルタル - GI
- ギリシャ - GR
- グリーンランド - GL
- グレナダ - GD
- グアドループ - GP
- グアム - GU
- グアテマラ - GT
- ガーンジー - GG
- ギニア - GN
- ギニアビサウ - GW
- ガイアナ - GY
- ハイチ - HT
- ハード島と McDonald 諸島 - HM
- ホンジュラス - HN
- 香港 - HK
- ハンガリー - HU
- アイスランド - IS
- インド - IN
- インドネシア - ID
- イラン - IR
- イラク - IQ
- アイルランド - IE
- マン島 - IM
- イスラエル - IL
- イタリア - IT
- コートジボワール - CI
- ジャマイカ - JM
- 日本 - JP
- ジャージー - JE
- ヨルダン - JO

- カザフスタン - KZ
- ケニア - KE
- キリバス - KI
- コソボ - XK
- クウェート - KW
- キルギスタン - KG
- ラオス - LA
- ラトビア - LV
- レバノン - LB
- レソト - LS
- リベリア - LR
- リビア - LY
- リヒテンシュタイン - LI
- リトアニア - LT
- ルクセンブルグ - LU
- マカオ - MO
- マダガスカル - MG
- マラウイ - MW
- マレーシア - MY
- モルディブ - MV
- マリ - ML
- マルタ - MT
- マーシャル諸島共和国 - MH
- マルティニーク - MQ
- モーリタニア - MR
- モーリシャス - MU
- マヨット - YT
- メキシコ - MX
- ミクロネシア - FM
- モルドバ - MD

- モナコ - MC
- モンゴル国 - MN
- モンテネグロ - ME
- モントセラト島 - MS
- モロッコ - MA
- モザンビーク - MZ
- ミャンマー - MM
- ナミビア - NA
- ナウル - NR
- ネパール - NP
- オランダ - NL
- ニューカレドニア - NC
- ニュージーランド - NZ
- ニカラグア - NI
- ニジェール - NE
- ナイジェリア - NG
- ニウエ - NU
- ノーフォーク島 - NF
- 北朝鮮 - KP
- 北マケドニア - MK
- 北マリアナ諸島 - MP
- ノルウェー - NO
- オマーン - OM
- パキスタン - PK
- パラオ - PW
- パレスチナ - PS
- パナマ - PA
- パプアニューギニア - PG
- パラグアイ - PY
- ペルー - PE

- フィリピン - PH
- ピトケアン諸島 - PN
- ポーランド - PL
- ポルトガル - PT
- プエルトリコ - PR
- カタール - QA
- コンゴ共和国 - CG
- レユニオン - RE
- ルーマニア - RO
- ロシア連邦 - RU
- ルワンダ - RW
- サン・バルテルミー - BL
- セントヘレナ・アセンションおよびトリスタンダクーニャ - SH
- セントクリストファーネイビス - KN
- セントルシア - LC
- サン・マルタン島 - MF
- サンピエール・ミクロン - PM
- セントビンセントおよびグレナディーン諸島 - VC
- サモア - WS
- サンマリノ - SM
- サントメプリンシペ - ST
- サウジアラビア - SA
- セネガル - SN
- セルビア - RS
- セイシェル - SC
- シエラレオネ - SL
- シンガポール - SG
- シントマーテルテン - SX
- スロバキア - SK
- スロベニア - SI

- ソロモン諸島 - SB
- ソマリア - SO
- 南アフリカ - ZA
- サウスジョージア・ サウスサンドウィッチ諸島 - GS
- 韓国 - KR
- 南スーダン - SS
- スペイン - ES
- スリランカ - LK
- スーダン - SD
- スリナム - SR
- スバルバル諸島およびヤンマイエン島 - SJ
- スウェーデン - SE
- スイス - CH
- シリア・ アラブ共和国 - SY
- 台湾 - TW
- タジキスタン - TJ
- タンザニア - TZ
- タイ - TH
- 東ティモール - TL
- トーゴ - TG
- トケラウ - TK
- トンガ - TO
- トリニダード・ トバゴ - TT
- チュニジア - TN
- トルコ - TR
- トルクメニスタン - TM
- タークスおよびカイコス諸島 - TC
- ツバル - TV
- ウガンダ - UG
- ウクライナ - UA

- アラブ首長国連邦 - AE
- 英国 - GB
- 国連 - UN
- 米国 - US
- 合衆国領有小離島 - UM
- アメリカ領バージン諸島 - VI
- ウルグアイ - UY
- ウズベキスタン - UZ
- バヌアツ - VU
- バチカン市国 - VA
- ベネズエラ - VE
- ベトナム - VN
- ウォリス・フツナ - WF
- 西サハラ - EH
- イエメン - YE
- ザンビア - ZM
- ジンバブエ - ZW

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `^[a-zA-Z0-9\_\-]+`

必須: はい

以下の資料も参照してください。

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)





## Tag

サービス: Amazon Forecast Service

サービスに適用し、リソースの分類と整理に役立つオプションのメタデータ。タグはそれぞれ、1つのキーとオプションの1つの値で設定されており、どちらもお客様側が定義します。

タグには以下のベーシックな制限があります。

- リソースあたりのタグの最大数は 50 です。
- タグキーは、リソースごとにそれぞれ一意である必要があります。また、各タグキーに設定できる値は 1 つのみです。
- キーの最大長 - 128 文字 (Unicode) (UTF-8)。
- 値の最大長 - 256 文字 (Unicode) (UTF-8)。
- 複数の のサービス間およびリソース間でタグ付けスキーマを使用する場合、他のサービスにも許容される文字数に制限がある可能性があることに注意してください。一般的に使用が許可される文字は、UTF-8 で表現可能な文字、数字、スペース、および以下の文字です:  
+、-、=、.、\_、:、/、@。
- タグのキーと値では、大文字と小文字が区別されます。
- aws:、AWS:、またはキーのプレフィックスなどの大文字または小文字の組み合わせは使用しないでください。AWS 使用のために予約されています。このプレフィックスが含まれるタグキーを編集したり削除することはできません。値にはこのプレフィックスを付けることができます。タグ値には aws がプレフィックスとして付されているが、キーには付されていない場合、Forecast は、それをユーザータグとみなし、タグ数の上限である 50 個を計算する際に算入します。aws のキープレフィックスのみを持つタグは、リソース制限あたりのタグに算入されません。

## 内容

### Key

タグを構成するキーと値のペアの一部。key は、より具体的なタグ値のカテゴリのように動作する、一般的なラベルです。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 128 です。

パターン: `^([\p{L}\p{Z}\p{N}_.:/+\\-@]*)$`

必須 : はい

## Value

タグを構成するキーと値のペアのオプションの一部。value はタグカテゴリー (キー) の記述子として機能します。

型: 文字列

長さの制限: 最小長は 0 です。最大長は 256 です。

パターン : `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

必須 : はい

以下の資料も参照してください。

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## TestWindowSummary

サービス: Amazon Forecast Service

バックテストのステータス、開始時間、終了時間、および該当する場合は失敗の理由。

内容

### Message

テストが失敗した場合、それが失敗した理由。

タイプ: 文字列

必須: いいえ

### Status

テストのステータス。可能なステータス値は次のとおりです。

- ACTIVE
- CREATE\_IN\_PROGRESS
- CREATE\_FAILED

型: 文字列

長さの制限: 最大長は 256 です。

必須: いいえ

### TestWindowEnd

テストが終了した時刻。

型: タイムスタンプ

必須: いいえ

### TestWindowStart

テストが開始された時刻。

型: タイムスタンプ

必須: いいえ

## その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## TimeAlignmentBoundary

サービス: Amazon Forecast Service

Forecast が、予想頻度に合わせてデータを調整および集計するために使用する時間境界です。時間単位と時間境界をキーバリューペアとして指定します。時間境界を指定しない場合、Forecast は [デフォルトの時間境界](#) のセットを使用します。

集計の詳細については、「[Data Aggregation for Different Forecast Frequencies](#)」を参照してください。カスタム時間境界の設定について詳しくは、「[Specifying a Time Boundary](#)」を参照してください。

内容

### DayOfMonth

集計中の時間調整に使用する日。

タイプ: 整数

有効範囲: 最小値は 1 です。最大値は 28 です。

必須: いいえ

### DayOfWeek

集計中の時間調整に使用する曜日。日付は大文字にする必要があります。

型: 文字列

有効な値: MONDAY | TUESDAY | WEDNESDAY | THURSDAY | FRIDAY | SATURDAY | SUNDAY

必須: いいえ

### Hour

集計中の時間調整に使用する時間。

タイプ: 整数

有効な範囲: 最小値は 0 です。最大値は 23 です。

必須: いいえ

## Month

集計中の時間調整に使用する月。月は大文字にする必要があります。

型: 文字列

有効な値 : JANUARY | FEBRUARY | MARCH | APRIL | MAY | JUNE | JULY | AUGUST  
| SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER

必須 : いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## TimeSeriesCondition

サービス: Amazon Forecast Service

変更された属性内の項目のサブセットを作成します。例えば、この操作を使用して 5 USD 以下のアイテムのサブセットを作成できます。そのためには、"AttributeName": "price"、"AttributeValue": "5"、"Condition": "LESS\_THAN" を指定します。このオペレーションを [CreateWhatIfForecast : TimeSeriesTransformations Action](#) オペレーション内の オペレーションと組み合わせて、属性の変更方法を定義します。

内容

### AttributeName

変更する item\_id、ディメンション名、IM 名、またはタイムスタンプ。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: はい

### AttributeValue

選択した Condition に適用される値。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `.+`

必須: はい

### Condition

適用する条件。有効な値は、EQUALS、NOT\_EQUALS、LESS\_THAN、GREATER\_THAN です。

型: 文字列

有効な値: EQUALS | NOT\_EQUALS | LESS\_THAN | GREATER\_THAN

必須: はい

以下の資料も参照してください。

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)



## TimeSeriesIdentifiers

サービス: Amazon Forecast Service

予想を作成する時系列を含むインポートファイルの詳細。

内容

### DataSource

データのソース、Amazon Forecast がデータにアクセスできるようにする AWS Identity and Access Management (IAM) ロール、およびオプションで AWS Key Management Service (KMS) キー。

タイプ: [DataSource](#) オブジェクト

必須: いいえ

### Format

データの形式は CSV または PARQUET のいずれかです。

型: 文字列

長さの制限: 最大長は 7 です。

Pattern: ^CSV|PARQUET\$

必須: いいえ

### Schema

データセットのフィールドを定義します。

タイプ: [Schema](#) オブジェクト

必須: いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)

- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## TimeSeriesReplacementsDataSource

サービス: Amazon Forecast Service

代替データセットは、What-If 予想で含みたい値のみを含むベースライン関連の時系列を修正したバージョンです。代替データセットには、ベースライン関連の時系列の予想ディメンションおよびアイテム識別子と、少なくとも 1 つの変更された時系列が含まれている必要があります。このデータセットをベースライン関連の時系列と統合して、What-If 予想に使用される変換済みデータセットを作成します。

### 内容

#### S3Config

Amazon Simple Storage Service (Amazon S3) バケット内のファイルへのパス、および Amazon Forecast がファイルにアクセスするために引き受けることができる AWS Identity and Access Management (IAM) ロール (複数可)。オプションで、には AWS Key Management Service (KMS) キーが含まれます。このオブジェクトは、[CreateDatasetImportJob](#) リクエストで送信された [DataSource](#) オブジェクトの一部であり、かつ、[DataDestination](#) オブジェクトの一部です。

型: [S3Config](#) オブジェクト

必須: はい

#### Schema

データセットのフィールドを定義します。

型: [Schema](#) オブジェクト

必須: はい

#### Format

代替データセットの形式は CSV である必要があります。

型: 文字列

長さの制限: 最大長は 7 です。

Pattern: ^CSV|PARQUET\$

必須: いいえ

## TimestampFormat

代替データのタイムスタンプ形式。

型: 文字列

長さの制限 : 最大長は 256 です。

パターン: `^[a-zA-Z0-9\-\:\.\,\'\s]+$`

必須: いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## TimeSeriesSelector

サービス: Amazon Forecast Service

`TimeSeriesIdentifiers` オブジェクト内の予想を作成するために使用される時系列のセットを定義します。

`TimeSeriesIdentifiers` オブジェクトには次の情報が必要です。

- `DataSource`
- `Format`
- `Schema`

### 内容

#### `TimeSeriesIdentifiers`

予想を作成する時系列を含むインポートファイルの詳細。

タイプ: [TimeSeriesIdentifiers](#) オブジェクト

必須: いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## TimeSeriesTransformation

サービス: Amazon Forecast Service

変換関数は、関連する時系列の行を選択して変更する 2 つのオペレーションです。条件操作で目的の行を選択し、変換オペレーションで行を変更します。すべての条件は AND 演算で結合されます。つまり、変換を適用するにはすべての条件を満たす必要があります。変換は、記載された順序で適用されます。

内容

### Action

時系列とその変換方法を定義するアクションの配列。これらの変換によって新しい時系列が作成され、その時系列が what-if 分析に使用されます。

タイプ: [Action](#) オブジェクト

必須: いいえ

### TimeSeriesConditions

関連する時系列のどのメンバーが変換されるかを定義する条件の配列。

タイプ: [TimeSeriesCondition](#) オブジェクトの配列

の配列メンバー: 最小数は 0 項目です。最大数は 10 項目です。

必須: いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## WeightedQuantileLoss

サービス: Amazon Forecast Service

分位数の重み付き分位損失。このオブジェクトは [Metrics](#) オブジェクトの一部です。

内容

LossValue

すべての分位数の合計で除することによって重み付け (正規化) された、分位数の予測値と実際の値の差。

型: 倍精度浮動小数点数

必須: いいえ

Quantile

分位数。分位数は、確率分布を等しい確率の領域に分割します。例えば、分布が等しい確率の 5 つの領域に分割された場合、分位数は 0.2、0.4、0.6、および 0.8 になります。

型: 倍精度浮動小数点数

必須: いいえ

その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## WhatIfAnalysisSummary

サービス: Amazon Forecast Service

[ListWhatIfAnalyses](#) オペレーションで使用される What-If 分析のプロパティの概要を提供します。プロパティの完全なセットを取得するには、[DescribeWhatIfAnalysis](#) オペレーションを呼び出して、概要にリストされている `WhatIfAnalysisArn` を指定します。

内容

### CreationTime

What-If 分析が作成された日時。

型: タイムスタンプ

必須: いいえ

### ForecastArn

この What-If 分析で使用されるベースライン予想の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

必須: いいえ

### LastModificationTime

リソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。

- `CREATE_PENDING` (CreationTime)
- `CREATE_IN_PROGRESS` - 現在のタイムスタンプ。
- `CREATE_STOPPING` - 現在のタイムスタンプ。
- `CREATE_STOPPED` - ジョブが停止した時点。
- `ACTIVE` または `CREATE_FAILED` - ジョブが終了または失敗した時点。

型: タイムスタンプ

必須: いいえ



## Message

エラーが発生した場合は、エラーに関する情報メッセージ。

タイプ: 文字列

必須: いいえ

## Status

What-If 分析のステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED

### Note

分析にアクセスするには、What-If 分析の Status が ACTIVE である必要があります。

型: 文字列

長さの制限: 最大長は 256 です。

必須: いいえ

## WhatIfAnalysisArn

What-If 分析の Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: いいえ

## WhatIfAnalysisName

What-If 分析の名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: いいえ

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## WhatIfForecastExportSummary

サービス: Amazon Forecast Service

[ListWhatIfForecastExports](#) オペレーションで使用される What-If 予想のエクスポートプロパティの概要を提供します。プロパティの完全なセットを取得するには、[DescribeWhatIfForecastExport](#) オペレーションを呼び出して、概要にリストされている `WhatIfForecastExportArn` を指定します。

内容

### CreationTime

What-If 予想エクスポートが作成された日時。

型: タイムスタンプ

必須: いいえ

### Destination

予測がエクスポートされる Amazon Simple Storage Service (Amazon S3) バケットへのパス。

タイプ: [DataDestination](#) オブジェクト

必須: いいえ

### LastModificationTime

リソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。

- CREATE\_PENDING (CreationTime)
- CREATE\_IN\_PROGRESS - 現在のタイムスタンプ。
- CREATE\_STOPPING - 現在のタイムスタンプ。
- CREATE\_STOPPED - ジョブが停止した時点。
- ACTIVE または CREATE\_FAILED - ジョブが終了または失敗した時点。

型: タイムスタンプ

必須: いいえ

### Message

エラーが発生した場合は、エラーに関する情報メッセージ。

タイプ: 文字列

必須: いいえ

## Status

What-If 予想エクスポートのステータス。含まれる状態は次のとおりです。

- ACTIVE
- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED

### Note

分析にアクセスするには、What-If 分析の Status が ACTIVE である必要があります。

型: 文字列

長さの制限: 最大長は 256 です。

必須: いいえ

## WhatIfForecastArns

エクスポートに含まれる What-If 予想を定義する Amazon リソースネーム (ARN) の配列。

タイプ: 文字列の配列

配列メンバー: 最小数は 1 項目です。最大数は 50 項目です。

長さの制限: 最大長は 300 です。

Pattern: `arn:([a-z\d-]+):forecast:.*:.*:.*`

必須: いいえ

## WhatIfForecastExportArn

What-If 予想エクスポートの Amazon リソースネーム (ARN)。

型: 文字列

長さの制限: 最大長は 300 です。

Pattern: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

必須: いいえ

### WhatIfForecastExportName

What-If 予想エクスポートの名前。

型: 文字列

長さの制限: 最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: いいえ

### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## WhatIfForecastSummary

サービス: Amazon Forecast Service

[ListWhatIfForecasts](#) オペレーションで使用される What-If 予想のプロパティの概要を提供します。プロパティの完全なセットを取得するには、[DescribeWhatIfForecast](#) オペレーションを呼び出して、概要にリストされている WhatIfForecastArn を指定します。

内容

### CreationTime

What-If 予想が作成された日時。

型: タイムスタンプ

必須: いいえ

### LastModificationTime

リソースの最終変更時刻。タイムスタンプは、ジョブのステータスによって異なります。

- CREATE\_PENDING (CreationTime)
- CREATE\_IN\_PROGRESS - 現在のタイムスタンプ。
- CREATE\_STOPPING - 現在のタイムスタンプ。
- CREATE\_STOPPED - ジョブが停止した時点。
- ACTIVE または CREATE\_FAILED - ジョブが終了または失敗した時点。

型: タイムスタンプ

必須: いいえ

### Message

エラーが発生した場合は、エラーに関する情報メッセージ。

タイプ: 文字列

必須: いいえ

### Status

What-If 予想のステータス。含まれる状態は次のとおりです。

- ACTIVE

- CREATE\_PENDING, CREATE\_IN\_PROGRESS, CREATE\_FAILED
- CREATE\_STOPPING, CREATE\_STOPPED
- DELETE\_PENDING, DELETE\_IN\_PROGRESS, DELETE\_FAILED

 Note

分析にアクセスするには、What-If 分析の Status が ACTIVE である必要があります。

型: 文字列

長さの制限: 最大長は 256 です。

必須: いいえ

#### WhatIfAnalysisArn

この What-If 予想を含む What-If 分析の Amazon リソース名前 (ARN)。

型: 文字列

長さの制限: 最大長は 256 です。

パターン: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

必須: いいえ

#### WhatIfForecastArn

What-If 予想の Amazon リソース名前 (ARN)。

型: 文字列

長さの制限: 最大長は 300 です。

Pattern: `arn:([a-z\d-]+):forecast:.*:.*:.*+`

必須: いいえ

#### WhatIfForecastName

What-If 予想の名前。

型: 文字列

長さの制限：最小長は 1 です。最大長は 63 です。

パターン: `^[a-zA-Z][a-zA-Z0-9_]*`

必須: いいえ

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)



## WindowSummary

サービス: Amazon Forecast Service

データセットの評価部分内の時間範囲のメトリクス。このオブジェクトは [EvaluationResult](#) オブジェクトの一部です。

TestWindowStart および TestWindowEnd パラメータは、[EvaluationParameters](#) オブジェクトの BackTestWindowOffset パラメータによって決定されます。

内容

### EvaluationType

評価のタイプ。

- SUMMARY - すべてのウィンドウの平均メトリクス。
- COMPUTED - 指定されたウィンドウのメトリクス。

型: 文字列

有効な値 : SUMMARY | COMPUTED

必須 : いいえ

### ItemCount

ウィンドウ内のデータポイントの数。

タイプ: 整数

必須: いいえ

### Metrics

予測子のパフォーマンスを評価するために使用されるメトリクスを提供します。

タイプ : [Metrics](#) オブジェクト

必須: いいえ

### TestWindowEnd

ウィンドウの終わりを定義するタイムスタンプ。

型: タイムスタンプ

必須: いいえ

TestWindowStart

ウィンドウの開始を定義するタイムスタンプ。

型: タイムスタンプ

必須: いいえ

その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## Amazon Forecast クエリサービス

次のデータ型は、Amazon Forecast Query Service でサポートされています:

- [DataPoint](#)
- [Forecast](#)

## DataPoint

サービス: Amazon Forecast Query Service

特定の日付の予測値。 [Forecast](#) オブジェクトの一部。

内容

Timestamp

特定の予測のタイムスタンプ。

タイプ: 文字列

必須: いいえ

Value

予測値。

型: 倍精度浮動小数点数

必須: いいえ

その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## Forecast

サービス: Amazon Forecast Query Service

予測に関する情報を提供します。[QueryForecast](#) レスポンスの一部として返されます。

### 内容

#### Predictions

予測。

string-to-array マップの文字列は、次のいずれかの値です。

- p10
- p50
- p90

デフォルトの設定は、["0.1", "0.5", "0.9"] です。[CreateForecast](#) オペレーションのオプション `ForecastTypes` パラメータを使用して、値を変更します。値は、これがどのように設定されているかによって異なり、最小値を 1、最大値を 5 とします。

タイプ: [DataPoint](#) オブジェクトマップの配列への文字列

キーの長さの制限: 最大長は 4 です。

必須: いいえ

#### その他の参照資料

言語固有の AWS SDKs のいずれかでこの API を使用方法の詳細については、以下を参照してください。

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## 共通エラー

このセクションでは、AWS のすべてのサービスの API アクションに共通のエラーを一覧表示しています。このサービスの API アクションに固有のエラーについては、その API アクションのトピックを参照してください。

### AccessDeniedException

このアクションを実行する十分なアクセス権限がありません。

HTTP ステータスコード: 400

### IncompleteSignature

リクエストの署名が AWS 基準に適合しません。

HTTP ステータスコード: 400

### InternalFailure

リクエストの処理が、不明なエラー、例外、または障害により実行できませんでした。

HTTP ステータスコード: 500

### InvalidAction

リクエストされたアクション、またはオペレーションは無効です。アクションが正しく入力されていることを確認します。

HTTP ステータスコード: 400

### InvalidClientTokenId

指定された x.509 証明書、または AWS アクセスキー ID が見つかりません。

HTTP ステータスコード: 403

### NotAuthorized

このアクションを実行するにはアクセス許可が必要です。

HTTP ステータスコード: 400

### OptInRequired

サービスを利用するためには、AWS アクセスキー ID を取得する必要があります。

HTTP ステータスコード: 403

#### RequestExpired

リクエストの日付スタンプの 15 分以上後またはリクエストの有効期限 (署名付き URL の場合など) の 15 分以上後に、リクエストが到着しました。または、リクエストの日付スタンプが現在より 15 分以上先です。

HTTP ステータスコード: 400

#### ServiceUnavailable

リクエストは、サーバーの一時的障害のために実行に失敗しました。

HTTP ステータスコード: 503

#### ThrottlingException

リクエストは、制限が必要なために実行が拒否されました。

HTTP ステータスコード: 400

#### ValidationError

入力が、AWS サービスで指定された制約を満たしていません。

HTTP ステータスコード: 400

## 共通パラメータ

次のリストには、すべてのアクションが署名バージョン 4 リクエストにクエリ文字列で署名するために使用するパラメータを示します。アクション固有のパラメータは、アクションのトピックに示されています。Signature Version 4 の詳細については、「IAM ユーザーガイド」の「[AWS API リクエストの署名](#)」を参照してください。

#### Action

実行するアクション。

型: 文字列

必須: はい

#### Version

リクエストが想定している API バージョンである、YYYY-MM-DD 形式で表示されます。

型: 文字列

必須: はい

### X-Amz-Algorithm

リクエストの署名を作成するのに使用したハッシュアルゴリズム。

条件: HTTP 認証ヘッダーではなくクエリ文字列に認証情報を含める場合は、このパラメータを指定します。

型: 文字列

有効な値: AWS4-HMAC-SHA256

必須: 条件による

### X-Amz-Credential

認証情報スコープの値で、アクセスキー、日付、対象とするリージョン、リクエストしているサービス、および終了文字列 ("aws4\_request") を含む文字列です。値は次の形式で表現されます。[access\_key/YYYYYYYYMMDD/リージョン/サービス/aws4\_request]

詳細については、「IAM ユーザーガイド」の「[署名付きAWS API リクエストの作成](#)」を参照してください。

条件: HTTP 認証ヘッダーではなくクエリ文字列に認証情報を含める場合は、このパラメータを指定します。

型: 文字列

必須: 条件による

### X-Amz-Date

署名を作成するときに使用する日付です。形式は ISO 8601 基本形式の YYYYMMDD'T'HHMMSS'Z' でなければなりません。例えば、日付 20120325T120000Z は、有効な X-Amz-Date の値です。

条件: X-Amz-Date はすべてのリクエストに対してオプションです。署名リクエストで使用する日付よりも優先される日付として使用できます。ISO 8601 ベーシック形式で日付ヘッダーが指定されている場合、X-Amz-Date は必要ありません。X-Amz-Date を使用すると、常に Date ヘッダーの値よりも優先されます。詳細については、「IAM ユーザーガイド」の「[AWS API リクエスト署名の要素](#)」を参照してください。

タイプ: 文字列

必須: 条件による

#### X-Amz-Security-Token

AWS Security Token Service (AWS STS) への呼び出しで取得された一時的なセキュリティトークン。AWS STS の一時的なセキュリティ認証情報をサポートするサービスのリストについては、「IAM ユーザーガイド」の「[IAM と連携するAWS のサービス](#)」を参照してください。

条件: AWS STS の一時的なセキュリティ認証情報を使用する場合、セキュリティトークンを含める必要があります。

タイプ: 文字列

必須: 条件による

#### X-Amz-Signature

署名する文字列と派生署名キーから計算された 16 進符号化署名を指定します。

条件: HTTP 認証ヘッダーではなくクエリ文字列に認証情報を含める場合は、このパラメータを指定します。

型: 文字列

必須: 条件による

#### X-Amz-SignedHeaders

正規リクエストの一部として含まれていたすべての HTTP ヘッダーを指定します。署名付きヘッダーの指定に関する詳細については、「IAM ユーザーガイド」の「[署名付き AWS API リクエストの作成](#)」を参照してください。

条件: HTTP 認証ヘッダーではなくクエリ文字列に認証情報を含める場合は、このパラメータを指定します。

型: 文字列

必須: 条件による



# Amazon Forecast のドキュメント履歴

次の表は、Amazon Forecast デベロッパーガイドの重要な変更点を示しています。ドキュメントの更新に関する通知については、RSS フィードにサブスクライブできます。

- ドキュメント最新更新日: 2021 年 3 月 3 日

変更	説明	日付
<a href="#">新機能</a>	カスタム予想頻度を指定できるようになりました。詳細については、「 <a href="#">Creating a Predictor</a> 」を参照してください。	2022 年 8 月 29 日
<a href="#">新機能</a>	What-If 分析を作成して、さまざまなシナリオを検討できるようになりました。詳細については、「 <a href="#">What-if analysis</a> 」を参照してください。	2022 年 8 月 22 日
<a href="#">新機能</a>	Amazon EventBridge または Amazon CloudWatch Events に、進行中の Amazon Forecast リソースジョブのステータスアップデートを通知させることができるようになりました。詳細については、「 <a href="#">通知の設定</a> 」を参照してください。	2021 年 3 月 15 日
<a href="#">新機能</a>	Amazon Forecast は、データセットのインポートジョブ、予測子、予測子バックテストエクスポートジョブ、予測、および予測エクスポートジョブ	2021 年 3 月 3 日

ブの手動停止をサポートするようになりました。

### 新機能

Amazon Forecast は、履歴気象情報および予測された気象情報をモデルに自動的に組み込むビルトインの特徴量化をサポートするようになりました。詳細については、「[Amazon Forecast Weather Index](#)」(Amazon Forecast の Weather Index) を参照してください。

2020 年 12 月 8 日

### 新機能

Amazon Forecast は、予測子バックテスト予測と精度メトリクスをエクスポートする機能をサポートするようになりました。詳細については、「[Evaluating Predictor Accuracy](#)」(予測子の精度の評価) を参照してください。

2020 年 11 月 23 日

### 新機能

Amazon Forecast で、予測分位を指定する機能が追加されました。詳細については、「[Evaluating Predictor Accuracy](#)」(予測子の精度の評価) を参照してください。

2020 年 11 月 11 日

### 新機能

Amazon Forecast は、CNN-QR アルゴリズムをサポートするようになりました。詳細については、「[CNN-QR](#)」を参照してください。

2020 年 8 月 10 日

## 新機能

Amazon Forecast は、データセットグループ、データセット、データセットのインポートジョブ、予測子、予測、および予測エクスポートジョブのリソースのタグ付けをサポートするようになりました。詳細については、「[Amazon Forecast リソースにタグを付ける](#)」を参照してください。

2020 年 7 月 9 日

## 新機能

Amazon Forecast では、関連する時系列データセットの欠落した値の filling がサポートされるようになりました。詳細については、「[欠落した値の処理](#)」を参照してください。

2020 年 5 月 14 日

## 新しいリージョン

Amazon Forecast で、アジアパシフィック (ソウル)、アジアパシフィック (ムンバイ)、および欧州 (フランクフルト) の各リージョンのサポートが追加されました。Amazon Forecast でサポートされているすべての AWS リージョンのリストについては、[AWS リージョン表](#)または AWS 全般のリファレンスの [AWS リージョンとエンドポイント](#)を参照してください。

2020 年 3 月 17 日

<a href="#">新しいリージョン</a>	Amazon Forecast で、アジアパシフィック (ソウル) リージョンのサポートが追加されました。Amazon Forecast でサポートされているすべての AWS リージョンのリストについては、 <a href="#">AWS リージョン表</a> または AWS 全般のリファレンスの <a href="#">AWS リージョンとエンドポイント</a> を参照してください。	2020 年 1 月 27 日
<a href="#">新機能</a>	Forecast で、予測分位を指定する機能が追加されました。詳細については、Forecast API ガイドの「 <a href="#">CreateForecast</a> 」を参照してください。	2019 年 11 月 22 日
<a href="#">Amazon Forecast の一般利用</a>	Amazon Forecast が一般的にご利用いただけるようになりました。	2019 年 8 月 21 日
<a href="#">Amazon Forecast のプレビューのリリース</a>	本書は Amazon Forecast ドキュメントのプレビュー版の初版リリースです。	2018 年 11 月 28 日

# AWS 用語集

AWS の最新の用語については、「AWS の用語集リファレンス」の「[AWS 用語集](#)」を参照してください。

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。