aws

デベロッパーガイド

Amazon Machine Learning

Version Latest

Amazon Machine Learning: デベロッパーガイド

Table of Contents

Amazon Machine Learning サービスの更新や、その新しいユーザーの受け入れは行っていません。このドキュメントは既存のユーザー向けに提供されていますが、更新は終了しています。詳細については、「Amazon Machine Learning とは」を参照してください。

Amazon Machine Learning とは

Amazon Machine Learning (Amazon ML) サービスの更新や、その新しいユーザーの受け入れは行っていません。このドキュメントは既存のユーザー向けに提供されていますが、更新は終了しています。

AWS では現在、堅牢なクラウドベースのサービスである Amazon SageMakerを提供しているため、あらゆるスキルレベルの開発者が機械学習テクノロジーを利用できます。SageMaker は、強力な機械学習モデルを作成できる、フルマネージド型の機械学習サービスです。SageMaker では、データサイエンティストやデベロッパーが機械学習モデルの構築とトレーニングを行うことができ、それらを稼働準備が整ったホストされている環境に直接デプロイできます。

詳細については、「SageMaker に関するドキュメント」を参照してください。

トピック

Amazon Machine Learning の主要なコンセプト

このセクションでは、以下の主要なコンセプトをまとめ、Amazon ML でどのように使用されているかを詳しく説明します。

- データソース には Amazon ML への入力データと関連付けられたメタデータが含まれています
- ML モデル は、入力データから抽出されたパターンを使用して予測を生成します
- 評価 は ML モデルの品質を測定します。
- バッチ予測 は複数の入力データ監視に対し、非同期的に予測を生成します。
- リアルタイム予測 は個々のデータ監視に対し、同期的に予測を生成します。

データソース

データソースは、入力データに関するメタデータを含むオブジェクトです。Amazon ML は入力データを読み出し、属性の詳細な統計情報をコンピューティングし、データソースオブジェクトの一部と

して、スキーマとその他の情報とともに、統計を保存します。次に、Amazon ML はデータソースを使用して ML モデルをトレーニング、評価して、バッチ予測を生成します。

> **Important**
>
> データソースには、入力データのコピーは保存されません。代わりに、入力データがある Amazon S3 の場所への参照が保存されます。Amazon S3 ファイルを移動または変更した場合、Amazon ML は ML モデルの作成、評価の生成、または予測の生成のためにそれにアクセスする、または使用することはできなくなります。

次の表では、データソースに関連する用語が定義されています。

期間	定義
属性	観測値の中の、一意の、名前の付いたプロパティです。スプレッドシートまたはコンマ区切り値 (CSV) ファイルなどの、表形式のデータでは、列見出しは属性を表し、行には各属性の値が表示されます。 シノニム: 変数、変数名、フィールド、列
データソース名	(オプション) データソースに人間が読み取れる名前を指定できます。これらの名前を使用すると、Amazon ML コンソールでデータソースの検索および管理ができます。
入力データ	データソースにで使用されるすべての観測値に対する集合的な名前。
ロケーション	入力データの場所。現在、Amazon ML は Amazon S3 バケット、Amazon Redshift データベース、または Amazon Relational Database Service (RDS) にある MySQL データベースの中に保存されているデータを使用できます。
監視結果	単一の入力データの単位です。たとえば、不正な取引を検出するために ML モデルを作成する場合、入力データは多くの観測値から構成され、それぞれが個々のトランザクションを表します。 シノニム: レコード、例、インスタンス、行

期間	定義
行 ID	(オプション) フラグは、もし指定する場合、予測出力に含まれる入力データ内の属性を示します。この属性を使用すると、予測と観測の対応性を関連付けしやすくなります。 シノニム: 行識別子
スキーマ	入力データを解釈するために必要な情報のことで、属性の名前および割り当てられたデータタイプ、特殊な属性の名前などが含まれます。
統計	入力データの各属性の統計の概要。これらの統計には 2 つの目的があります。 Amazon ML コンソールではグラフで表示され、データを一目で理解し、不規則性やエラーを特定するのに役立ちます。 Amazon ML はトレーニングプロセス中にそれらを使用し、作成される ML モデルの品質を向上させます。
ステータス	データソースの現在の状態を示します (進行中、完了、または失敗など)。
ターゲット属性	ML モデルのトレーニングにおいて、ターゲット属性は、「正しい」回答を含む入力データ内の属性の名前を識別します。Amazon ML では、これを使用して入力データ内のパターンを検出し、ML モデルを生成します。予測の評価と生成において、ターゲット属性はトレーニングされた ML モデルにより予測される値を持つ属性です。 シノニム: ターゲット

ML モデル

ML モデルは、データにパターンを見出すことで予測を生成する数学モデルです。Amazon ML は、バイナリ分類、複数クラス分類、回帰の 3 つのタイプの ML モデルに対応しています。

次の表では、ML モデルに関連する用語が定義されています。

期間	定義
回帰	回帰 ML モデルのトレーニングにおける目標は、数値を予測することです。
複数クラス	複数クラス ML モデルのトレーニングにおける目標は、制限され、事前定義された、一連の許容値に属する値を予測することです。
バイナリ	バイナリ ML モデルのトレーニングにおける目標は、true または false のように 2 つの状態のいずれかとなる値を予測することです。
モデルサイズ	ML モデルはパターンをキャプチャして保存します。ML モデルは、保存するパターンが多いほど、より大きくなります。ML モデルサイズは MB 単位で表されます。
パスの数	ML モデルをトレーニングするときは、データソースからのデータを使用します。各データレコードを学習プロセスの間に複数回利用することにメリットがある場合があります。Amazon ML が同じデータレコードを使用するのを許可した回数をパスの数と呼びます。
正則化	正則化は、高品質なモデルを得るために使用できる機械学習の手法です。Amazon ML は、ほとんどの場合は、デフォルトの設定でうまく機能します。

評価

評価は、ML モデルの品質を測定し、パフォーマンスに問題がないかを判断します。

次の表では、評価に関連する用語が定義されています。

期間	定義
モデルインサイト	Amazon ML が提供するメトリクスと多数の洞察を活用して、モデルの予測パフォーマンスを評価できます。
AUC	ROC の曲線下面積 (AUC) は、バイナリ ML モデルの能力を測定して、正の例についてより高いスコアを予測し負の例と比較します。

期間	定義
平均 F1 スコア	マクロ平均 F1 スコアは、複数クラス ML モデルの予測パフォーマンスを評価するために使用します。
RMSE	二乗平均平方根誤差 (RMSE) は、回帰 ML モデルの予測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です。
カットオフ	ML モデルは数値予測スコアを生成することで機能します。カットオフ値を適用することで、システムはこれらのスコアを 0 と 1 のラベルに変換します。
Accuracy	精度は正しい予測の割合 (%) を測定します。
精度	精度は、取得されたインスタンス (正と予測されたもの) のうち、実際の正の (誤検出ではない) インスタンスの割合を示します。つまり、選択された項目のうち、正であるものの数です。
リコール	リコールは、該当するインスタンス (実際の正) の合計数のうち、実際の正の割合を示します。つまり、選択された正の項目の数です。

バッチ予測

バッチ予測は、すべてを一度に実行できる一連の観測です。これが最も適しているのは、リアルタイムの要件がない予測分析です。

次の表では、バッチ予測に関連する用語が定義されています。

期間	定義
Output Location	バッチ予測の結果は S3 バケットの出力場所に保存されます。
マニフェストファイル	このファイルは、各入力データファイルを、関係するバッチ予測の結果に関連付けます。これは S3 バケットの出力場所に保存されます。

リアルタイム予測

リアルタイム予測は、インタラクティブなウェブ、モバイル、またはデスクトップアプリケーションなど、低レイテンシーの要件があるアプリケーションに適しています。低レイテンシーのリアルタイム API を使用して、任意の ML モデルに対して予測のためのクエリを実行できます。

次の表では、リアルタイム予測に関連する用語が定義されています。

期間	定義
リアルタイム予測 API	リアルタイム予測 API は、リクエストペイロードで 1 つの入力観測を受け入れ、レスポンスで予測を返します。
リアルタイム予測エンドポイント	リアルタイム予測 API で ML モデルを使用するには、リアルタイム予測エンドポイントを作成する必要があります。一度作成されると、エンドポイントにはリアルタイム予測をリクエストするために使用できる URL が含まれます。

Amazon Machine Learning へのアクセス

次のいずれかを使用して Amazon ML にアクセスできます。

Amazon ML コンソール

Amazon ML コンソールにアクセスするには、AWS マネジメントコンソールにサインインして Amazon ML コンソール (https://console.aws.amazon.com/machinelearning/) を開きます。

AWS CLI

AWS CLI をインストールして設定する方法については、「AWS Command Line Interface ユーザーガイド」の「AWS コマンドラインインターフェイスの設定」を参照してください。

Amazon ML API

Amazon ML API の詳細については、「Amazon ML API リファレンス」を参照してください。

AWS SDK

AWS SDK については、「アマゾン ウェブ サービスのツール」を参照してください。

のリージョンとエンドポイント

Amazon Machine Learning (Amazon ML) は、リアルタイム予測エンドポイントを次の 2 つのリージョンでサポートしています。

リージョン名	リージョン	エンドポイント	プロトコル
米国東部 (バージニア北部)	us-east-1	machinelearning.us-east-1.amazonaws.com	HTTPS
欧州 (アイルランド)	eu-west-1	machinelearning.eu-west-1.amazonaws.com	HTTPS

データセットをホストし、モデルをトレーニングおよび評価して、どのリージョンでも予測をトリガーできます。

すべてのリソースを同じリージョンに保管しておくことをお勧めします。入力データが Amazon ML リソースとは異なるリージョンにある場合は、リージョン間のデータ転送料金が発生します。リアルタイム予測エンドポイントは、どのリージョンからでも呼び出すことができますが、呼び出すエンドポイントがないリージョンからエンドポイントを呼び出すと、リアルタイム予測のレイテンシーに影響する可能性があります。

Amazon ML の料金

AWS サービスでは、利用した分のみのお支払いとなります。最低料金や前払いの義務は発生しません。

Amazon Machine Learning (Amazon ML) は、データ統計の計算およびモデルのトレーニングと評価に使用される時間単位の料金と、アプリケーションのために生成された予測の数に応じた料金が請求されます。リアルタイムの予測に対しては、モデルのサイズに基づいて時間単位のリザーブドキャパシティー料金もかかります。

Amazon ML は Amazon ML コンソール でのみ予測のコストを見積もります。

Amazon ML の料金の詳細については、「Amazon Machine Learning の料金」を参照してください。

トピック

バッチ予測コストの見積り

バッチ予測の作成ウィザードを使用して Amazon ML モデルからバッチ予測をリクエストすると、Amazon ML はこれらの予測のコストを見積もります。見積もりを計算する方法は、利用可能なデータのタイプに応じて異なります。

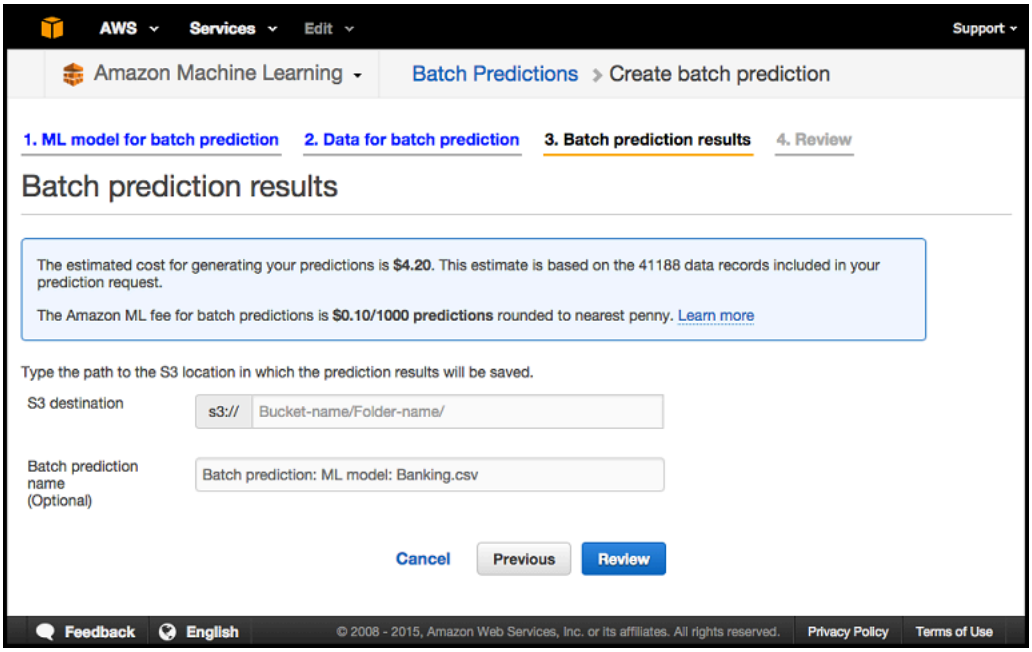
データ統計が利用可能な場合のバッチ予測コストの見積もり

最も正確なコスト見積もりは Amazon ML が予測をリクエストするために使用されたデータソースでサマリー統計をすでに計算している場合に取得されます。これらの統計は、Amazon ML コンソールを使用して作成されたデータソースに対して常に計算されます。API ユーザーは、CreateDataSourceFromS3、CreateDataSourceFromRedshift、CreateDataSourceFromRDS API を使用してデータソースをプログラムで作成するときは、`ComputeStatistics` フラグを `True` に設定する必要があります。統計を使用可能にするには、データソースが `READY` 状態でなければなりません。

Amazon ML が計算する統計の 1 つは、データレコードの数です。データレコードの数が利用可能な場合、Amazon ML バッチ予測の作成ウィザードは、データレコードの数に、バッチ予測の料金を掛けて予測数を見積もります。

実際のコストは、次の理由によりこの見積もりと異なる場合があります。

- 一部のデータレコードの処理が失敗する可能性があります。失敗したデータレコードからの予測については請求されません。
- 見積もりには、既存のクレジットや AWS によって適用されるその他の調整は考慮されていません。



データサイズのみが利用可能な場合のバッチ予測コストの見積もり

バッチ予測をリクエストし、リクエストデータソースのデータ統計が利用できない場合、Amazon ML は以下に基づいてコストを見積もります。

- データソースの検証中に計算され保持される合計データサイズ
- Amazon ML がデータファイルの最初の 100 MB を読み込み、解析することによって推定する平均データレコードサイズ

バッチ予測のコストを見積もるために、Amazon ML は合計データサイズを平均データレコードサイズで割ります。このコスト予測の方法は、データファイルの最初のレコードが平均レコードサイズを正確に表していない可能性があるため、データレコードの数が利用可能なときに使用された方法よりも正確ではありません。

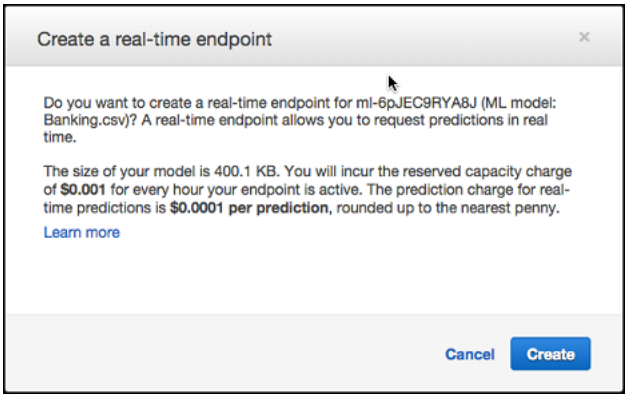
データ統計とデータサイズが利用できない場合のバッチ予測コストの見積もり

データ統計もデータサイズも利用できない場合、Amazon ML はバッチ予測のコストを見積もることはできません。バッチ予測をリクエストするために使用しているデータソースがまだ Amazon ML によって検証されていない場合、これが一般的です。これは、Amazon Redshift (Amazon Redshift) または Amazon Relational Database Service (Amazon RDS) クエリに基づいたデータソースを作成し、データ転送がまだ完了していない場合や、データソースの作成がアカウントで他の操作の後にキューイングされている場合に発生します。この場合、Amazon ML コンソールはバッチ予測の料金について通知します。見積もりなしでバッチ予測リクエストを続行するか、ウィザードをキャンセル

して予測に使用するデータソースが INPROGRESS または READY 状態になった後に戻ることができます。

リアルタイム予測コストの見積り

Amazon ML コンソールを使用してリアルタイム予測エンドポイントを作成すると、予測処理のためにエンドポイントを予約するための継続的な料金である推定キャパシティー予約料金が表示されます。この料金は、「サービスの料金ページ」で説明されているように、モデルのサイズによって異なります 。また、標準の Amazon ML リアルタイム予測料金についてもお知らせします。



機械学習の概念

機械学習 (ML) は、履歴データを使用してより良いビジネス上の意思決定をするのに役立ちます。ML アルゴリズムはデータ内のパターンを発見し、これらの発見を使って数学的モデルを構築します。その後、そのモデルを使用して将来のデータを予測することができます。たとえば、機械学習モデルの 1 つの考えられる用途は、顧客が過去の行動に基づいて特定の製品を購入する可能性を予測することです。

トピック

Amazon Machine Learning でビジネス上の問題を解決する

Amazon Machine Learning を使用して、実際の回答の例がある問題に機械学習を適用することができます。たとえば、Amazon Machine Learning を使用して電子メールがスパムかどうかを予測するには、スパムかどうかを正しく分類した電子メールの例を収集する必要があります。その後、機械学習を使用し、これらの電子メールの例から一般化して、新しい電子メールがスパムである可能性を予測することができます。実際の回答でラベル付けされたデータからのこの学習アプローチは、教師あり機械学習と呼ばれています。

バイナリ分類 (2 つの可能な結果のうちの 1 つを予測する)、複数クラスの分類 (2 つを超える結果のうちの 1 つを予測する)、および回帰 (数値を予測する) のような、特定の機械学習タスクに対して監視 ML アプローチを使用できます。

バイナリ分類問題の例。

- 顧客はこの製品を購入するでしょうか、それともしないでしょうか。
- このメールはスパムでしょうか。
- この商品は本でしょうか、それとも家畜でしょうか。
- このレビューは顧客によって書かれたものでしょうか、それともロボットによって書かれたものでしょうか。

複数クラスの分類問題の例。

- この製品は書籍、映画、衣類のいずれですか。
- この映画はロマンチックコメディ、ドキュメンタリー、またはスリラーですか。
- この顧客にとって最も関心のある商品のカテゴリはどれですか。

回帰分類問題の例。

- 明日のシアトルの温度はどうなりますか。
- この製品の販売台数は何台ですか。
- この顧客はアプリケーションをいつまで使用しますか。
- この家はどのような値段で売れるでしょうか。

機械学習をいつ使うべきか

ML はあらゆる種類の問題の解決策ではないことを覚えておくことが重要です。ML テクニックを使用せずに堅牢なソリューションが開発できる場合があります。たとえば、簡単なルール、計算、またはデータ駆動型学習を必要とせずにプログラムできる所定の手順を使用して目標値を決定できる場合は、ML は必要ありません。

以下の状況では、機械学習を使用します。

- ルールをコーディングできない。(電子メールがスパムかどうかを認識するなどの) 人間によるタスクの多くは、単純な (決定的な) ルールベースのソリューションを使用して適切に解決することはできません。多数の要因が答えに影響する可能性があります。ルールがあまりにも多くの要因に依存していたり、ルールの多くが重複しているか、または非常に細かく調整される必要がある場合、人間がルールを正確にコーディングすることはすぐに困難になります。この問題を効果的に解決するために、ML を使用できます。
- スケーリングできない。数百の E メールを手動で認識し、スパムかどうかを判断できる場合があります。ただし、このタスクが、何百万もの E メールとなると手間がかかります。ML ソリューションは、大規模な問題を処理するのに効果的です。

機械学習アプリケーションの構築

ML アプリケーションの構築は、一連のステップを含む反復プロセスです。ML アプリケーションを構築するには、以下の一般的な手順を実行します。

1. 観測されることとモデルに予測させたい答えの観点から、主要な ML の問題を構成します。
2. ML モデルトレーニングアルゴリズムで使用できるようにデータを収集し、クリーンにし、準備します。データを可視化して分析し、健全性チェックを実行してデータの品質を検証し、データを理解します。
3. 多くの場合、未加工データ (入力変数) と回答 (目標) は、高度な予測モデルをトレーニングするために使用できる方法では表されません。したがって、通常は、未加工の変数からより予測的な表現または機能を構築しようとする必要があります。
4. その結果得られた機能を学習アルゴリズムに供給してモデルを構築し、モデル構築から取り出されたデータにおけるモデルの品質を評価します。
5. モデルを使用して、新しいデータインスタンスのターゲット回答の予測を生成します。

問題の策定

機械学習の第一歩は、予測するものを決定することです。これは、ラベルまたはターゲット回答と呼ばれます。製品を製造するシナリオを想像してみてください。しかし、各製品を製造する決定は、潜在的な販売数によって異なります。このシナリオでは、各製品が何回購入されるかを予測します (売上数の予想)。機械学習を使用してこの問題を定義する方法は複数あります。問題を定義する方法の選択は、ユースケースやビジネスニーズによって異なります。

顧客が各製品に対して行う購入数を予測しますか (その場合、ターゲットは数値であり、回帰問題を解きます)。または、どの製品が 10 以上購入されるかを予測するのでしょうか (その場合、ターゲットはバイナリで、バイナリ分類の問題を解きます)。

問題を複雑にしすぎないようにし、ニーズに合った最も簡単な解決策を立てることが重要です。しかし、情報、特に過去の回答の情報を失わないことも重要です。ここで、実際の過去の販売数をバイナリ変数「10 以上」と「より少ない」に変換すると、貴重な情報が失われます。どのターゲットを予測するのが最も理にかなっているかを判断するのに時間を費やすことで、こちらの質問に答えないモデルを構築せずにすみます。

ラベル付きデータの収集

ML の問題はデータから始まります。できれば、すでにターゲットの回答が分かっているデータが多くあればよいでしょう。ターゲットの回答がすでに分かっているデータを、ラベル付きデータといいます。監視された ML では、提供されるラベル付きの例から、アルゴリズムが自ら学びます。

データのそれぞれの例や観察には、2 つの要素が含まれています。

- ターゲット – 予測しようとする回答。学習のために ML アルゴリズムにターゲット (正解) でラベル付けされたデータを提供します。次に、トレーニングされた ML モデルを使用して、ターゲット回答がわからないデータに対するこの回答を予測します。
- 変数/機能 – これは、ターゲット回答を予測するパターンを識別するために使用できる例の属性です。

たとえば、E メール分類の問題の場合、ターゲットは E メールがスパムかどうかを示すラベルです。変数には、E メールの送信者、E メールの本文中のテキスト、件名のテキスト、E メールが送信された時刻、および送信者と受信者の間の以前のやり取りの存在などが例としてあります。

多くの場合、データはラベル付けされた形式では容易に入手できません。変数とターゲットを収集して準備することは、多くの場合、ML 問題を解決するための最も重要なステップです。サンプルデータは、予測を行うためにモデルを使用しているときのデータを表す必要があります。たとえば、E メールが迷惑メールかどうかを予測するには、機械学習アルゴリズムが正 (迷惑メール) と否 (迷惑メール以外のメール) の 2 つのタイプの E メールを区別するパターンを見つけるために、両方を収集する必要があります。

ラベル付けされたデータを取得したら、そのデータをアルゴリズムまたはソフトウェアが受け入れ可能な形式に変換する必要があるかもしれません。例えば、Amazon ML を使用するには、データをコンマ区切り (CSV) 形式に変換し、それぞれの例が CSV ファイルの 1 行を構成し、各列は 1 つの入力変数を含み、1 列はターゲット回答を含んでいる必要があります。

データの分析

ラベル付きデータを ML アルゴリズムに送る前に、データを検査して問題を特定し、使用しているデータについての洞察を得ることをお勧めします。モデルの予測する能力は、供給しているデータの質にかかっています。

データを分析するときは、以下の点を考慮する必要があります。

- 変数とターゲットデータの概要 – 変数が取る値と、データの中での主要な値を理解することは役立ちます。解決したい問題について、Subject Matter Expert によってこの要約を実行できます。自問、または Subject Matter Expert へ尋ねてください。データは期待通りのものですか。データ収集に問題があるようですか。ターゲットのあるクラスは、他のクラスより頻繁ですか。思っていたより多くの不足している値や無効なデータがありますか。
- 変数とターゲットの間の相関 – 高い相関は変数とターゲットクラスの間に関係があることを意味するので、各変数とターゲットクラスの間の相関を知ることは役立ちます。通常、相関が高い変数

は予測力が高い変数 (シグナル) であるため、含めることにし、相関の低い変数は関連がない可能性が高いので、除外します。

Amazon ML では、データソースを作成し結果のデータレポートを確認することで、データを分析できます。

機能の処理

データの概要と可視化によりデータを把握した後、変数をさらに意味のあるものにするために、変数をさらに変換することが必要な場合があります。これは機能処理と呼ばれます。たとえば、イベントが発生した日時をキャプチャした変数があるとします。この日時は決して繰り返されることはないので、ターゲットの予測には役立たないでしょう。しかし、この変数を、一日の時間帯、曜日、および月を表す機能に変換すると、これらの変数により、イベントが特定の時間帯、平日、または月に発生する傾向にあるかどうかを知るのに役立ちます。学習のためにより一般化できるデータのポイントを作成するこのような特徴処理により、予測モデルを大幅に改善できます。

その他の一般的な機能処理の例:

- 不足しているデータや無効なデータをより意味のある値に置き換えることができます (たとえば、商品タイプ変数の不足している値が書籍であることが実際に分かっている場合は、商品タイプのすべての不足している値を書籍の値に置き換えることができます)。不足している値を補うために使用される一般的な方法は、不足している値を平均値または中央値で置き換えることです。不足している値を置き換える方法を選択する前に、データを理解することが重要です。
- 1 つの変数と別の変数のデカルト積を形成します。たとえば、人口密度 (都市部、郊外、農村部) と州 (ワシントン州、オレゴン州、カリフォルニア州) といった 2 つの変数がある場合、これらの 2 つの変数のデカルト積によって形成される機能 (ワシントン州都市部、ワシントン州郊外、ワシントン州農村部、オレゴン州都市部、オレゴン州郊外、オレゴン州農村部、カリフォルニア州都市部、カリフォルニア州郊外、カリフォルニア州農村部) に役立つ情報があるかもしれません。
- 数値変数をカテゴリにビニングするなどの非線形変換。多くの場合、数値機能とターゲットの関係は線形ではありません (機能の値は単調に増減しません)。そのような場合、数値機能のさまざまな範囲を表すカテゴリ機能に数値機能を格納すると便利です。各カテゴリ機能 (bin) は、ターゲットとのそれ自身の線形関係を持つものとしてモデル化することができます。たとえば、継続的な数値機能である年齢が、書籍を購入する可能性と直線的に相関していないことがわかったとします。ターゲットとの関係をより正確に把握できるようなカテゴリ機能に分類することができます。数値変数の最適なビン数は、変数の特性とターゲットとの関係に依存します。これは、実験を通じて最もよく決定されます。Amazon ML では、推奨レシピのデータ統計に基づいて数値機能の最適なビン数を示唆しています。推奨レシピの詳細については、開発者ガイドを参照してください。

- ドメイン固有の機能 (たとえば、長さ、幅、高さが別々の変数である場合、これら 3 つの変数の積を新しいボリューム機能として作成できます)。
- 変数固有の機能。テキスト機能、ウェブページの構造をキャプチャする機能、または、文の構造などの一部の変数タイプは、構造とコンテキストを抽出するのに役立つ一般的な処理方法を備えています。たとえば、「the fox jumped over the fence」というテキストから n-grams を形成する方法は、ユニグラム で表すと、the、fox、jumped、over、fence となります。または、バイグラムでは、the fox、fox jumped、jumped over、over the、the fence となります。

より関連性の高い機能を含めると、予測能力を向上させるのに役立ちます。明らかに、「シグナル」のある機能、または予測に影響のある機能をいつも事前に知ることができるとは限りません。したがって、ターゲットラベルに関連する可能性のあるすべての機能を含めておき、モデルトレーニングアルゴリズムが最も強い相関を持つ機能を選択するようにします。Amazon ML では、モデル作成時にレシピで機能処理を指定できます。使用可能な機能処理のリストについては、「開発者ガイド」を参照してください。

データをトレーニングデータと評価データに分割する

ML の基本的な目標は、モデルのトレーニングに使用するデータインスタンスを超えて一般化することです。トレーニングされていないデータに対してモデルがパターンを一般化する品質を評価する必要があります。しかし、将来のインスタンスには未知のターゲット値があり、将来のインスタンスの予測の精度を今確認することはできないため、将来のデータのプロキシとして、すでに回答が分かっているデータの一部を使用する必要があります。トレーニングに使用されたのと同じデータを持つモデルを評価することは有用ではありません。なぜなら、トレーニングデータを一般化するのではなく、トレーニングデータを「覚える」モデルに有利になるからです。

一般的な戦略は、利用可能なすべてのラベル付きデータをトレーニングと評価のサブセットに分割することで、通常、トレーニングの方を 70〜80%、評価の方を 20〜30% とします。ML システムは、トレーニングデータを使用してモデルがパターンを理解するようにし、評価データを使用してトレーニングモデルの予測品質を評価します。ML システムは、さまざまなメトリクスを使用して、評価データセットでの予測を true 値と比較する (グランドトゥルースと呼ばれる) ことによって、予測パフォーマンスを評価します。通常は、ターゲット回答が分からない将来のインスタンスの予測を作成するために、評価サブセットの「最適な」モデルを使用します。

Amazon ML は、モデルのトレーニング用に Amazon ML コンソールを通じて送信されたデータを、トレーニング用に 70%、評価用に 30% に分割します。デフォルトでは、Amazon ML は入力データの最初の 70% をトレーニングデータソースのソースデータに表示されている順序で使用し、評価データソースにデータの残り 30%を使用します。Amazon ML では、最初の 70% を使用

する代わりに、ソースデータの 70% をトレーニング用にランダムに選択し、このランダムなサブセットの残りを評価用に使用することもできます。Amazon ML API を使用してカスタム分割比率を指定し、Amazon ML の外で分割されたトレーニングおよび評価データを提供することができます。Amazon ML には、データを分割する方法もあります。分割する方法の詳細については、「データの分割」を参照してください。

モデルのトレーニング

これで、ML アルゴリズム (つまり、学習アルゴリズム) にトレーニングデータを提供する準備が整いました。アルゴリズムは、変数をターゲットにマッピングするトレーニングデータパターンから学習し、これらの関係をキャプチャするモデルを出力します。ML モデルを使用して、ターゲット回答が分からない新しいデータでターゲットを予測できます。

線形モデル

利用可能な ML モデルは多数あります。Amazon ML は、ML モデルの 1 つのタイプである線形モデルを学習します。線形モデルという用語は、モデルが機能の線形組み合わせとして指定されていることを意味します。トレーニングデータに基づいて、学習プロセスは各機能につき 1 つのウェイトを計算して、ターゲット値を予測または推定することができるモデルを形成します。たとえば、ターゲットが顧客の購入する保険金額で、変数が年齢と所得である場合、単純な線形モデルは次のようになります。

```
Estimated target = 0.2 + 5·age + 0.0003·income
```

学習アルゴリズム

学習アルゴリズムのタスクは、モデルのウェイトを学習することです。ウェイトは、モデルが学習しているパターンがデータの実際の関係を反映している可能性を示します。学習アルゴリズムは、損失関数と最適化技術で構成されています。損失とは、ML モデルによって提供されるターゲットの推定値がターゲットとちょうど等しくない場合に生じるペナルティです。損失関数は、このペナルティを単一の値として定量化します。最適化技術は損失を最小限に抑えることを目指しています。Amazon Machine Learning では、3 つの損失関数を使用し、3 つのタイプの予測の問題に 1 つの関数がそれぞれ対応します。Amazon ML で使用される最適化手法は、オンライン確率的勾配降下法 (SGD) です。SGD はトレーニングデータ上で順次パスを行い、パスごとに、損失を最小限に抑える最適なウェイトに近づけるために、一度に 1 つの例の機能ウェイトを更新します。

Amazon ML は次の学習アルゴリズムを使用します。

- バイナリ分類の場合、Amazon ML はロジスティック回帰 (ロジスティックロス関数 + SGD) を使用します。

- マルチクラス分類の場合、Amazon ML はマルチクラスロジスティック回帰 (多項ロジスティック ロス損失 + SGD) を使用します。
- 回帰の場合、Amazon ML は線形回帰 (二乗損失関数 + SGD) を使用します。

トレーニングパラメータ

Amazon ML 学習アルゴリズムでは、ハイパーパラメータまたはトレーニングパラメータと呼ばれるパラメータを使用して、結果として生じるモデルの品質を制御できます。ハイパーパラメータに応じて、Amazon ML は設定を自動的に選択するか、ハイパーパラメータの静的デフォルトを提供します。デフォルトのハイパーパラメータ設定では一般的に有用なモデルが生成されますが、ハイパーパラメータ値を変更することでモデルの予測パフォーマンスを向上できる場合があります。以降のセクションでは、Amazon ML により作成されるような線形モデルの学習アルゴリズムに関連した一般的なハイパーパラメータについて説明します。

学習レート

学習レートは確率的勾配降下法 (SGD) アルゴリズムで使用される一定の値です。学習レートは、アルゴリズムが最適なウェイトに到達する (収束する) 速度に影響します。SGD アルゴリズムは、検出するすべてのデータ例について線形モデルのウェイトを更新します。これらの更新のサイズは、学習レートによって制御されます。学習レートが大きすぎると、ウェイトが最適解に近づかない可能性があります。値が小さすぎると、アルゴリズムは最適ウェイトに近づくために多くのパスを必要とします。

Amazon ML では、データに基づいて学習レートが自動的に選択されます。

モデルサイズ

多くの入力機能があると、データ内の可能なパターンの数により、大きなモデルになり得ます。大きなモデルは、トレーニング中や予測生成時にモデルを保持するためにより多くの RAM を必要とするなど、実用的意義があります。Amazon ML では、L1 正則化を使用するか、最大サイズを指定してモデルサイズを具体的に制限することによって、モデルサイズを縮小できます。モデルサイズを小さくすると、モデルの予測能力が低下する可能性があることに注意してください。

デフォルトのモデルサイズに関する詳細は、「トレーニングパラメータ: タイプとデフォルト値」を参照してください。正則化の詳細については、「正則化」を参照してください。

パスの数

SGD アルゴリズムは、トレーニングデータを順次通過させます。`Number of passes` パラメータは、アルゴリズムがトレーニングデータに対して行うパスの数を制御します。パス数が多いほどデー

タに適したモデルが得られます (学習レートがあまり大きくない場合) が、パス数の増加に伴ってメリットは失われます。小規模なデータセットの場合、パスの数を大幅に増やすことができ、学習アルゴリズムがデータに効果的により適合するようになります。非常に大きなデータセットの場合は、1 つのパスで十分です。

デフォルトのパスの数に関する詳細は、「トレーニングパラメータ: タイプとデフォルト値」を参照してください。

データシャッフル

Amazon ML では、SGD アルゴリズムがトレーニングデータの行の順序の影響を受けるため、データをシャッフルする必要があります。トレーニングデータをシャッフルすると、より良い ML モデルが得られます。これは、SGD アルゴリズムが、検出する最初の種類のデータには最適でも全範囲のデータには最適でないソリューションを避けるのに役立つためです。シャッフルは、データの順序をミックスして、SGD アルゴリズムがあまりに多くの連続した観測で 1 つのタイプのデータに遭遇することがないようにします。連続した多数のウェイトの更新で 1 つのタイプのデータのみを見る場合、更新が大きすぎて、アルゴリズムは新しいデータ型のモデルウェイトを修正できないことがあります。さらに、データがランダムに提示されない場合、アルゴリズムがすべてのデータタイプの最適なソリューションを迅速に見つけることは困難です。場合によってはアルゴリズムが最適なソリューションをどうしても見つけられないことがあります。トレーニングデータをシャッフルすることで、アルゴリズムが最適なソリューションにすぐにたどり着くのに役立ちます。

たとえば、ML モデルをトレーニングして製品タイプを予測しようとしていて、トレーニングデータには、映画、玩具、ビデオゲームの製品タイプが含まれているとします。Amazon S3 にデータをアップロードする前に製品タイプの列でデータをソートすると、アルゴリズムはデータを製品タイプ別にアルファベット順に見ていきます。アルゴリズムは、映画のすべてのデータを最初に見ていき、ML モデルは映画のパターンを学習し始めます。次に、モデルが玩具のデータに遭遇したとき、アルゴリズムが行うすべての更新は、その更新が映画に適したパターンを劣化させるとしても、モデルを玩具の製品タイプに適合させようとします。この映画から玩具のタイプへの突然の切り替えにより、製品タイプについての精度の高い予測を学習できないモデルが生成されます。

デフォルトのシャッフルタイプに関する詳細は、「トレーニングパラメータ: タイプとデフォルト値」を参照してください。

正則化

正則化は、線形モデルが極端なウェイトの値を課すことによって、トレーニングデータ例をオーバーフィットする (つまり、一般化する代わりにパターンを記憶する) のを防ぐのに役立ちます。L1 正則化は、さもなければ小さなウェイトを持つ機能のウェイトを 0 にすることによって、モデルで使

用される機能の数を減らす効果があります。結果として、L1 正則化はモデルをまばらにし、モデル内のノイズの量を低減します。L2 正則化により、全体のウェイトの値が小さくなり、入力機能間の相関性が高い場合にウェイトを安定させます。Regularization type および Regularization amount パラメータを使用して、L1 または L2 正則化の適用を調整します。非常に大きな正則化値では、すべての機能のウェイトがゼロとなり、モデルがパターンを学習できなくなります。

デフォルトの正則化の値に関する詳細は、「トレーニングパラメータ: タイプとデフォルト値」を参照してください。

モデル精度の評価

ML モデルの目標は、トレーニング中に表示されたデータを記憶するのではなく、見えないデータを一般化するパターンを学習することです。モデルを作成したら、モデルのトレーニングに使用していない見えない例でもモデルのパフォーマンスが良好かどうかを確認することが重要です。これを行うには、モデルを使用して評価データセット (保持データ) の回答を予測し、予測されたターゲットを実際の回答 (グランドトゥルース) と比較します。

モデルの予測精度を測定するのに、ML では多数のメトリクスが使用されています。精度メトリクスの選択は ML タスクによって異なります。これらのメトリクスを確認して、モデルのパフォーマンスを判断することが重要です。

バイナリの分類

多くのバイナリ分類アルゴリズムの実際の出力は予測スコアです。スコアは、指定された観測が正のクラスに属しているというシステムの確実性を示します。このスコアの利用者として、観察を正または負に分類するかどうかを決定するために、分類しきい値 (カットオフ) を選択してスコアと比較することにより、スコアを解釈します。スコアがしきい値より大きい観測は正のクラスと予測され、スコアがしきい値より小さい場合は、負のクラスとして予測されます。

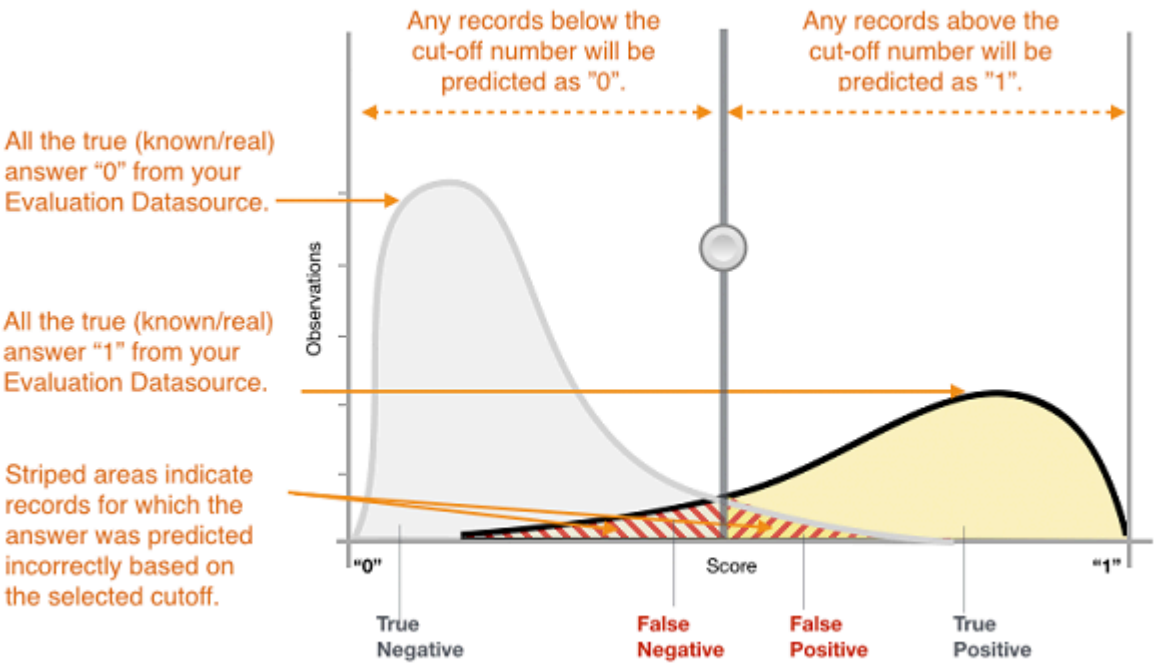


図 1: バイナリ分類モデルのスコア分布

実際の既知の回答と予測回答に基づいて、予測は以下の 4 つのグループに分類されます: 正しい正の予測 (正しい検出)、正しい負の予測 (正しい非検出)、誤った正の予測 (誤検出) と誤った負の予測 (検出漏れ)。

バイナリ分類精度メトリクスは、2 種類の正しい予測と 2 種類のエラーを定量化します。典型的なメトリクスは、精度 (ACC)、正確さ (precision)、リコール、誤検出率、F1 測定値です。各メトリクスは、予測モデルの異なる面を測定します。精度 (ACC) は正しい予測の割合を測定します。正確さ (Precision) は、正と予測されるこれらの例の中で実際の正の割合を測定します。リコールは、実際の正の割合のうち正と予測されたものの数を測定します。F1 測定値は、正確さとリコールを組み合わせた手法です。

AUC は、別のタイプのメトリクスです。モデルの能力を測定して、正の例についてより高いスコアを予測し負の例と比較します。AUC は選択したしきい値から独立しているため、しきい値を選択せずに AUC メトリクスからモデルの予測パフォーマンスを知ることができます。

ビジネス上の問題によっては、これらのメトリクスの特定のサブセットでうまくいくモデルにもっと興味があるかもしれません。たとえば、2 つのビジネスアプリケーションで、ML モデルの要件が非常に異なる場合があります。

- 一方のアプリケーションでは、正の予測が実際に正 (高い正確性) であると確認し、いくつかの正の例を負 (中程度のリコール) として誤分類する可能性があります。
- 別のアプリケーションでは、可能な限り多くの正の例を正しく予測する必要があるかもしれないため (高いリコール)、正として間違って分類されるいくつかの負の例を受け入れます (中程度の正確性)。

Amazon ML では、観測により、予測された [0,1] の範囲のスコアを取得します。例を 0 または 1 として分類する決定をするためのスコアしきい値は、デフォルトで 0.5 に設定されています。Amazon ML により、異なるスコアしきい値を選択することによる影響を確認でき、ビジネスニーズに合った適切なしきい値を選択できます。

複数クラスの分類

バイナリ分類問題のプロセスとは異なり、予測を行うのにスコアしきい値を選択する必要はありません。予測される回答は、予測スコアが最も高いクラス (つまり、ラベル) です。時には、スコアが高いと予測される場合にのみ予測回答を使用することもできます。この場合は、予測された回答を受け入れるかどうかを決定するための予測スコアのしきい値を選択することができます。

複数クラスで使用される一般的なメトリクスは、バイナリ分類のケースで使用されるメトリクスと同じです。他のすべてのクラスを第 2 クラスに属するとしてグループ化した後、メトリクスをバイナリ分類問題として扱うことによって、メトリクスはクラスごとに計算されます。次に、マクロ平均 (各クラスを同様に扱う)、または、加重平均 (クラスの頻度により重みづけされる) のいずれかを得るために、バイナリメトリクスはすべてのクラスで平均されます。Amazon ML では、マクロ平均 F1 測定値を使用して、複数クラス分類子の予測の成功を評価します。

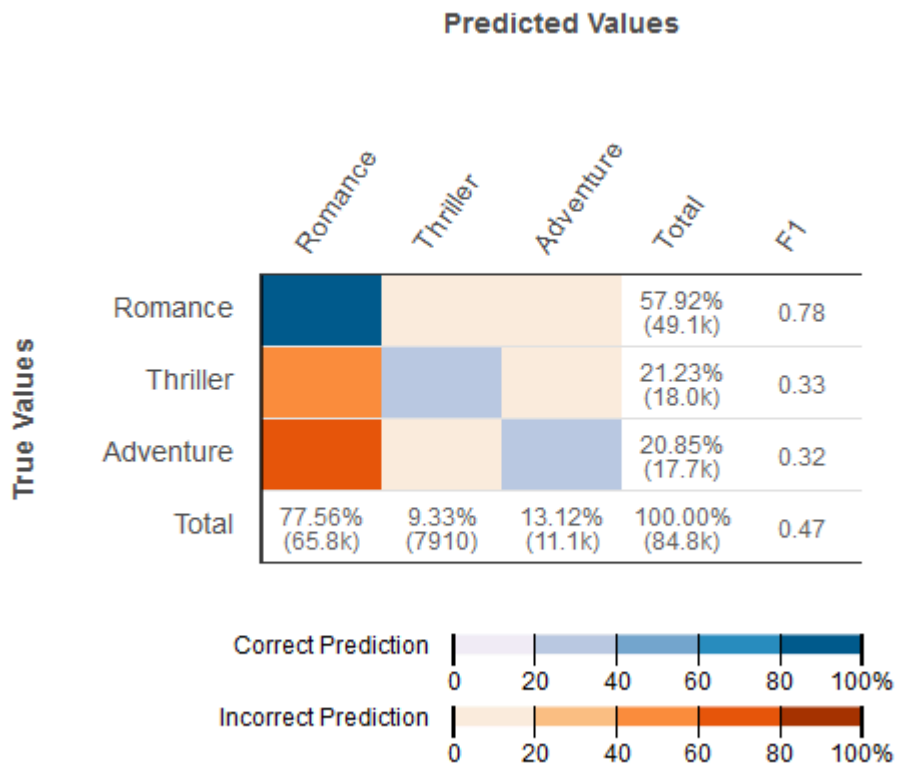


図 2: 複数クラス分類モデルの混同行列

複数クラス問題で混合行列を確認することは有用です。混同行列とは、評価データの各クラスと、正しい予測と誤った予測の数または割合を示す表です。

回帰

回帰タスクの場合、一般的な精度メトリクスは、二乗平均平方根誤差 (RMSE) および平均絶対誤差率 (MAPE) です。これらのメトリクスでは、予測された数値ターゲットと実際の数値解の間の距離を測定します (グランドトゥルース)。Amazon ML では、回帰モデルの予測の正確性を評価するために RMSE メトリクスが使用されます。

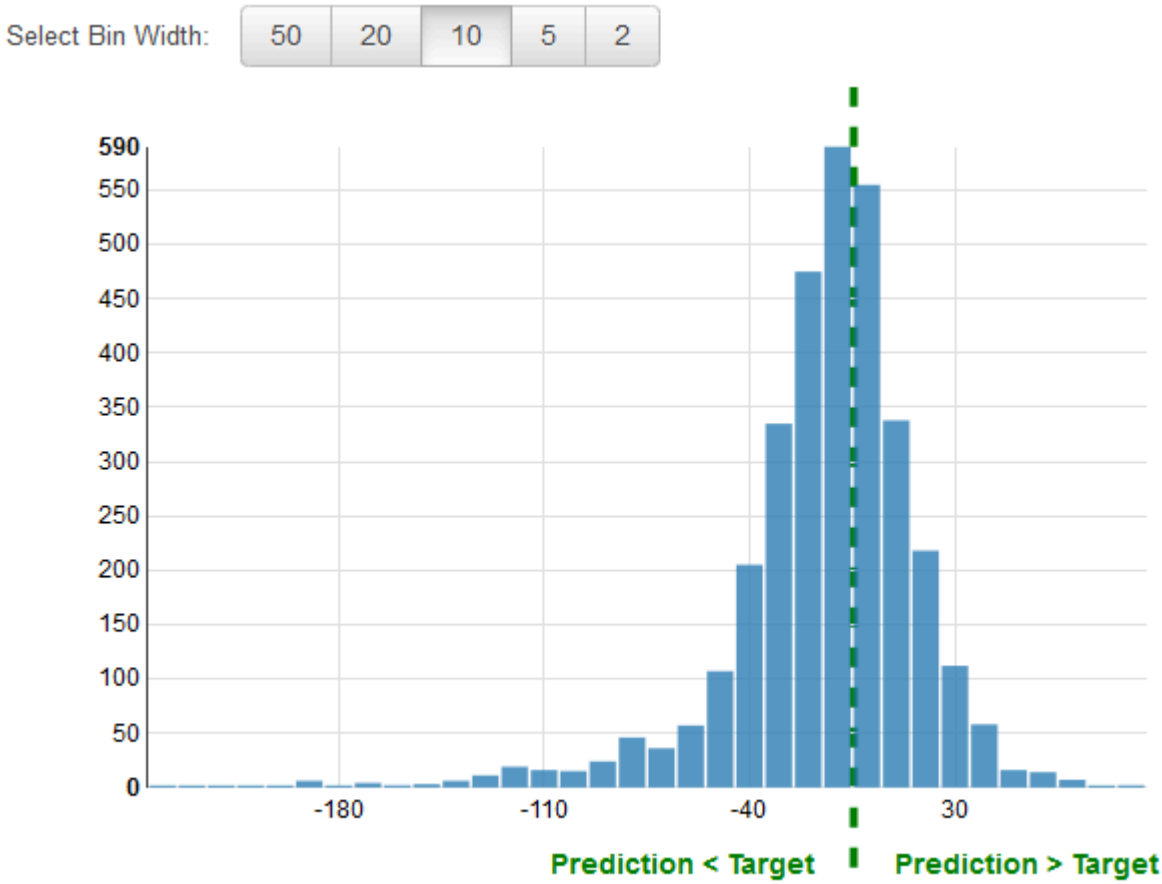


図 3: 回帰モデルの残余の分布

回帰問題では残余をレビューするのが一般的な方法です。評価データの観測の残余とは、真のターゲットと予測されたターゲットの違いを意味しています。残余は、モデルが予測できないターゲットの部分を表しています。正の残余は、モデルがターゲットを過少評価している (実際のターゲットが予測ターゲットより大きい) ことを示します。負の残余は、モデルがターゲットを過大評価している (実際のターゲットが予測ターゲットより小さい) ことを示します。評価データの残余のヒストグラムが、ゼロを中心とするベル形状で分布している場合、モデルがランダムにミスを犯していて、ターゲット値の特定の範囲で体系的に過大予測または過小予測していないことを示します。残余がゼロを中心としたベル形状にならない場合、モデルの予測エラーに何かの構造が存在しています。モデルに変数を追加すると、現在のモデルでキャプチャしていないパターンをモデルがキャプチャする役に立つかもしれません。

モデル精度の向上

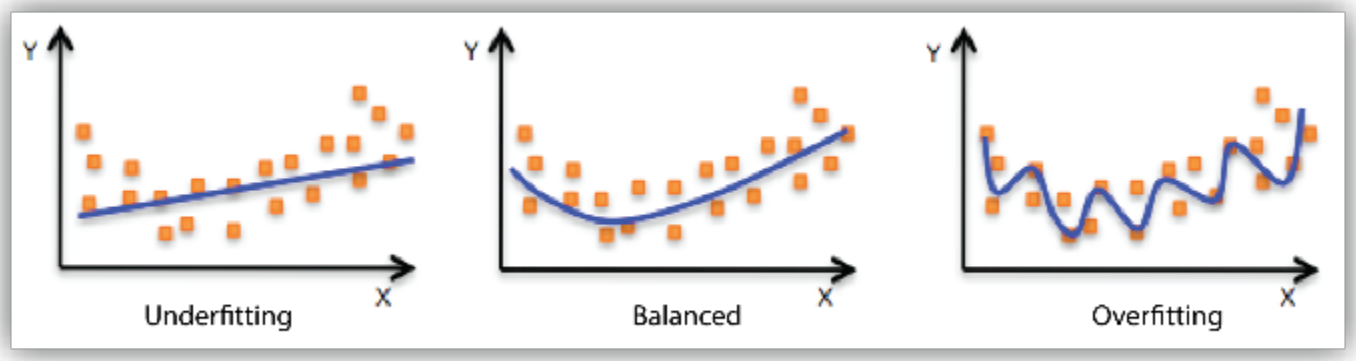
ニーズに合った ML モデルを得るため、通常は、この ML プロセスを繰り返し実行し、いくつかのバリエーションを試します。最初の反復で非常に予測的なモデルを得ることはできないかもしれません

し、モデルを改善してさらに良い予測を得ようとするかもしれません。パフォーマンスを向上させるために、これらのステップを繰り返すことができます。

1. データ収集: トレーニング例の数を増やす
2. 機能処理: 変数とより良い処理機能を追加する
3. モデルパラメータのチューニング: 学習アルゴリズムで使用するトレーニングパラメータの値を変えてみる

モデルフィット: アンダーフィットとオーバーフィット

モデルの適合性を理解することは、モデルの精度が悪いという問題の根本的な原因を理解する上で重要です。これを理解することで、修正のための手段を講じることができます。トレーニングデータと評価データの予測エラーを見て、予測モデルがトレーニングデータに対してアンダーフィットかオーバーフィットかを判断できます。



トレーニングデータでのモデルのパフォーマンスが悪いときは、モデルがトレーニングデータにアンダーフィットしています。これは、モデルが入力例 (X と呼ばれることが多い) とターゲット値 (Y と呼ばれることが多い) の関係をキャプチャできないことが原因です。トレーニングデータでのモデルのパフォーマンスがよくても、評価データでのパフォーマンスはよくないことが観察される場合、モデルがトレーニングデータにオーバーフィットしています。これは、モデルが見たデータを記憶していて、見ていない例に対して一般化できないことが原因です。

トレーニングデータでのパフォーマンスが悪い原因は、モデルがターゲットを説明するには単純すぎる (入力機能が十分に説明されていない) ことである可能性があります。モデルの柔軟性を高めることで、パフォーマンスを改善できます。モデルの柔軟性を高めるには、次の操作を試してください。

- 新しいドメイン固有の機能および機能のデカルト製品を追加する、また、機能処理のタイプを変更する (n グラムサイズの増加など)

- 使用する正則化の量を減らす

モデルがトレーニングデータにオーバーフィットしている場合は、モデルの柔軟性を低下させる措置を取るのが適切です。モデルの柔軟性を低下させるには、次の操作を試してください。

- 機能の選択: 機能の組み合わせを少なくする、n グラムのサイズを小さくする、および、数値属性ビンの数を減らすことを検討します。
- 使用する正則化の量を増やす。

学習アルゴリズムに学習するのに十分なデータがないため、トレーニングデータとテストデータの精度が悪くなることがあります。次の手順を実行してパフォーマンスを向上させることができます。

- トレーニングデータの例の量を増やす。
- 既存のトレーニングデータのパスの数を増やす。

モデルを使用した予測の作成

これで、ML モデルのパフォーマンスがよくなり、予測の作成に使用できます。Amazon Machine Learning では、モデルを使用した予測を作成する 2 つの方法があります。

バッチ予測

バッチ予測は、一連の観測の予測を一度に生成してから、一定の割合または数の観測に対してアクションを実行する場合に便利です。通常、そのようなアプリケーションには、低レイテンシーの要件がありません。たとえば、ある製品の広告キャンペーンの一部としてターゲットにする顧客を決定する場合は、すべての顧客の予測スコアを取得し、モデルの予測をソートして、どの顧客が最も購入する可能性が高いかを識別し、購入する可能性が最も高いおそらく上位 5% の顧客をターゲットにします。

オンライン予測

オンライン予測シナリオは、レイテンシーの低い環境で、他のサンプルとは独立してサンプルごとに 1 つずつ予測を生成する場合に使用します。たとえば、予測を使用して、特定のトランザクションが不正なトランザクションである可能性が高いかどうかを即座に判断できます。

新しいデータでのモデルの再トレーニング

モデルが精度の高い予測をするためには、予測の基になっているデータが、モデルがトレーニングされたデータと同様の分布を持っている必要があります。データの分布は時間の経過とともに変化することが予想されるため、モデルのデプロイは、1 回限り実行されるのではなく、むしろ連続的な処理です。データの分布が元のトレーニングデータの分布から大幅に逸脱していることがわかったら、受信データを継続的に監視し、より新しいデータでモデルを再トレーニングすることをお勧めします。データ配信における変化を検出するための監視はオーバヘッドが大きいのであれば、より簡単な戦略は、たとえば、毎日、毎週、または毎月など、定期的にモデルをトレーニングすることです。Amazon ML でモデルを再トレーニングするには、新しいトレーニングデータに基づいて新しいモデルを作成する必要があります。

Amazon Machine Learning プロセス

次の表は、このドキュメントで説明されている ML プロセスを、Amazon ML コンソールを使用して実行する方法を説明します。

ML プロセス	Amazon ML タスク
データの分析	Amazon ML でデータを分析するには、データソースを作成し、データインサイトページを確認してください。
トレーニングおよび評価データソースにデータを分割	Amazon ML はデータソースを分割して、データの 70% をモデル トレーニング、30% をモデルの予測パフォーマンスの評価に使用できます。 ML モデルの作成ウィザードをデフォルトの設定で使用すると、Amazon ML はデータを分割します。 ML モデルの作成ウィザードをカスタム設定で使用し、ML モデルの評価が選択されている場合、Amazon ML がデータを分割するのを許可するためのオプションが表示され、データの 30% で評価を実行します。
トレーニングデータのシャッフル	ML モデルの作成ウィザードをデフォルトの設定で使用すると、Amazon ML はデータをシャッフルします。また、データを Amazon ML にインポートする前にシャッフルすることもできます。
プロセス機能	トレーニングデータを学習と一般化のための最適な形式にまとめるプロセスは、機能変換と呼ばれます。ML モデルの作成ウィザードをデフォ

ML プロセス	Amazon ML タスク
	ルトの設定で使用すると、Amazon ML はデータの機能処理設定の候補を表示します。 機能処理設定を指定するには、ML モデルの作成ウィザードの [カスタム] オプションを使用して機能処理レシピを提供します。
モデルのトレーニング	ML モデルの作成ウィザードを使用して Amazon ML でモデルを作成する場合、Amazon ML がモデルのトレーニングを行います。
モデルパラメータの選択	Amazon ML では、モデルの予測パフォーマンスに影響する 4 つのパラメータを調整できます。それらは、モデルサイズ、合格の数、シャッフルのタイプ、および正規化です。ML モデルの作成ウィザードを使用して ML モデルを作成する場合、[カスタム] オプションを選択するとこれらのパラメータを設定できます。
モデルパフォーマンスの評価	評価の作成ウィザードを使用して、モデルの予測パフォーマンスを評価します。
機能の選択	Amazon ML の学習アルゴリズムでは、学習プロセスにあまり影響しない機能を削除できます。これらの機能を削除することを示すには、ML モデルの作成時に [`L1 regularization `] パラメータを選択します。
予測精度のスコアしきい値の設定	異なるスコアしきい値でモデルの評価レポートの予測パフォーマンスを確認し、それからビジネスアプリケーションに基づいてスコアしきい値を設定します。スコアしきい値は、モデルがマッチ予測を定義する方法を決定します。誤検出および検出漏れを制御する数を調整します。
モデルの使用	バッチ予測の作成ウィザードを使用して、モデルによる観測バッチの予測を取得します。 または、`Predict` API を使用して ML モデルのリアルタイム予測を有効にし、オンデマンドで個々の観測の予測を取得します。

Amazon Machine Learning の設定

Amazon Machine Learning を初めて使用するには、AWS アカウントが必要になります。アカウントをお持ちでない場合は、「AWS へのサインアップ」を参照してください。

AWS にサインアップする

アマゾン ウェブ サービス (AWS) にサインアップすると、Amazon ML を含む AWS のすべてのサービスに対して 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。すでに AWS アカウントをお持ちの場合は、この手順をスキップしてください。AWS アカウントをお持ちでない場合は、次に説明する手順に従ってアカウントを作成してください。

サインアップして AWS アカウントを作成するには

1. http://aws.amazon.com にアクセスし、[サインアップ] を選択します。
2. 画面上の指示に従ってください。

サインアップ手順の一環として、通話呼び出しを受け取り、電話のキーパッドを用いて PIN を入力することが求められます。

チュートリアル: Amazon ML を使用してマーケティングオファーへの応答を予測する

Amazon Machine Learning (Amazon ML) で、予測モデルを構築してトレーニングし、スケーラブルクラウドソリューションにアプリケーションをホストすることができます。このチュートリアルでは、Amazon ML コンソールを使用してデータソースを作成する方法、機械学習 (ML) モデルを構築する方法、およびアプリケーションで使用できる予測を生成するモデルを使用する方法を説明します。

このサンプル演習では、ターゲットを絞ったマーケティングキャンペーンの潜在的なお客様を識別する方法を示していますが、さまざまな ML モデルを作成して使用するのに同じ原則が適用できます。サンプル演習を完了するには、カリフォルニア大学アーバイン校 (UCI) Machine Learning Repository にある一般に利用可能な銀行およびマーケティングデータセットを使用します。これらのデータセットには、顧客に関する一般情報と、顧客が以前のマーケティング活動にどのように応答したかに関する情報が含まれています。このデータを使用して、譲渡性預金証書 (CD) としても知られる新製品の定期預金を購入する可能性が最も高いと思われるお客様を識別します。

> **Warning**
>
> このチュートリアルは、AWS 無料利用枠に含まれていません。Amazon ML の料金の詳細については、「Amazon Machine Learning の料金」を参照してください。

前提条件

チュートリアルを実行するには、AWS アカウントが必要です。AWS アカウントをまだお持ちでない場合は、「Amazon Machine Learning のセットアップ」を参照してください。

ステップ

- ステップ 1: データを準備する
- ステップ 2: トレーニングデータソースを作成する
- ステップ 3: ML モデルの作成
- ステップ 4: ML モデルの予測パフォーマンスを確認し、スコアのしきい値を設定する
- ステップ 5: ML モデルを使用して予測を生成する

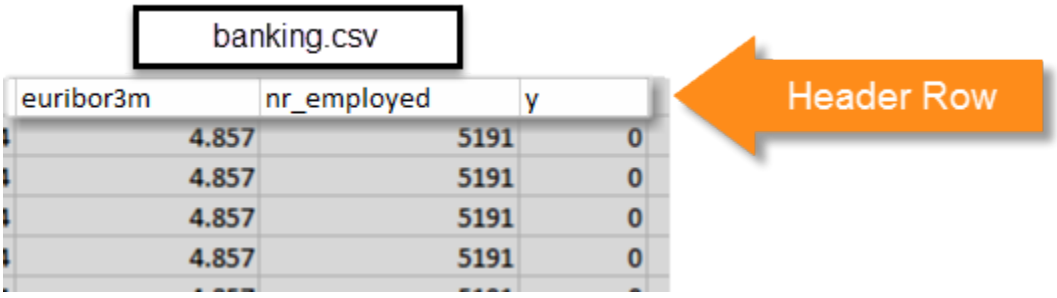
ステップ 1: データを準備する

機械学習では、通常はデータを取得し、トレーニングを開始する前にそれが正しくフォーマットされていることを確認します。このチュートリアルのために、サンプルデータセットを UCI Machine Learning リポジトリから取得し、Amazon ML ガイドラインに準拠するようフォーマットし、ダウンロードできるようにしました。このトピックの手順に従って、データセットを Amazon Simple Storage Service (Amazon S3) ストレージの場所からダウンロードし、自分の S3 バケットにアップロードしてください。

Amazon ML フォーマット要件については、「Amazon ML のデータ形式について」を参照してください。

データセットをダウンロードするには

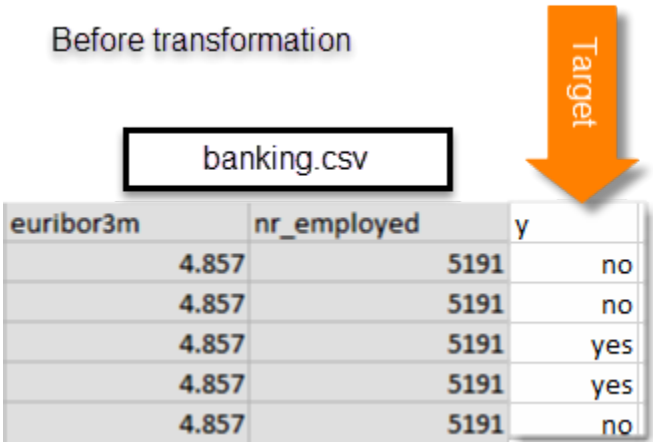
1. banking.zip をクリックして、あなたの銀行の定期預金に似ている製品を購入したお客様の履歴データが保存されているファイルをダウンロードします。フォルダーを解凍し、banking.csv ファイルをコンピュータに保存します。
2. banking-batch.zip をクリックして、可能性のある顧客が提供に反応するかどうかの予測に使用するファイルをダウンロードします。フォルダーを解凍し、banking-batch.csv ファイルをコンピュータに保存します。
3. banking.csv を開きます。データの行と列が表示されます。ヘッダー行 には、各列の属性名が含まれています。属性 は一意の指名プロパティで、各カスタマーの特定の特性を記述するもので、たとえば nr_employed ならカスタマーの雇用状態を表します。各行は、単一のカスタマーに関する観測のコレクションを表します。

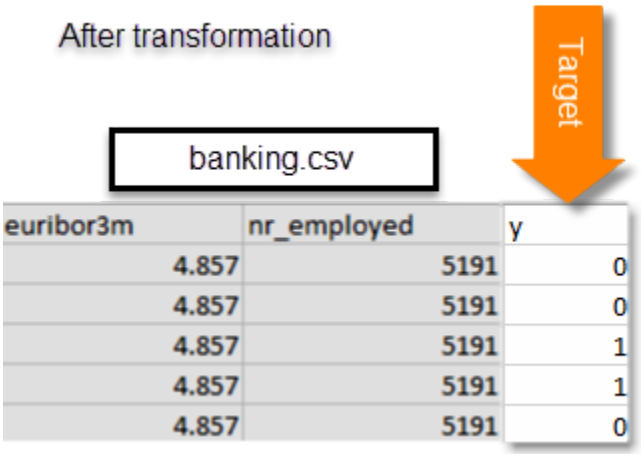


ML モデルに、「このカスタマーは新しい製品をサブスクライブしますか」という質問に答えてもらうとします。この質問への答えは banking.csv データセットの [y] 属性値で、値は 1 (はい) または 0 (いいえ) です。Amazon ML に予測方法を学習してもらいたい属性を、ターゲット属性 と呼びます。

> **Note**
>
> 属性 [y] はバイナリ属性です。2 つの値のいずれか 1 つのみを含めることができ、この場合は 0 または 1 です。元の UCI データセットでは、y 属性は、[Yes] または [No] です。元のデータセットは編集されています。[y] 属性の yes を意味するすべての値が 1 に、no を意味するすべての値が 0 になっています。独自のデータを使用する場合は、バイナリ属性に他の値を使用することができます。有効な値の詳細については、「AttributeType フィールドの使用」を参照してください。

以下の例は、[y] 属性の値をバイナリ属性 0 および 1 に変更する前後のデータを示しています。





banking-batch.csv ファイルに [y] 属性は含まれていません。ML モデルを作成した後で、そのモデルを使用してファイル内の各レコードの [y] を予測します。

次に、banking.csv および banking-batch.csv ファイルを Amazon S3 にアップロードします。

Amazon S3 の場所にファイルをアップロードするには

1. AWS Management Console にサインインし、Amazon S3 コンソール https://console.aws.amazon.com/s3/ を開きます。
2. [すべてのバケット] リストで、バケットを作成するか、ファイルをアップロードする場所を選択します。
3. ナビゲーションバーで、[アップロード] を選択します。
4. [Add Files] を選択します。
5. ダイアログボックスでデスクトップに移動してから `banking.csv` および `banking-batch.csv` を選択し、[オープン] を選択します。

これで、トレーニングデータソースを作成する 準備ができました。

ステップ 2: トレーニングデータソースを作成する

`banking.csv` データセットを Amazon Simple Storage Service (Amazon S3) の場所にアップロードした後で、それを使用してトレーニングデータソースを作成します。データソースは、入力データの場所と入力データに関する重要なメタデータが保存されている Amazon Machine Learning (Amazon ML) オブジェクトです。Amazon ML は、ML モデルのトレーニングや評価などの操作でデータソースを使用します。

データソースを作成するには、以下を指定します。

- データの Amazon S3 の場所とデータへのアクセス許可
- データ内の属性の名前および各属性の型 (数値、文字、カテゴリ、またはバイナリ) を含むスキーマ
- Amazon ML が予測を学習して応答する属性の名前、つまりターゲット属性

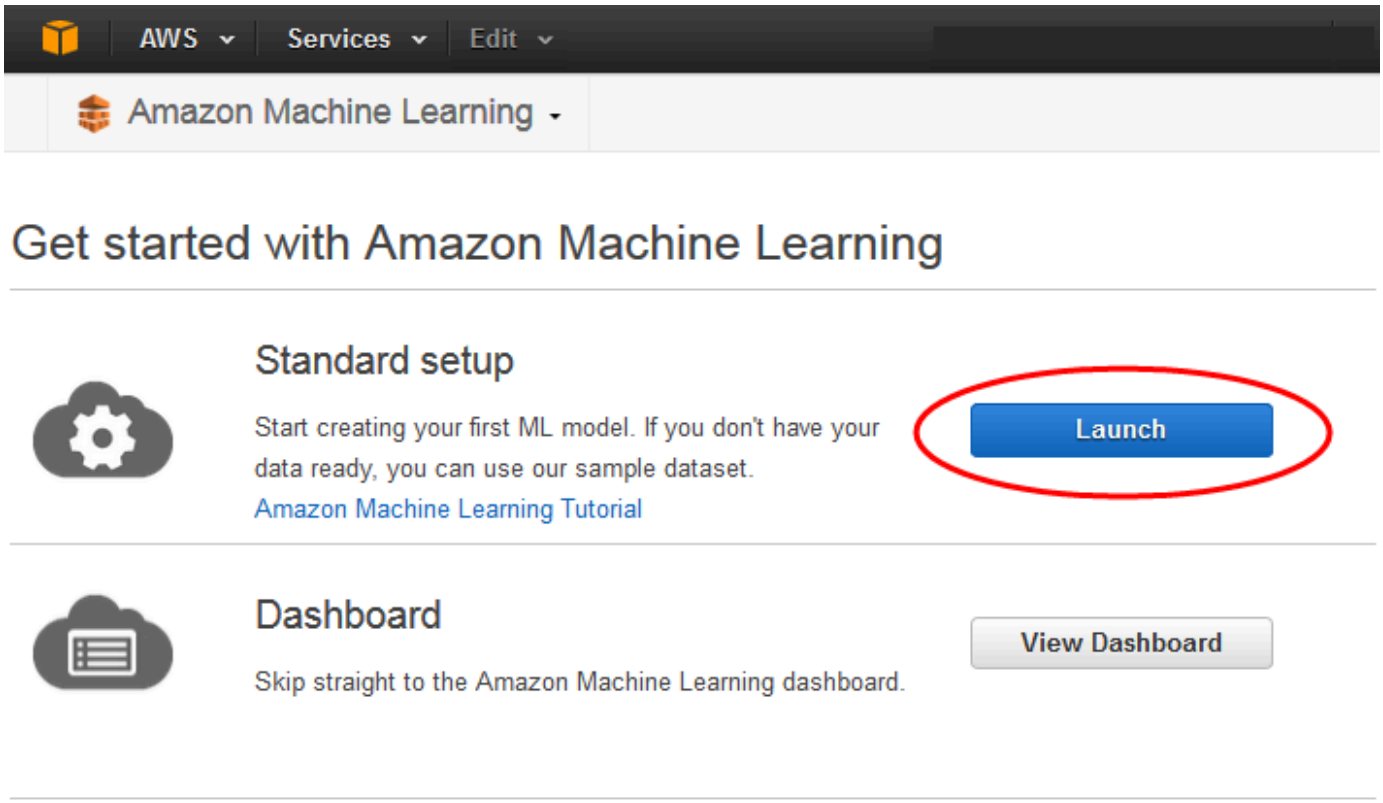
> **Note**
>
> データソースは参照のみで、実際にはデータを保存しません。Amazon S3 に保存されているファイルを移動または変更しないようにします。移動または変更した場合、Amazon ML は ML モデルの作成、評価の生成、または予測の生成のためにそれらにアクセスできなくなります。

トレーニングデータソースを作成するには

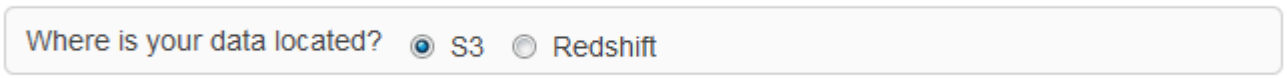
1. Amazon Machine Learning コンソール (https://console.aws.amazon.com/machinelearning/) を開きます。
2. [開始する] を選択します。

> **Note**
>
> このチュートリアルでは、Amazon ML を初めて使用することを前提としています。Amazon ML を使用したことがある場合は、Amazon ML ダッシュボードの [Create new...] (新規作成...) ドロップダウンリストを使用して、データソースを新規に作成します。

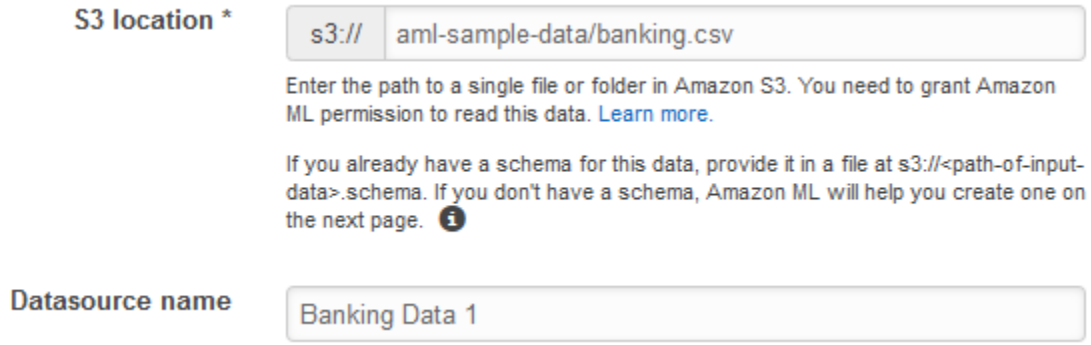
3. [Get started with Amazon Machine Learning] (Amazon Machine Learning の使用開始) ページで、[Launch] (起動) を選択します。



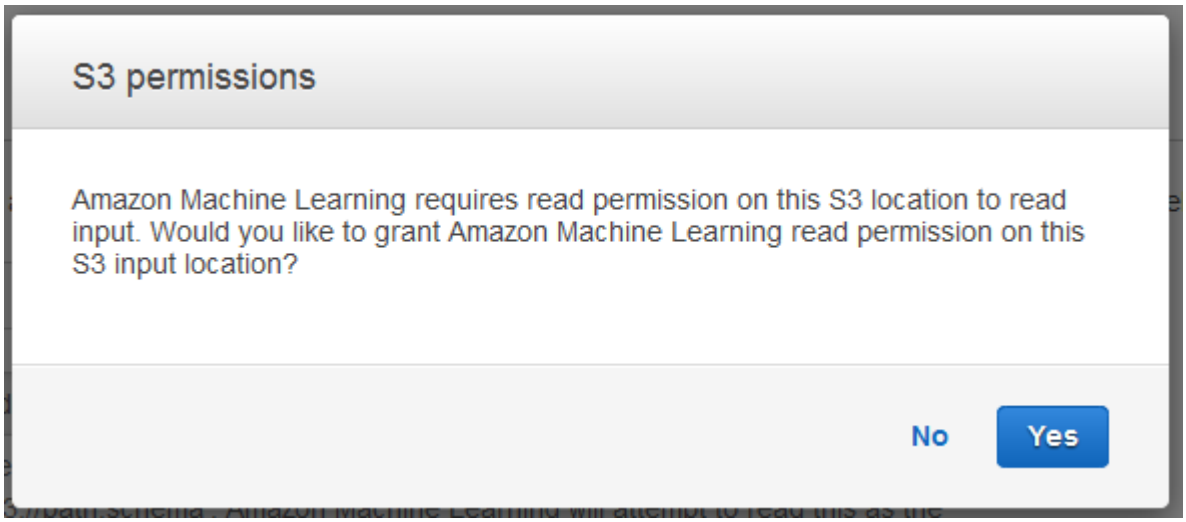
4. [入力データ] ページの、[データの場所] で [S3] が選択されていることを確認します。



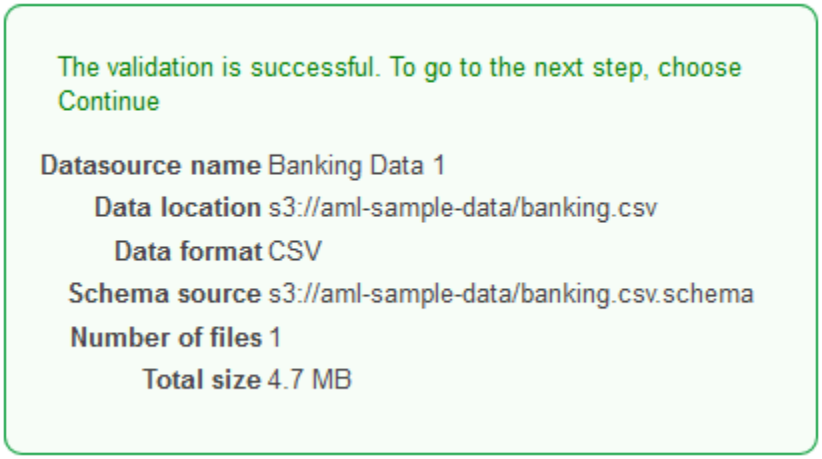
5. [S3 の場所] には、「ステップ 1: データを準備する」で作成した `banking.csv` ファイルの完全な場所を入力します。例えば、*`your-bucket`***`/banking.csv`** などです。Amazon ML がバケット名に s3:// を付加します。
6. [データソース名] に「**`Banking Data 1`**」を入力します。



7. [検証] を選択します。
8. [S3 アクセス権限] ダイアログボックスで、[はい] を選択します。



9. Amazon ML が S3 の場所でデータファイルにアクセスでき読み取れた場合は、次のようなページが表示されます。プロパティを確認し、[続行] を選択します。



次に、スキーマを確立します。スキーマとは、Amazon ML が入力データを ML モデル用に解釈するために必要な情報のことです。それには、属性の名前および割り当てられたデータタイプ、特殊な属性の名前などがあります。Amazon ML にスキーマに渡すには、2 つの方法があります。

- Amazon S3 データをアップロードするときに、別のスキーマファイルを提供します。
- Amazon ML に属性タイプの推測とスキーマの作成を許可します。

このチュートリアルでは、Amazon ML にスキーマを推測させます。

別のスキーマファイルの作成の詳細については、「Amazon ML のデータスキーマを作成する」を参照してください。

Amazon ML にスキーマを推測させるには

1. [Schema] (スキーマ) ページに、Amazon ML が推測したスキーマが表示されます。Amazon ML が推測した属性のデータ型を確認します。Amazon ML がデータを正しく取り込み、属性に正しい機能処理が行われるために重要な点は、属性に正しいデータ型が割り当てられていることです。

   - たとえば yes または no など 2 つの状態のみを持つ属性は、[バイナリ] とマークされている必要があります。
   - カテゴリを示すために数値または文字列が使用される属性は、[カテゴリ] とマークされている必要があります。
   - 順序に意味を持つ数値が使用される属性は、[数値] としてマークされている必要があります。

- スペースで区切られた単語からなる文字列が使用される属性は、[テキスト] としてマークされている必要があります。

Name	Data Type	Sample Field Value 1
age	Numeric	56
campaign	Numeric	1
cons_conf_idx	Numeric	-36.4
cons_price_idx	Numeric	93.994
contact	Categorical	telephone
day_of_week	Categorical	mon
default	Categorical	no
duration	Numeric	261
education	Categorical	basic.4y
emp_var_rate	Numeric	1.1

2. このチュートリアルでは、Amazon ML がすべての属性のデータ型を正しく識別しているため、[Continue] (続行) を選択します。

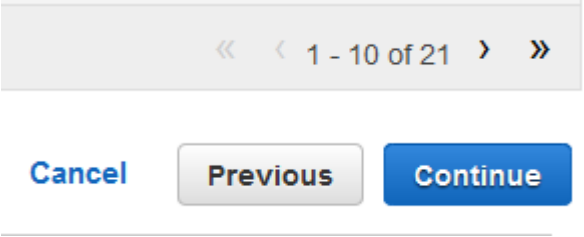
次に、ターゲット属性を選択します。

ターゲットは、ML モデルが予測を学習する必要がある属性であることに注意してください。属性 [y] は、個々の顧客が過去にキャンペーンに加入しているかどうかを、1 (はい) または 0 (いいえ) で示します。

> **Note**
>
> データソースを ML モデルのトレーニングおよび評価に使用する場合にのみ、ターゲット属性を選択してください。

[y] 属性をターゲット属性として選択するには

1. テーブルの右下で、単一の矢印を選択してテーブルの最後のページに進みます。ここで、[y] 属性が表示されます。



2. [ターゲット] 列で [y] を選択します。

   Amazon ML は、ターゲットとして [y] が選択されていることを確認します。

3. [Continue] (続行) をクリックします。
4. [行 ID] ページの、[データには識別子が含まれていますか?] で、デフォルトの [No] が選択されていることを確認します。
5. [レビュー] を選択し、[続行] を選択します。

トレーニングデータソースを作成したため、モデルの作成 の準備ができました。

ステップ 3: ML モデルの作成

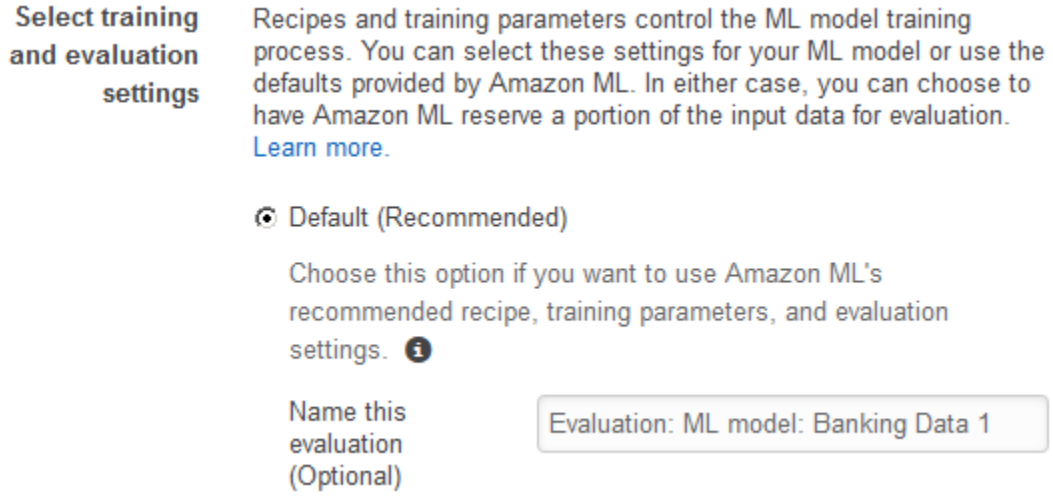
トレーニングデータソースを作成した後、それを使用して ML モデルを作成し、モデルをトレーニングして、結果を評価します。ML モデルは、Amazon ML がトレーニング中にデータで見つけるパターンの集まりです。モデルを使用して予測を作成します。

ML モデルを作成するには

1. 使用開始ウィザードはトレーニングデータソースとモデルの両方を作成するため、Amazon Machine Learning (Amazon ML) は作成したトレーニングデータソースを自動的に使用し、[ML model settings] (ML モデル設定) ページに直接移動します。[ML モデル設定] ページで、[ML モデル名] に対して、デフォルトの [**ML model: Banking Data 1**] が表示されていることを確認します。

   デフォルトなどのわかりやすい名前を使用すると、ML モデルを簡単に識別して管理するのに役立ちます。

2. [トレーニングおよび評価設定] で、[デフォルト] が選択されていることを確認します。



3. [この評価に名前を設定] で、デフォルト値 **Evaluation: ML model: Banking Data 1** をそのまま使用します。

4. [レビュー] を選択して、設定を確認し、[完了] を選択します。

   [Finish] (完了) を選択すると、Amazon ML は処理キューにモデルを追加します。Amazon ML がモデルを作成すると、デフォルトを適用して次のアクションを実行します。

   - 1 つはデータの 70% を、もう 1 つは残りの 30% を含むように、トレーニングデータソースを 2 つのセクションに分割します
   - 入力データの 70% を含むセクションで ML モデルをトレーニングします
   - 入力データの残りの 30% を使用してモデルを評価します

   モデルがキューに入っている間、Amazon ML はステータスを [Pending] (保留中) として報告します。Amazon ML がモデルを作成している間、Amazon ML はステータスを [In Progress] (進

行中) として報告します。すべてのアクションが完了すると、ステータスが [完了済み] としてレポートされます。続行する前に、評価が完了するまで待ちます。

これで、モデルのパフォーマンスを確認し、カットオフのスコアを設定する準備が整いました。

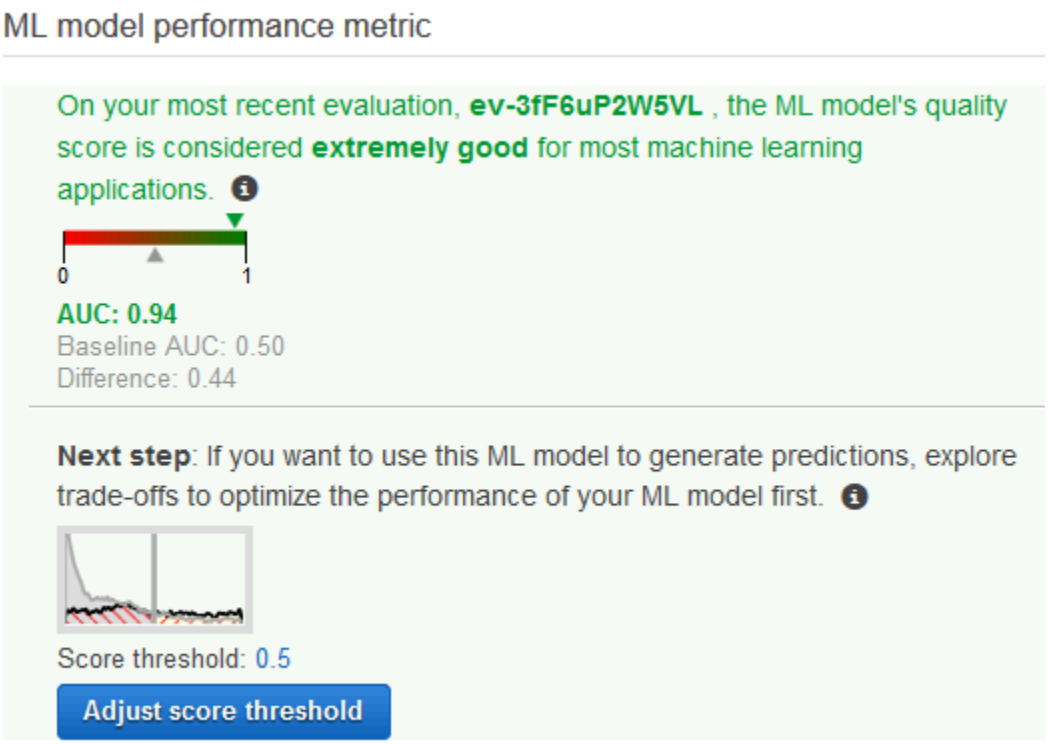
モデルのトレーニングおよび評価に関する詳細は、ML モデルのトレーニング および evaluate an ML model を参照してください。

ステップ 4: ML モデルの予測パフォーマンスを確認し、スコアのしきい値を設定する

ML モデルを作成し、Amazon Machine Learning (Amazon ML) によって評価したので、実際に使用できるかどうか見てみましょう。評価中、Amazon ML は、ML モデルのパフォーマンス品質を表現する曲線下面積 (AUC) メトリクスと呼ばれる業界標準の品質メトリクスを計算しました。また、Amazon ML は AUC メトリクスを解釈して、ML モデルの品質がほとんどの機械学習アプリケーションに適しているかどうかを知らせます。(AUC の詳細については、ML モデルの正確性の測定 を参照してください。) AUC メトリクスを確認した後、スコアのしきい値やカットオフを調整して、モデルの予測パフォーマンスを最適化しましょう。

ML モデルの AUC メトリクスを確認するには

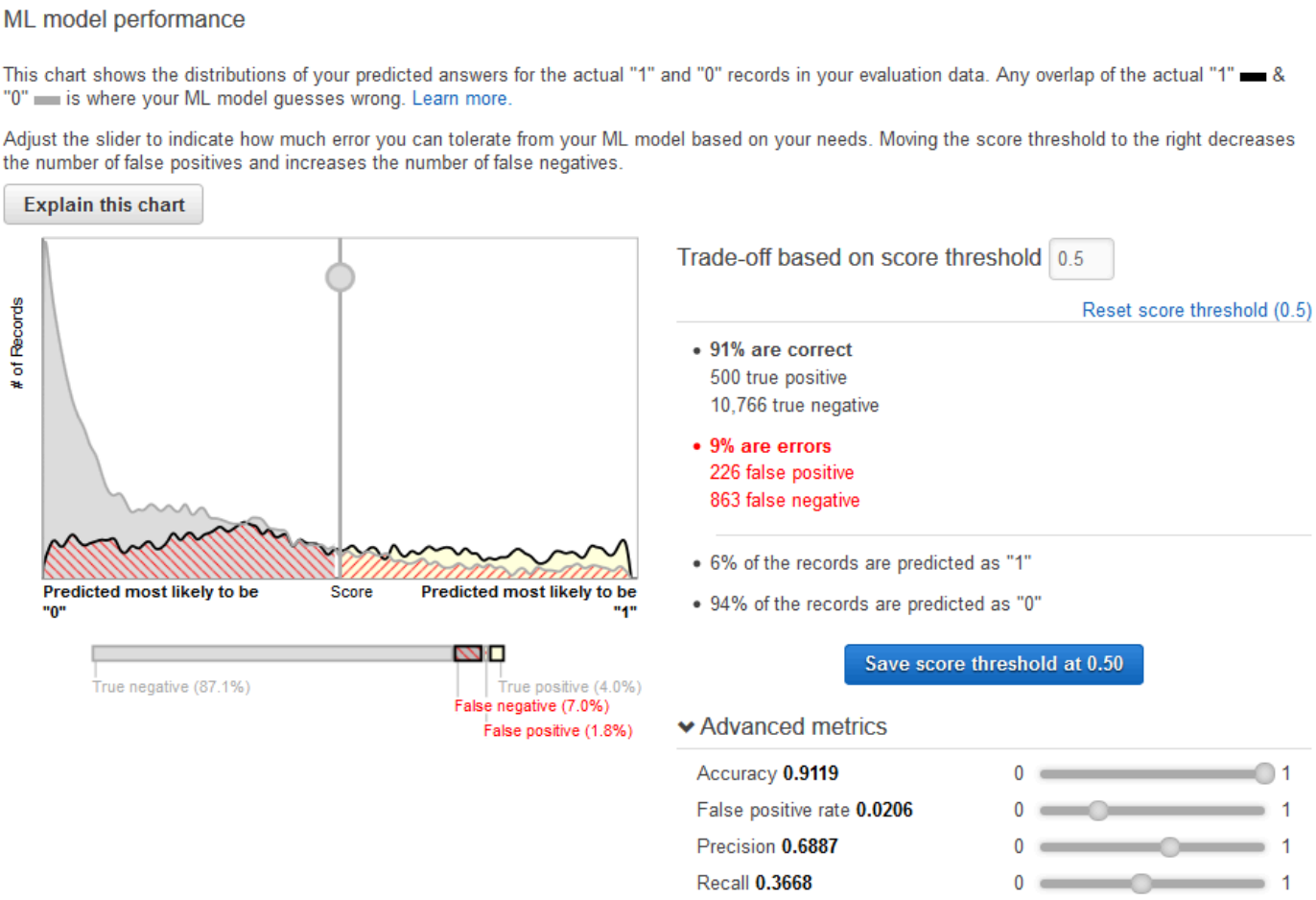
1. [ML モデルの要約] ページの [ML モデルレポート] ナビゲーションペインで、[評価]、[Evaluation: ML model: Banking model 1 (評価: ML モデル: 銀行モデル 1)]、[概要] の順に選択します。
2. [評価の概要] ページで、モデルの AUC パフォーマンスメトリクスを含め、評価の概要を確認します。



ML モデルは、予測データソース内の各レコードの数値予測スコアを生成し、しきい値を適用して 0 (いいえの場合) または 1 (はいの場合) のバイナリラベルに変換します。スコアのしきい値を変更することで、ML モデルがこれらのラベルをどのように割り当てるかを調整できます。ここで、スコアしきい値を設定します。

ML モデルのスコアのしきい値を設定するには

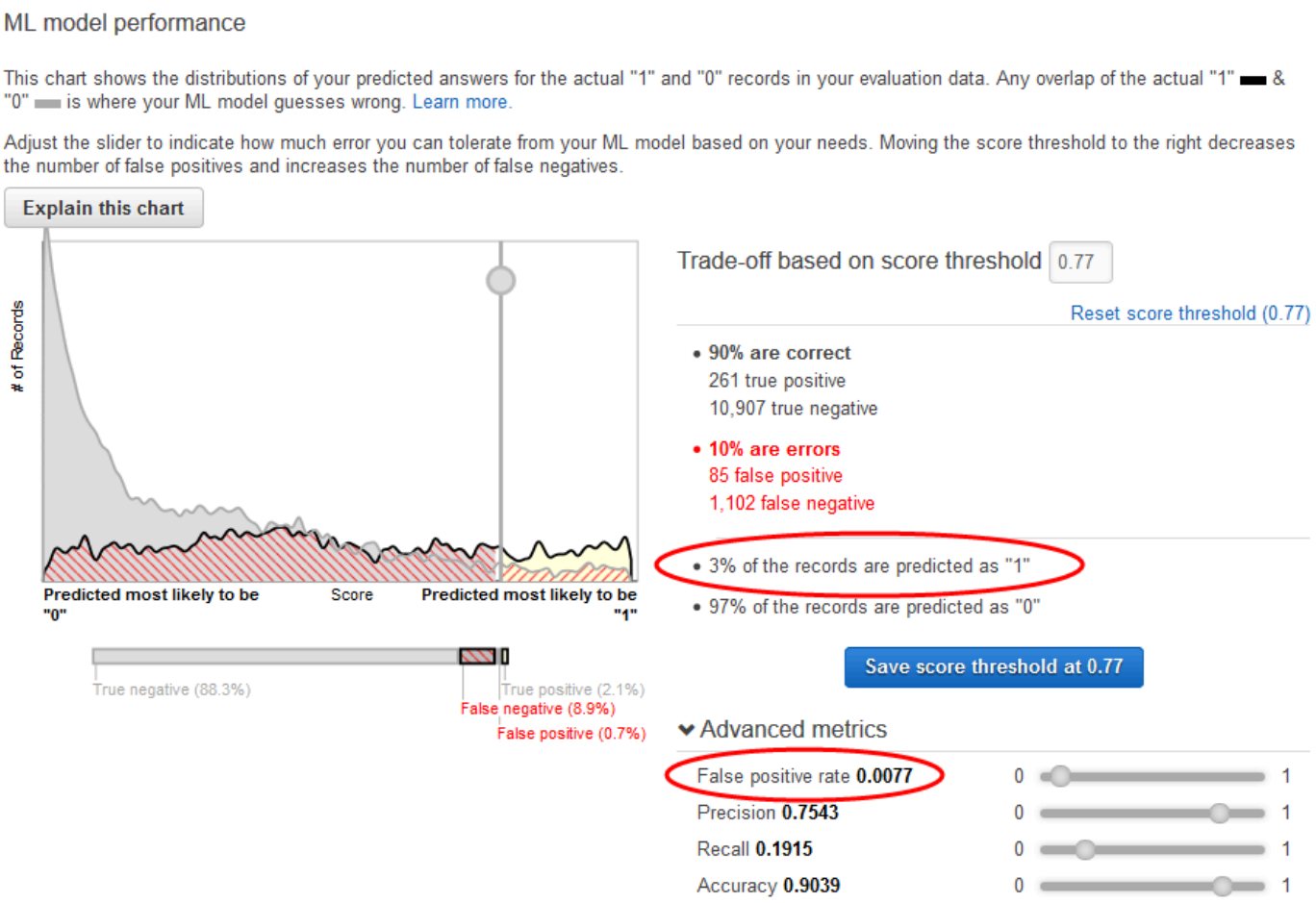
1. [評価の概要] ページで、[スコアしきい値を調整] を選択します。



スコアしきい値を調整することで ML モデルのパフォーマンスメトリクスを微調整できます。この値を調整すると、予測が正しいとみなされる前に、モデルが予測に対して持つ必要のある信頼度が変わります。また、予測で許容する誤検出および検出漏れの数が変更されます。

正しい検出である可能性が最も高い予測が正しいとみなされるまでスコアのしきい値を上げることによって、モデルが正しい予測とみなすもののカットオフを制御できます。また、検出漏れがなくなるまでスコアのしきい値を減らすこともできます。ビジネスのニーズに応じて、カットオフを選択します。このチュートリアルでは、誤検出の場合はキャンペーン費用がかかるため、誤検出よりも正しい検出の割合を高くします。

2. たとえば、商品をサブスクライブする顧客の上位 3% をターゲットに設定するとします。垂直セレクタをスライドさせて、スコアのしきい値が [3% of the records are predicted as "1" ("1" と予測されるコードの 3%)] になるよう設定します。



このスコアのしきい値が ML モデルのパフォーマンスに及ぼす影響に注意してください。誤検出率は 0.007 です。この誤検出率は許容できるとします。

3. [Save score threshold at 0.77 (スコアのしきい値を 0.77 で保存)] を選択します。

この ML モデルを使用して予測を行うたびに、0.77 を超えるレコードは「1」、残りのレコードは「0」と予測されます。

スコアのしきい値の詳細については、バイナリの分類 を参照してください。

これで、モデルを使用して予測を作成する準備ができました。

ステップ 5: ML モデルを使用して予測を生成する

Amazon Machine Learning (Amazon ML) は、バッチとリアルタイムの 2 種類の予測を生成できます。

リアルタイム予測は、Amazon ML がオンデマンドで生成する単一の観測に対する予測です。リアルタイム予測は、モバイルアプリ、ウェブサイト、および結果をインタラクティブに使用する必要のある他のアプリケーションに最適です。

バッチ予測は、一連の観測に対する予測のセットです。Amazon ML はバッチ予測でレコードを処理するため、処理に時間がかかることがあります。 がインタラクティブに結果を使用しない一連の観測または予測を必要とするアプリケーションには、バッチ予測を使用します。

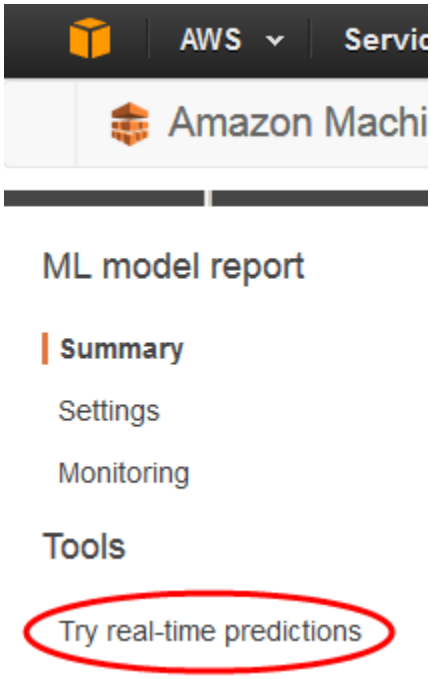
このチュートリアルでは、潜在的な顧客が新製品をサブスクライブするかどうかを予測するリアルタイム予測を生成します。潜在的な顧客の大きなバッチ予測も生成します。バッチ予測の場合は、「banking-batch.csv」でアップロードした ステップ 1: データを準備する ファイルを使用します。

リアルタイム予測を作成しましょう。

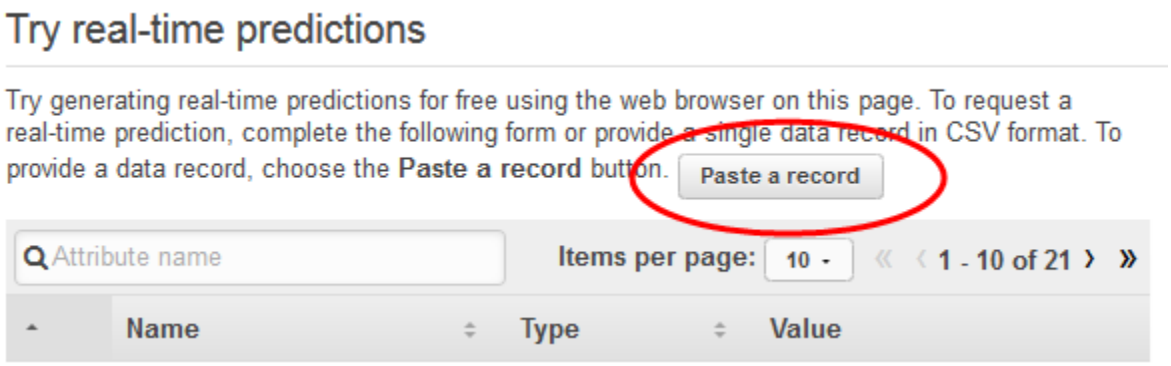
> **Note**
>
> リアルタイム予測が必要なアプリケーションの場合は、ML モデルのリアルタイムエンドポイントを作成する必要があります。リアルタイムエンドポイントが利用可能な間は料金が発生します。リアルタイム予測を使用して関連するコストを負担する前に、リアルタイムエンドポイントを作成せずにウェブブラウザでリアルタイム予測機能を試すことができます。このチュートリアルでは、以下を実行します。

リアルタイム予測を試用するには

1. [ML モデルレポート] ナビゲーションペインで、[Try real-time predictions (リアルタイム予測の試用)] を選択します。



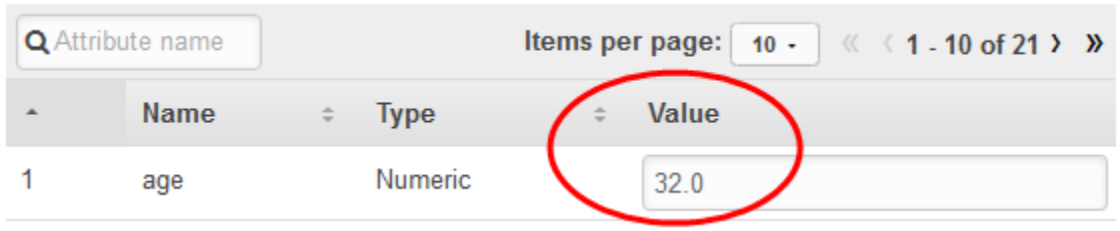
2. [レコードの貼り付け] を選択します。



3. [レコードの貼り付け] ダイアログボックスで、次の条件を貼り付けます。

```
32,services,divorced,basic.9y,no,unknown,yes,cellular,dec,mon,110,1,11,0,nonexistent,-1.8,9
```

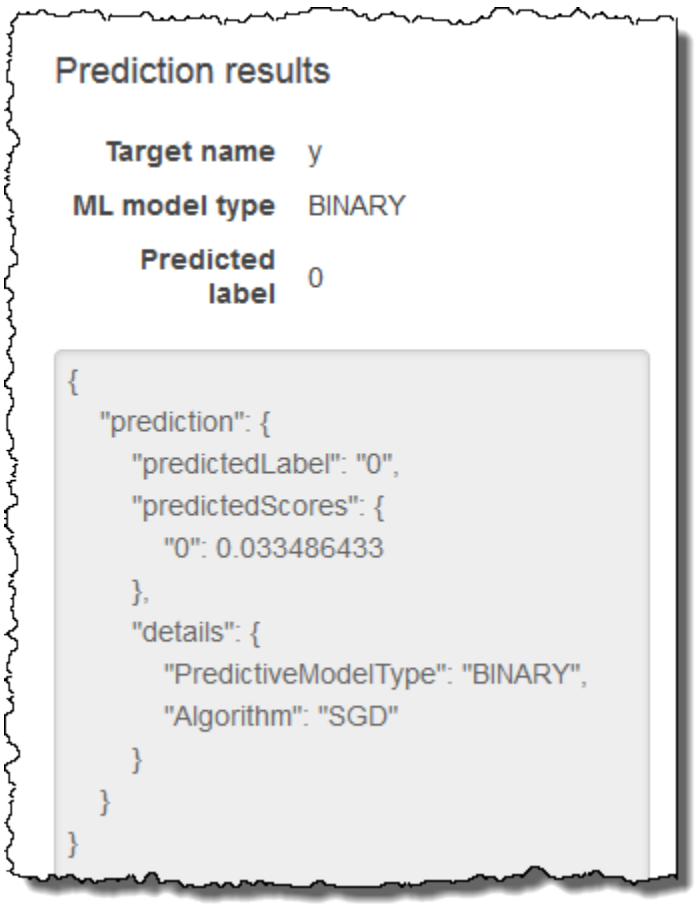
4. [Paste a record] (レコードの貼り付け) ダイアログボックスで [Submit] (送信) を選択し、この観測の予測を生成することを確認します。Amazon ML はリアルタイム予測フォームに値を入力します。



> **Note**
>
> 個々の値を入力して、[値] フィールドに値を入力することもできます。選択したメソッドにかかわらず、モデルをトレーニングするために使用しなかった観察を提供する必要があります。

5. ページの下部で、[予測の作成] を選択します。

   予測は右側の [Prediction results (予測結果)] ペインに表示されます。この予測の [Predicted label (予測ラベル)] は `0` です。つまり、この潜在的な顧客はキャンペーンに反応しそうにありません。[Predicted label (予測ラベル)] 1 は、顧客がキャンペーンに応答する可能性が高いことを意味します。

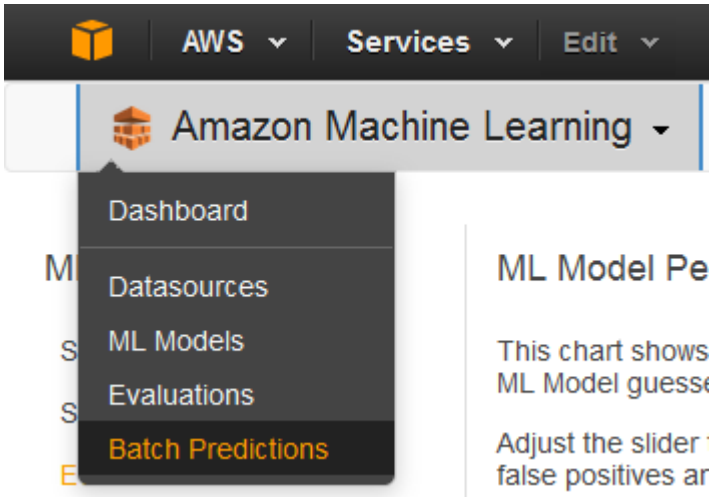


ここで、バッチ予測を作成します。Amazon ML に使用している ML モデルの名前、および予測を生成する入力データの Amazon Simple Storage Service (Amazon S3) の場所 (Amazon ML はこのデー

タからバッチ予測データソースを作成します)、結果を格納するための Amazon S3 の場所を提供します。

バッチ予測を作成するには

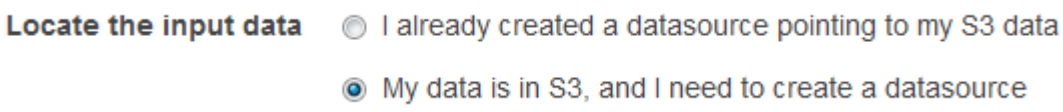
1. [Amazon Machine Learning]、[バッチ予測] の順に選択します。



2. [新しいバッチ予測を生成] を選択します。
3. [ML model for batch predictions (バッチ予測の ML モデル)] ページで、[ML model: Banking Data 1 (ML モデル: 銀行データ 1)] を選択します。

   Amazon ML は、ML モデル名、ID、作成時刻、および関連するデータソース ID を表示します。
4. [Continue] (続行) をクリックします。
5. 予測を生成するには、予測に必要なデータを Amazon ML に提供する必要があります。これは入力データと呼ばれます。まず、入力データをデータソースに入力して、Amazon ML からアクセスできるようにします。

   [入力データを特定] で、[データは S3 にあり、データソースを作成する必要があります] を選択します。



6. [データソース名] に「**`Banking Data 2`**」を入力します。
7. [S3 の場所] に、`banking-batch.csv` ファイルの完全な場所を入力します: *`your-bucket/banking-batch.csv`***`/banking-batch.csv`**。
8. [CSV の 1 行目には列名が含まれていますか?] で、[はい] を選択します。

9. [検証] を選択します。

   Amazon ML はデータの場所を検証します。

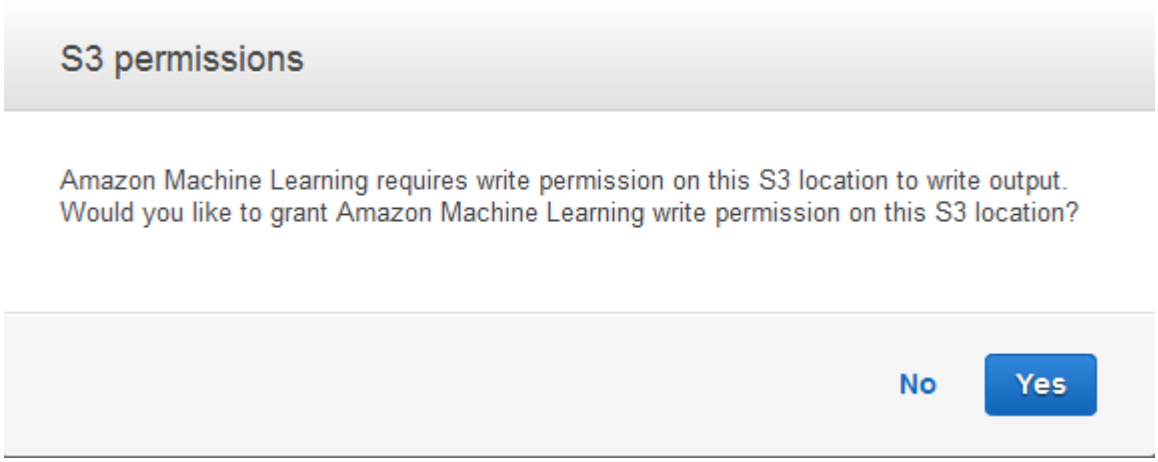
10. [Continue] (続行) をクリックします。

11. [S3 destination] (S3 送信先) には、「ステップ 1: データを準備する」でファイルをアップロードした Amazon S3 の場所の名前を入力します。Amazon ML はそこに予測結果をアップロードします。

12. [Batch prediction name] (バッチ予測名) では、デフォルトの **`Batch prediction: ML model: Banking Data 1`** をそのまま使用します。Amazon ML は、予測の作成に使用するモデルに基づいてデフォルトの名前を選択します。このチュートリアルでは、モデルと予測にはトレーニングデータソースの後に `Banking Data 1`という名前が付けられています。

13. [Review] (レビュー) を選択します。

14. [S3 アクセス権限] ダイアログボックスで、[はい] を選択します。

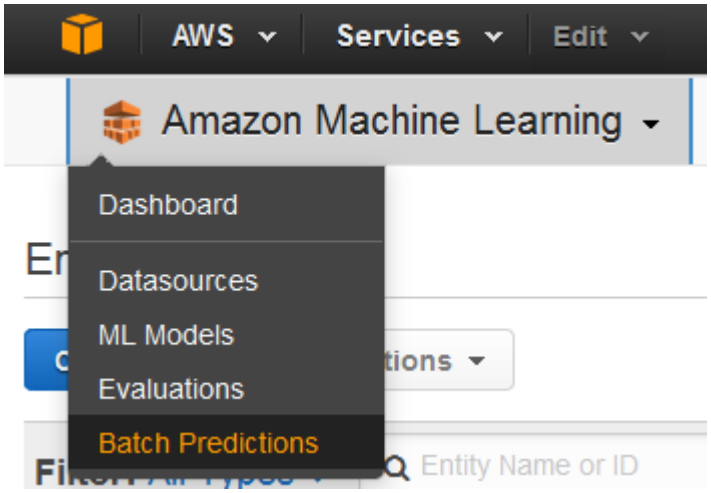


15. [レビュー] ページで、[完了] を選択します。

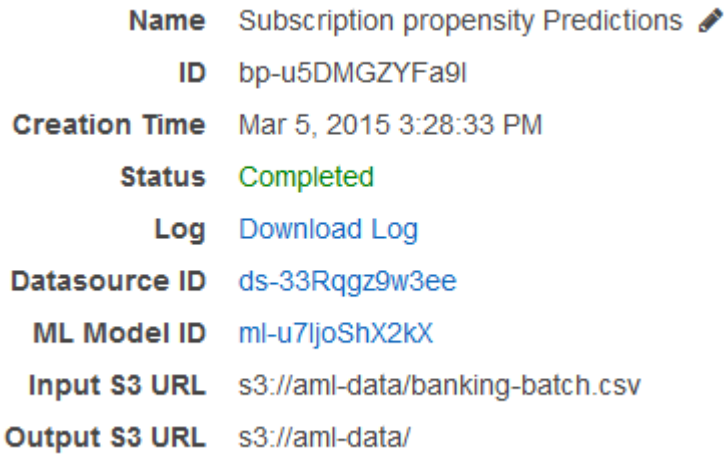
    バッチ予測のリクエストが Amazon ML に送信され、キューに入ります。Amazon ML がバッチ予測を処理するのにかかる時間は、データソースのサイズと ML モデルの複雑さによって異なります。Amazon ML はリクエストを処理している間、[In Progress] (進行中) のステータスを報告します。バッチ予測が完了すると、リクエストのステータスが [完了済み] に変わります。ここで、結果を表示することができます。

予測を表示するには

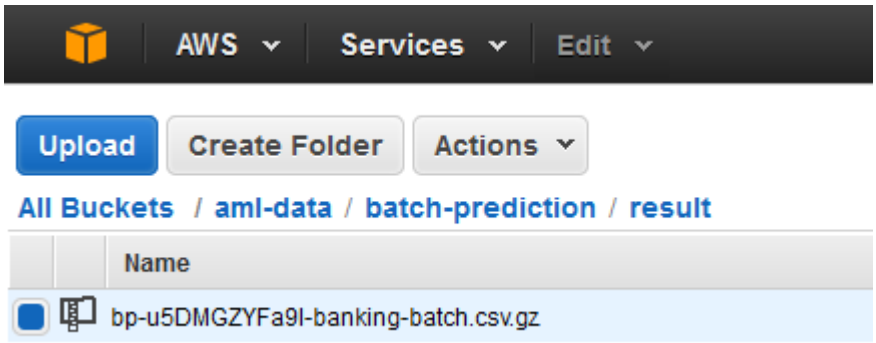
1. [Amazon Machine Learning]、[バッチ予測] の順に選択します。



2. 予測一覧で、[Batch prediction: ML model: Banking Data 1 (バッチ予測: ML モデル: 銀行データ 1)] を選択します。[Batch prediction info (バッチ予測情報)] ページが表示されます。



3. バッチ予測の結果を表示するには、Amazon S3 コンソール (https://console.aws.amazon.com/s3/) にアクセスし、[Output S3 URL] (S3 出力 URL) フィールドで参照される Amazon S3 の場所に移動します。ここから `s3://aml-data/batch-prediction/result` と似た名前の結果フォルダに移動します。



予測は、.gz 拡張子の付いた圧縮された .gzip ファイルに保存されます。

4. 予測ファイルをデスクトップにダウンロードし、解凍して開きます。

bestAnswer	score
0	0.06046
0	0.00507
0	0.01410
0	0.00170
0	0.00184
0	0.07133
0	0.30811

このファイルには [bestAnswer] と [score] の 2 つの列と、データソース内の各観測の行があります。[bestAnswer] 列の結果は、「ステップ 4: ML モデルの予測パフォーマンスを確認し、スコアのしきい値を設定する」で設定したスコアのしきい値 0.77 に基づいています。0.77 を超える score は、bestAnswer が 1 であり、これは肯定的な応答または予測であり、score が 0.77 未満では bestAnswer が 0 であり、これは否定的な応答または予測です。

次の例は、スコアしきい値 0.77 に基づく正および負の予測を示します。

正の予測。

bestAnswer	score
1	0.8228876

この例では、[bestAnswer] の値は 1 で、score の値は 0.8228876 となります。[bestAnswer] の値が 1 であるのは、score がスコアのしきい値 0.77 よりも大きいためです。[bestAnswer] が 1 の場合、顧客が製品を購入する可能性が高いことを示すため、肯定的な予測とみなされます。

負の予測。

bestAnswer	score
0	0.7695356

この例では、[bestAnswer] の値は 0 です。これは score の値が、スコアのしきい値である 0.77 よりも少ない 0.7695356 であるためです。[bestAnswer] が 0 の場合、顧客が製品を購入する可能性が低いことを示すため、否定的な予測とみなされます。

バッチの各行は、バッチ入力 (データソースの監視) の行に対応します。

予測を分析した後、ターゲットを絞ったマーケティングキャンペーンを実行できます。たとえば、予測スコア 1 のすべての人にチラシを送ることができます。

これで、モデルを作成し、レビューして、使用したので、作成したデータと AWS リソースをクリーンアップして、不要な課金が発生しないようにし、ワークスペースを整った状態に保ちます。

ステップ 6: クリーンアップ

Amazon Simple Storage Service (Amazon S3) の追加料金を回避するために、Amazon S3 に保存されているデータを削除します。他の未使用 Amazon ML リソースに対して料金は発生しませんが、それらを削除してワークスペースを整理することをお勧めします。

Amazon S3 に保存されている入力データを削除するには

1. Amazon S3 コンソール (https://console.aws.amazon.com/s3/) を開きます。
2. `banking.csv` および `banking-batch.csv` ファイルが保存されている Amazon S3 の場所に移動します。
3. `banking.csv`、`banking-batch.csv`、および `.writePermissionCheck.tmp` ファイルを選択します。
4. [Actions] (アクション) を選択してから、[Delete] (削除) をクリックします。
5. 確認を求めるメッセージが表示されたら、[OK] を選択します。

Amazon ML が実行したバッチ予測の記録、またはチュートリアルで作成したデータソース、モデル、および評価を保存しても課金されませんが、ワークスペースが乱雑にならないようにそれらを削除することをお勧めします。

バッチ予測を削除するには

1. バッチ予測の出力を保存した Amazon S3 の場所に移動します。
2. `batch-prediction` フォルダを選択します。
3. [Actions] (アクション) を選択してから、[Delete] (削除) をクリックします。
4. 確認を求めるメッセージが表示されたら、[OK] を選択します。

Amazon ML リソースを削除するには

1. Amazon ML ダッシュボードで、次のリソースを選択します。

- Banking Data 1 データソース
- Banking Data 1_[percentBegin=0, percentEnd=70, strategy=sequential] データソース
- Banking Data 1_[percentBegin=70, percentEnd=100, strategy=sequential] データソース
- Banking Data 2 データソース
- ML model: Banking Data 1 ML モデル
- Evaluation: ML model: Banking Data 1 評価

2. [Actions] (アクション) を選択してから、[Delete] (削除) をクリックします。
3. ダイアログボックスで、[削除] を選択してすべての選択したリソースを削除します。

これでチュートリアルは完了です。コンソールを使用してデータソース、モデル、および予測の作成を続行するには、「Amazon Machine Learning デベロッパーガイド」を参照してください。API を使用する方法については、Amazon Machine Learning API リファレンスを参照してください。

データソースの作成と使用

Amazon ML データソースを使用して、ML モデルをトレーニングし、ML モデルを評価し、ML モデルを使用してバッチ予測を生成することができます。データソースオブジェクトには、入力データに関するメタデータが含まれています。データソースを作成すると、Amazon ML は入力データを読み取り、その属性に関する記述統計を計算し、統計、スキーマ、およびその他の情報をデータソースオブジェクトの一部として格納します。データソースを作成した後、Amazon ML のデータインサイトを使用して入力データの統計プロパティを調べることができ、データソースを使用して、ML モデルをトレーニングできます。

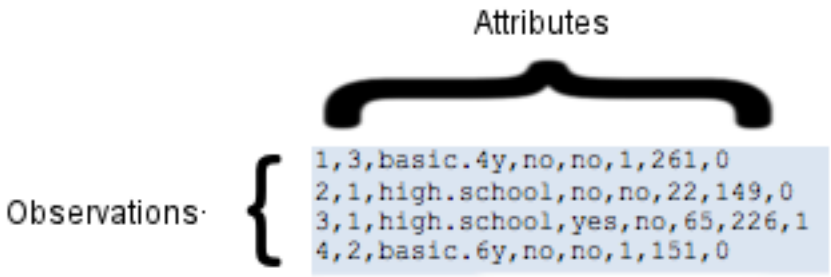
> **Note**
>
> このセクションでは、Amazon Machine Learning の概念に精通していることを前提としています。

トピック

- Amazon ML のデータ形式について
- Amazon ML のデータスキーマを作成する
- データの分割
- データ洞察
- Amazon ML での Amazon S3 の使用
- Amazon Redshift のデータから Amazon ML データソースを作成する
- Amazon RDS データベースのデータを使用して Amazon ML データソースを作成する

Amazon ML のデータ形式について

入力データは、データソースの作成に使用するデータです。入力データは、カンマ区切り値 (.csv) 形式で保存する必要があります。csv ファイルの各行は、単一のデータレコードまたは観測データです。.csv ファイルの各列には、観測の属性が含まれています。たとえば、次の図は、それぞれが独自の行に 4 つの観測値を持つ .csv ファイルの内容を示しています。各観測には、コンマで区切られた 8 つの属性が含まれています。属性は、customerId、jobId、教育、住宅、ローン、キャンペーン、期間、willRespondToCampaign のような、観察によって表される各個人に関する以下の情報を表します。



属性

Amazon ML では各属性の名前が必要です。属性名は次のように指定できます。

- 入力データとして使用する .csv ファイルの最初の行 (ヘッダー行とも呼ばれます) に属性名を含める
- 入力データと同じ S3 バケットにある別のスキーマファイルに属性名を含める

スキーマファイルの使用の詳細については、「データスキーマの作成」を参照してください。

次の .csv ファイルの例には、ヘッダー行の属性の名前が含まれています。

```
customerId,jobId,education,housing,loan,campaign,duration,willRespondToCampaign

1,3,basic.4y,no,no,1,261,0

2,1,high.school,no,no,22,149,0

3,1,high.school,yes,no,65,226,1

4,2,basic.6y,no,no,1,151,0
```

入力ファイル形式の要件

入力データを含む .csv ファイルは、次の要件を満たしている必要があります。

- ASCII、Unicode、EBCDIC などの文字セットを使用したプレーンテキストでなければなりません。
- 観測値で構成され、1 行に 1 つの観測値があります。

- 観測ごとに、属性値をコンマで区切る必要があります。
- 属性値にコンマ (区切り文字) が含まれている場合は、属性値全体を二重引用符で囲む必要があります。
- 各観測は、行の終わりを示す特殊文字または一連の文字である行末文字で終了する必要があります。
- 属性値が二重引用符で囲まれていても、属性値に行末文字を含めることはできません。
- すべての観測は、同じ数の属性と一連の属性を持っていなければなりません。
- 各観測値は 100 KB 以下でなければなりません。Amazon ML は、処理中に 100 KB を超えるすべての観測を拒否します。Amazon ML が 10,000 を超える観測を拒否すると、.csv ファイル全体が拒否されます。

Amazon ML へのデータ入力として複数のファイルを使用する

Amazon ML に入力を単一のファイルとして提供することも、ファイルの集合として提供することもできます。コレクションは次の条件を満たす必要があります。

- すべてのファイルに同じデータスキーマが必要です。
- すべてのファイルは同じ Amazon Simple Storage Service (Amazon S3) プレフィックスに存在し、コレクションに指定するパスはスラッシュ (/) で終わらなければなりません。

たとえば、データファイルの名前が input1.csv、input2.csv、および input3.csv で、S3 バケット名が s3://examplebucket の場合、ファイルパスは次のようになります。

s3://examplebucket/path/to/data/input1.csv

s3://examplebucket/path/to/data/input2.csv

s3://examplebucket/path/to/data/input3.csv

次の S3 の場所を Amazon ML の入力として提供します。

's3://examplebucket/path/to/data/'

CSV 形式の行末文字

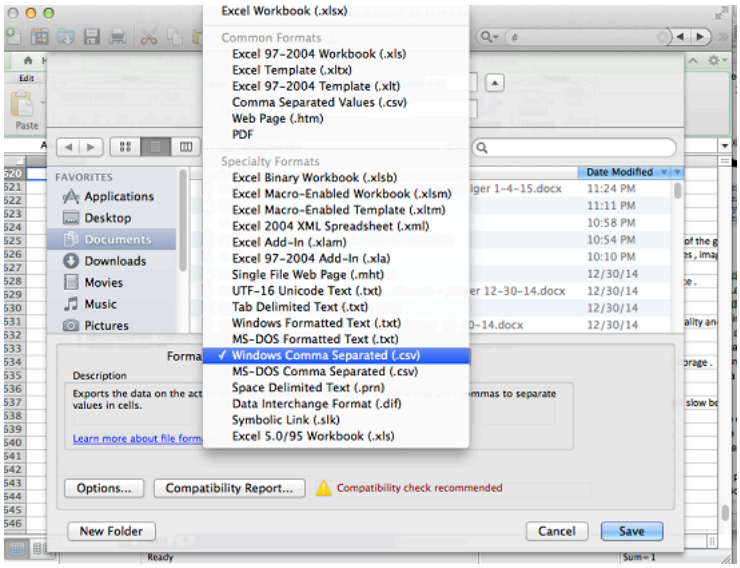
.csv ファイルを作成すると、各観測は特殊な行末文字で終了します。この文字は表示されませんが、Enter キーまたは Return キーを押すと、各観測の最後に自動的に含まれます。行末を表す特

殊文字は、オペレーティングシステムによって異なります。Linux や OS X などの Unix システムでは、\n (10 進数の ASCII コード 10 または 16 進数の 0x0a) で示されるラインフィード文字を使用します。Microsoft Windows では、\r\n (10 進数の ASCII コード 13 および 10、または 16 進数では 0x0d および 0x0a) で示される改行およびラインフィードと呼ばれる 2 つの文字を使用します。

OS X および Microsoft Excel を使用して .csv ファイルを作成する場合は、次の手順を実行します。正しい形式が選択されていることを確認してください。

OS X および Excel を使用して .csv ファイルを保存するには

1. .csv ファイルを保存するときは、[形式] を選択し、[Windows カンマ区切り (.csv)] を選択します。
2. [Save (保存)] を選択します。



> **Important**
>
> Amazon ML が読み込めないため、カンマ区切りの値 (.csv) または MS-DOS カンマ区切り (.csv) 形式を使用して .csv ファイルを保存しないでください。

Amazon ML のデータスキーマを作成する

スキーマは、入力データとそれに対応するデータタイプのすべての属性で構成されています。これにより、Amazon ML はデータソース内のデータを理解します。Amazon ML は、スキーマ内の情報を使用して、入力データの読み取りと解釈、統計の計算、正しい属性変換の適用、学習アルゴリズムの微調整を行います。スキーマを指定しない場合、Amazon ML はデータから推測します。

スキーマの例

Amazon ML が入力データを正しく読み取り、正確な予測を生成するには、各属性が正しいデータ型に割り当てられている必要があります。データ型が属性にどのように割り当てられているか、および、属性とデータ型がスキーマにどのように含まれているかの例を見てみましょう。どの顧客が E メールキャンペーンに反応するかを予測するために、「お客様キャンペーン」の例を呼び出します。入力ファイルは 9 つの列がある .csv ファイルです。

```
1,3,web developer,basic.4y,no,no,1,261,0

2,1,car repair,high.school,no,no,22,149,0

3,1,car mechanic,high.school,yes,no,65,226,1

4,2,software developer,basic.6y,no,no,1,151,0
```

以下は、このデータのスキーマです。

```
{
    "version": "1.0",
    "rowId": "customerId",
    "targetAttributeName": "willRespondToCampaign",
    "dataFormat": "CSV",
    "dataFileContainsHeader": false,
    "attributes": [
        {
            "attributeName": "customerId",
            "attributeType": "CATEGORICAL"
        },
        {
            "attributeName": "jobId",
            "attributeType": "CATEGORICAL"
        },
        {
            "attributeName": "jobDescription",
            "attributeType": "TEXT"
        },
        {
            "attributeName": "education",
            "attributeType": "CATEGORICAL"
        },
        {
```

```
            "attributeName": "housing",
            "attributeType": "CATEGORICAL"

        },
        {
            "attributeName": "loan",
            "attributeType": "CATEGORICAL"

        },
        {
            "attributeName": "campaign",
            "attributeType": "NUMERIC"
        },
        {
            "attributeName": "duration",
            "attributeType": "NUMERIC"
        },

        {
            "attributeName": "willRespondToCampaign",
            "attributeType": "BINARY"
        }

    ]
}
```

この例のスキーマファイルでは、rowId の値が customerId となっています。

```
"rowId": "customerId",
```

属性 willRespondToCampaign はターゲット属性として定義されています。

```
"targetAttributeName": "willRespondToCampaign ",
```

customerId 属性と CATEGORICAL データ型は最初の列に関連付けられていて、jobId 属性と CATEGORICAL データ型は 2 番目の列に関連付けられていて、jobDescription 属性と TEXT データ型は 3 番目の列に関連付けられていて、education 属性と CATEGORICAL データ型は 4 番目の列に関連付けられていて、以下同様に続きます。9 番目の列は willRespondToCampaign 属性と BINARY データ型に関連付けられていて、この属性はターゲット属性としても定義されています。

targetAttributeName フィールドの使用

targetAttributeName 値は、予測する属性の名前です。モデルを作成または評価するときは、targetAttributeName を割り当てる必要があります。

ML モデルのトレーニングや評価をするときは、targetAttributeName は、ターゲット属性の「正しい」回答を含む入力データ内の属性の名前を識別します。Amazon ML は、正解を含むターゲットを使用して、パターンを検出し、ML モデルを生成します。

モデルを評価する際、Amazon ML はターゲットを使用して予測の精度をチェックします。ML モデルを作成して評価したら、割り当てられいない targetAttributeName のデータを使用して、ML モデルで予測を生成することができます。

データソースを作成するとき Amazon ML コンソールで、または、スキーマファイルで、ターゲット属性を定義します。独自のスキーマファイルを作成する場合は、次の構文を使用してターゲット属性を定義します。

```
"targetAttributeName": "exampleAttributeTarget",
```

この例で、exampleAttributeTarget はダーゲット属性である入力ファイル内の属性の名前です。

rowID フィールドの使用

row ID は、入力データの属性に関連付けられているオプションのフラグです。指定した場合、row ID とマークされた属性が予測出力に含まれます。この属性を使用すると、予測と観測の対応性を関連付けしやすくなります。良い row ID の例は、カスタマー ID または同様の一意の属性です。

> **Note**
>
> 行 ID は参照用です。ML モデルをトレーニングするとき、Amazon ML がそれを使用することはありません。行 ID として属性を選択すると、ML モデルのトレーニングへは使用されなくなります。

データソースを作成するとき Amazon ML コンソールで、または、スキーマファイルで、row ID を定義します。独自のスキーマファイルを作成する場合は、次の構文を使用して row ID を定義します。

```
"rowId": "exampleRow",
```

前述の例で、exampleRow は行 ID として定義された入力ファイル内の属性の名前です。

バッチ予測を生成する場合、次のような出力が得られる場合があります。

```
tag,bestAnswer,score
55,0,0.46317
102,1,0.89625
```

この例では、RowID は属性 customerId を表します。たとえば、customerId 55 は低い信頼度 (0.46317) で E メールキャンペーンに反応することが予測され、一方、customerId 102 は高い信頼度 (0.89625) で E メールキャンペーンに反応することが予測されます。

AttributeType フィールドの使用

Amazon ML では、属性に 4 つのデータ型があります。

バイナリ

2 つの可能な状態のみを持つ属性に BINARY を選択します (yes または no など)。

たとえば、ある人が新規顧客であるかどうかを追跡するための isNew 属性には、その人が新規顧客であることを示す true 値と、その人が新規顧客ではないことを示す false 値があります。

有効な負の値は、0、n、no、f、および false です。

有効な正の値は、1、y、yes、t、および true です。

Amazon ML はバイナリ入力の大文字と小文字は無視し、周囲の空白は取り除きます。たとえば、" FaLSe " は有効なバイナリ値です。同じデータソース内で、使用するバイナリ値を混在させることができます (true、no、および 1 など)。バイナリ属性のための Amazon ML の出力は、0 と 1 のみです。

カテゴリ

限られた数の一意の文字列値を取る属性に CATEGORICAL を選択します。たとえば、ユーザー ID、月、郵便番号はカテゴリ値です。カテゴリ属性は単一文字列として扱われ、さらにトークン分割されることはありません。

数値

数量を値として取る属性には NUMERIC を選択します。

たとえば、温度、重量、およびクリック率は数値です。

数字を含むすべての属性が数値であるとは限りません。日付、ID などのカテゴリ属性は、多くの場合、数字で表されます。数値とみなされるには、数字が別の数字と比較できる必要があります。たとえば、顧客 ID 664727 からは顧客 ID 124552 について何も分かりませんが、重量 10 という属性は、その属性が重量 5 を持つ属性よりも重いことを意味します。日付は、ある月の 1 日が別の月の 2 日よりも前あるいは後のどちらにでも来る可能性があるため、数値ではありません。

> **Note**
>
> スキーマの作成に Amazon ML を使用する場合は、数字を使用するすべての属性に Numeric データ型が割り当てられます。Amazon ML がスキーマを作成する場合は、誤った割り当てをチェックし、それらの属性を CATEGORICAL に設定します。

[Text] (テキスト)

単語の文字列である属性には TEXT を選択します。テキスト属性は、読み込むとき Amazon ML によりトークンに変換され、空白で区切られます。

たとえば、`email subject` は `email` と `subject` になり、`email-subject here` は `email-subject` と `here` になります。

トレーニングスキーマの変数のデータ型が評価スキーマの変数のデータ型と一致しない場合、Amazon ML は評価データ型をトレーニングデータ型に合わせて変更します。例えば、トレーニングデータスキーマが TEXT のデータ型を変数 age に割り当てても、評価スキーマが NUMERIC のデータ型を age に割り当てる場合、Amazon ML は評価データで年齢を NUMERIC ではなく TEXT 変数として扱います。

各データ型に関連付けられた統計の詳細については、「記述統計」を参照してください。

Amazon ML にスキーマを提供する

すべてのデータソースにはスキーマが必要です。Amazon ML にスキーマを提供するには、2 つの方法から選択できます。

- Amazon ML が入力データファイル内の各属性のデータ型を推測し、自動的にスキーマを作成するのを許可します。
- Amazon Simple Storage Service (Amazon S3) データをアップロードするときに、スキーマファイルを提供します。

Amazon ML によるスキーマの作成を許可する

Amazon ML コンソールを使用してデータソースを作成すると、Amazon ML は変数の値に基づいた単純なルールを使用してスキーマを作成します。Amazon ML で作成したスキーマを確認し、正確でない場合はデータタイプを修正することを強くお勧めします。

スキーマを提供する

スキーマファイルを作成したら、Amazon ML で使用可能にする必要があります。これには 2 つのオプションがあります。

1. Amazon ML コンソールを使用してスキーマを提供します。

   コンソールを使用してデータソースを作成し、入力データファイルのファイル名に .schema 拡張子を追加してスキーマファイルを組み込みます。例えば、入力データへの Amazon Simple Storage Service (Amazon S3) URI が s3://my-bucket-name/data/input.csv である場合、スキーマへの URIは s3://my-bucket-name/data/input.csv.schema になります。Amazon ML は、データからスキーマを推測するのではなく、提供されたスキーマファイルを自動的に特定します。

   Amazon ML へのデータ入力としてファイルのディレクトリを使用するには、.schema 拡張子をディレクトリパスに追加します。たとえば、データファイルが s3://examplebucket/path/to/data/ という場所にある場合、スキーマへの URI は s3://examplebucket/path/to/data/.schema となります。
2. Amazon ML API を使用してスキーマを提供します。

   Amazon ML API を呼び出してデータソースを作成する予定の場合は、スキーマファイルを Amazon S3 にアップロードして、`CreateDataSourceFromS3` API の `DataSchemaLocationS3` 属性でそのファイルへの URI を提供します。詳細については、「CreateDataSourceFromS3」を参照してください。

   最初に Amazon S3 に保存する代わりに、`CreateDataSource*` APIs のペイロードで直接スキーマを提供することができます。これを行うには、`DataSchema`、`CreateDataSourceFromS3`、または `CreateDataSourceFromRDS` API の `CreateDataSourceFromRedshift` 属性にスキー

マの文字列をすべて配置します。詳細については、「Amazon 機械学習 API リファレンス」を参照してください。

データの分割

ML モデルの基本的な目標は、モデルをトレーニングするのに使用されるものを超えた将来のデータインスタンスについての正確な予測を行うことです。ML モデルを使用して予測を行う前に、モデルの予測パフォーマンスを評価する必要があります。未知のデータを含む ML モデル予測の品質を評価するために、すでに答えを知っているデータの一部を将来のデータのプロキシとして予約または分割し、ML モデルがそのデータの答えをどれだけ正しく予測するかを評価します。データソースをトレーニングデータソース用と評価データソース用に分割します。

Amazon ML には、データを分割するための 3 つのオプションがあります。

- データの事前分割 - Amazon Simple Storage Service (Amazon S3) にアップロードして 2 つの別々のデータソースを作成する前に、データを 2 つのデータ入力場所に分割することができます。
- Amazon ML シーケンシャル分割 - トレーニングデータソースと評価データソースを作成する際に Amazon ML にデータをシーケンシャルに分割するよう指示できます。
- Amazon ML ランダム分割 - トレーニングデータソースと評価データソースを作成する際に Amazon ML にシードされたランダムメソッドを使用してデータを分割するよう指示できます。

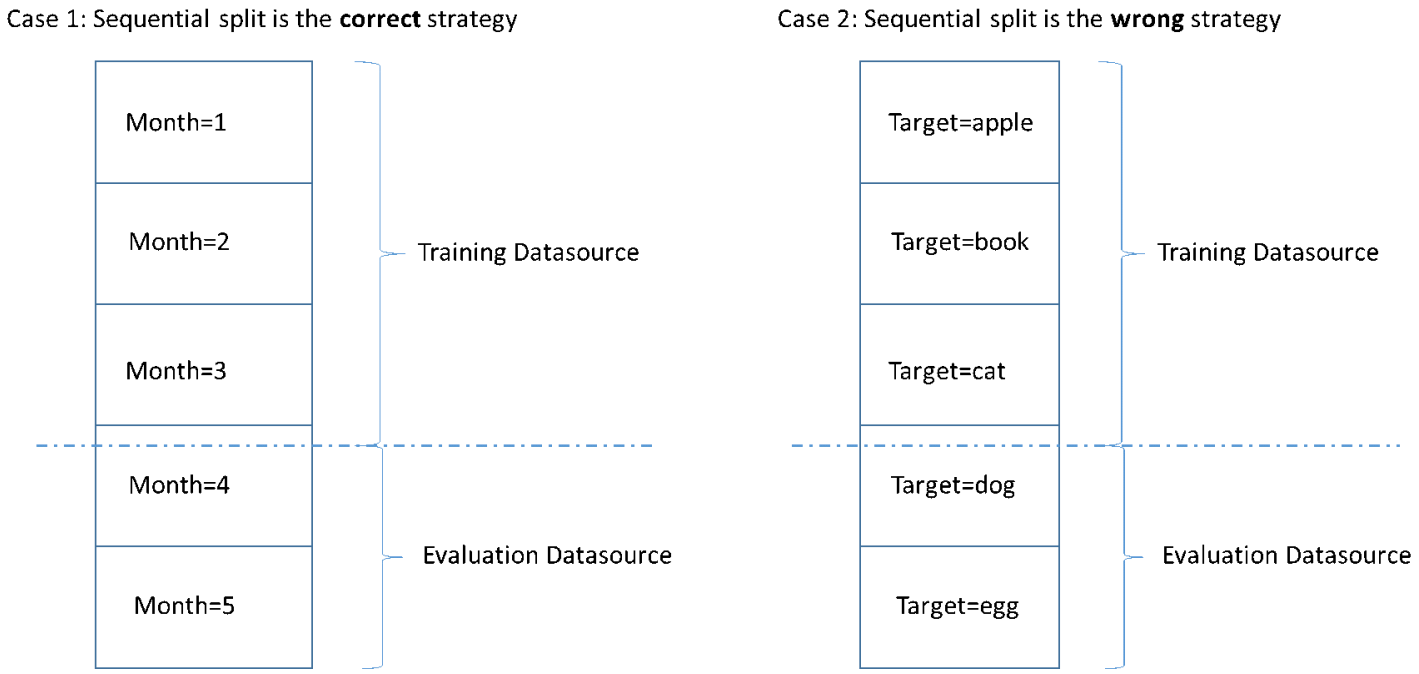
データの事前分割

トレーニングデータソースと評価データソースでデータを明示的に制御する場合は、データを別々のデータ場所に分割し、入力場所と評価場所の個別のデータソースを作成します。

データのシーケンシャルな分割

トレーニングと評価のために入力データを分割する簡単な方法は、データレコードの順序を保持しながら、データの重複していないサブセットを選択することです。このアプローチは、特定の日付または特定の時間範囲内のデータに対して ML モデルを評価する場合に便利です。たとえば、過去 5 か月間の顧客エンゲージメントデータがあり、この履歴データを使用して翌月の顧客エンゲージメントを予測するとします。トレーニングの範囲の始まりと評価範囲の終わりからのデータを使用すると、データ範囲全体から取得されたレコードデータを使用するよりも、モデルの品質をより正確に推定できます。

次の図は、シーケンシャル分割戦略を使用する必要がある場合と、ランダム戦略を使用する必要がある場合の例を示しています。



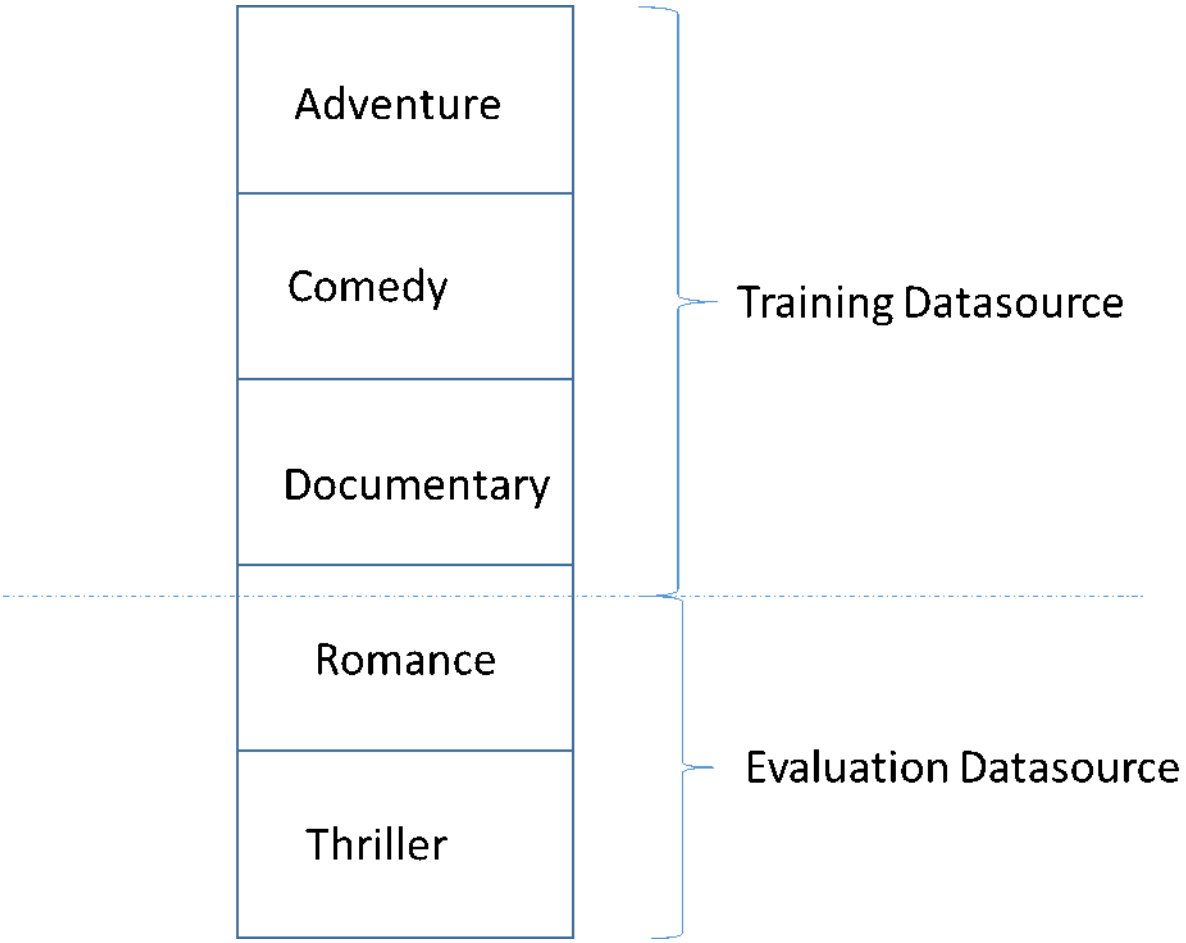
データソースを作成すると、データソースをシーケンシャルに分割することができ、Amazon ML は、トレーニングのためにデータの最初の 70% を使用し、評価のためにデータの残りの 30% を使用します。これは Amazon ML コンソールを使用してデータを分割するときのデフォルトのアプローチです。

データのランダムな分割

入力データをトレーニングデータソースと評価データソースにランダムに分割することで、トレーニングデータソースと評価データソースでデータの分布が似ていることが保証されます。入力データの順序を保持する必要がない場合は、このオプションを選択します。

Amazon ML はシード擬似乱数生成メソッドを使用してデータを分割します。シードは、一部は入力文字列値に、一部はデータ自体の内容に基づいています。デフォルトでは、Amazon ML コンソールは入力データの S3 の場所を文字列として使用します。API ユーザーはカスタム文字列を提供できます。つまり、S3 バケットとデータが同じ場合、Amazon ML は毎回同じ方法でデータを分割します。Amazon ML のデータ分割方法を変更するには、`CreateDatasourceFromS3`、`CreateDatasourceFromRedshift`、または `CreateDatasourceFromRDS` API を使用してシード文字列の値を指定します。これらの API を使用してトレーニングと評価用に別々のデータソースを作成する場合は、トレーニングデータと評価

データが重複しないように、1 つのデータソースに対してデータソースと補完フラグに同じシード文字列値を使用することが重要です。



高品質の ML モデルを開発する際の共通の落とし穴は、トレーニングに使用されたデータと似ていないデータで ML モデルを評価することです。たとえば、ML を使用して映画のジャンルを予測し、トレーニングデータには冒険、コメディ、ドキュメンタリーのジャンルの映画が含まれているとします。ただし、評価データにはロマンスとスリラーのジャンルのデータのみが含まれています。この場合、ML モデルはロマンスとスリラーのジャンルに関する情報を習得しておらず、評価はモデルがアドベンチャー、コメディ、ドキュメンタリーのジャンルのパターンをどれだけ習得したかを評価しませんでした。その結果、ジャンル情報は役に立たず、すべてのジャンルの ML モデル予測の品質が侵害されます。モデルと評価はあまりにも異なっている (記述統計が非常に異なっている) ため、役立ちません。これは、入力データがデータセット内の列の 1 つによってソートされてから、シーケンシャルに分割されるときに発生する可能性があります。

トレーニングデータソースと評価データソースのデータ分布が異なる場合、モデル評価で評価アラートが表示されます。評価アラートの詳細については、評価アラート を参照してください。

データソースの作成時に Amazon S3 で入力データをランダムにシャッフルしたり、Amazon Redshift SQL クエリの `random()` 関数または MySQL SQL クエリの `rand()` 関数を使用して入力データをランダム化している場合、Amazon ML でランダム分割を使用する必要はありません。このような場合、シーケンシャル分割オプションを使用して、同様の分布を持つトレーニングデータソースおよび評価データソースを作成することができます。

データ洞察

Amazon ML は、データを理解するために使用できる入力データに関する記述統計を計算します。

記述統計

Amazon ML はさまざまな属性タイプについて、以下の記述統計を計算します。

数値:

- 分布ヒストグラム
- 無効な値の数
- 最小値、中央値、平均値、最大値

バイナリとカテゴリ。

- カウント (カテゴリごとに異なる値のもの)
- 値分布ヒストグラム
- 最も頻繁な値
- 一意の値の数
- 真の値の割合 (バイナリのみ)
- 最も目立つワード
- 最も頻繁なワード

テキスト:

- 属性の名前
- ターゲットとの相関 (ターゲットが設定されている場合)
- 合計ワード数
- 一意のワード

- 行内のワード数の範囲
- ワードの長さの範囲
- 最も目立つワード

Amazon ML コンソールでのデータインサイトへのアクセス

Amazon ML コンソールで、任意のデータソースの名前または ID を選択して、[Data Insights] (データインサイト) ページを表示できます。このページでは、次の情報を含め、データソースに関連付けられた入力データについて学習するためのメトリクスと視覚化を提供します。

- データの要約
- 目標分布
- 欠落した値
- Invalid values (無効な値)
- データ型別の変数のサマリー統計
- データ型別の変数の分布

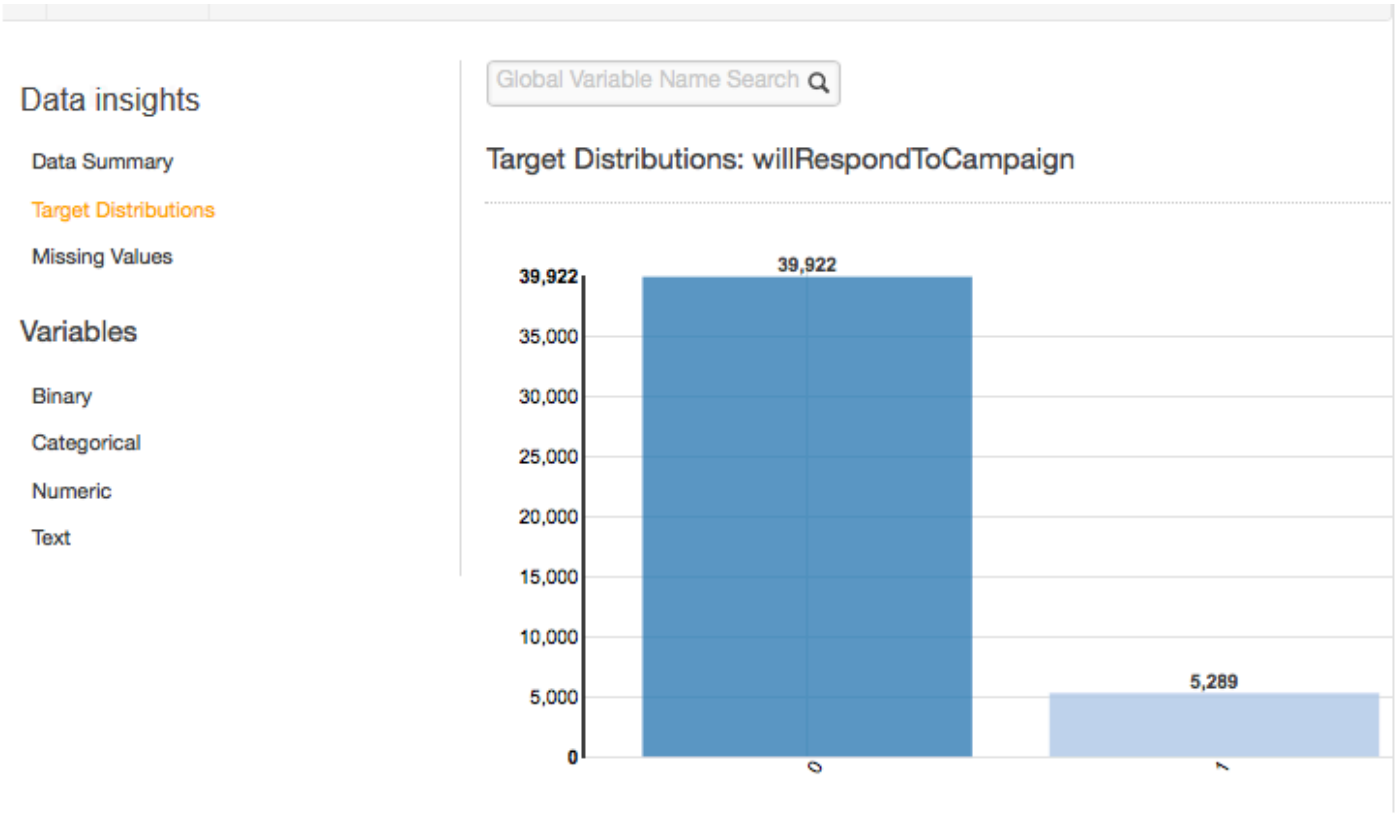
以下のセクションでは、メトリクスと可視化について詳しく説明します。

データの要約

データソースのデータ要約レポートには、データソース ID、名前、完了した場所、現在のステータス、ターゲット属性、入力データ情報 (S3 バケットの場所、データフォーマット、処理されたレコードの数および処理中に発生した不良レコードの数) およびデータ型別の変数の数を示します。

目標分布

ターゲット分布レポートには、データソースのターゲット属性の分布が表示されます。次の例では、willRespondToCampaign ターゲット属性が 0 に等しい 39,922 回の観測があります。これは、E メールキャンペーンに応答しなかった顧客の数です。willRespondToCampaign が 1 に等しい 5,289 回の観測があります。これは、E メールキャンペーンに応答した顧客の数です。



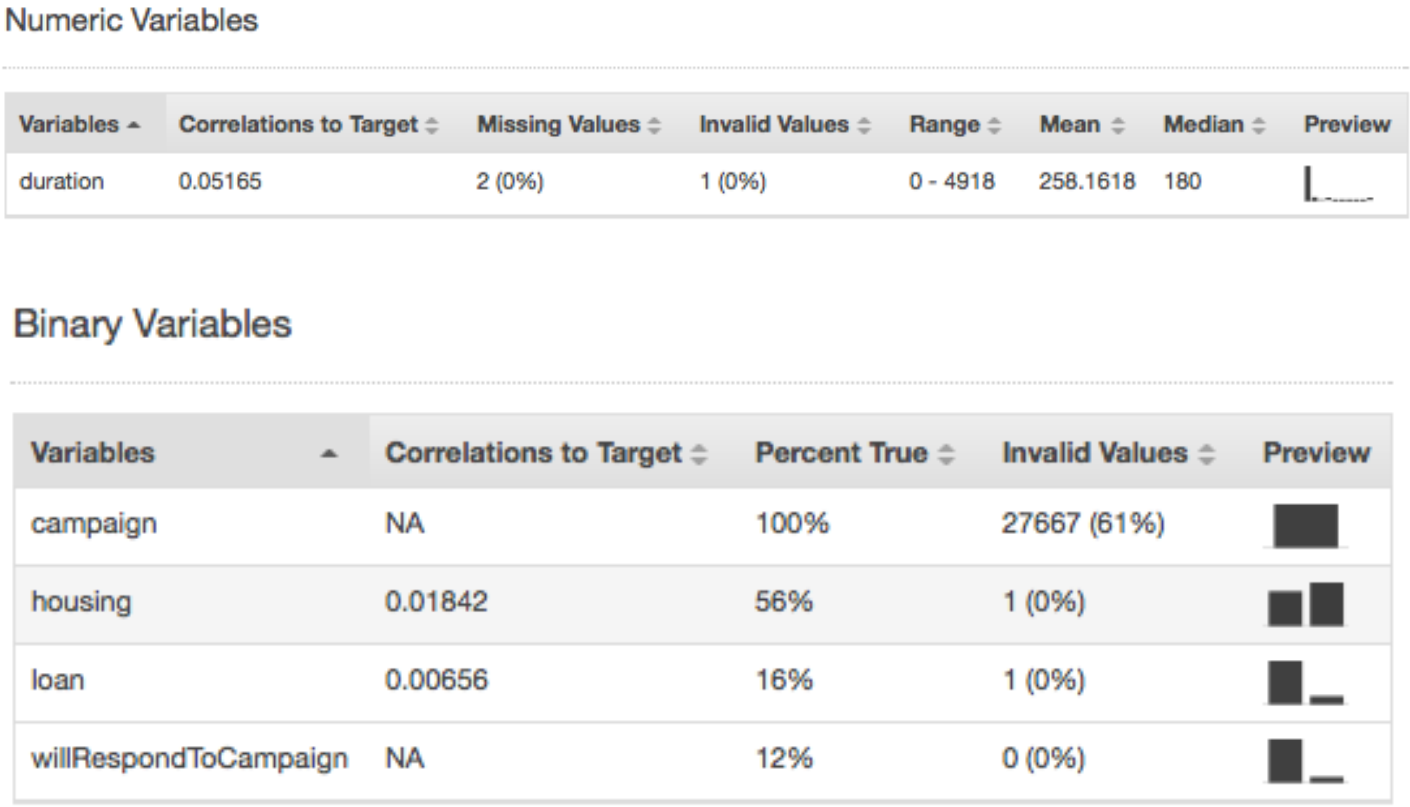
欠落した値

欠落した値のレポートには、値が欠落している入力データの属性がリストされます。数値データ型の属性のみが欠落した値を持つ可能性があります。欠落した値は ML モデルのトレーニングの質に影響する可能性があるため、可能であれば、欠落した値を提示することを推奨します。

ML モデルのトレーニング中に、ターゲット属性が見つからない場合、Amazon ML は対応するレコードを拒否します。ターゲット属性がレコードに存在するが、別の数値属性の値が欠落している場合、Amazon ML は欠落した値を見落とします。この場合、Amazon ML は代替属性を作成し、それを 1 に設定して、この属性が欠落していることを示します。これにより、Amazon ML は欠落した値の発生からパターンを学習できます。

無効な値

無効な値は、数値データ型とバイナリデータ型でのみ発生します。無効な値は、データ型レポートの変数のサマリー統計を表示して見つけることができます。次の例では、duration 数値属性に 1 つの無効な値とバイナリデータ型 (housing 属性に 1 つと loan 属性に 1 つ) に 2 つの無効な値があります。



Numeric Variables

Variables	Correlations to Target	Missing Values	Invalid Values	Range	Mean	Median	Preview
duration	0.05165	2 (0%)	1 (0%)	0 - 4918	258.1618	180	

Binary Variables

Variables	Correlations to Target	Percent True	Invalid Values	Preview
campaign	NA	100%	27667 (61%)	
housing	0.01842	56%	1 (0%)	
loan	0.00656	16%	1 (0%)	
willRespondToCampaign	NA	12%	0 (0%)	

変数とターゲットの相関

データソースを作成すると、Amazon ML はデータソースを評価し、変数とターゲットの間の相関や影響を特定できます。たとえば、製品の価格はベストセラーであるかどうかに大きな影響を及ぼす可能性がありますが、製品のディメンションには予測力がほとんどない可能性があります。

できるだけ多くの変数をトレーニングデータに含めることが一般的にはベストプラクティスです。しかし、予測力の少ない多くの変数を含めることによって導入されるノイズは、ML モデルの品質と精度に悪影響を与える可能性があります。

モデルをトレーニングするときに影響の少ない変数を削除することで、モデルの予測パフォーマンスを向上させることができます。レシピで機械学習プロセスで使用できる変数を定義することができます。これは Amazon ML の変換メカニズムです。レシピの詳細については、「機械学習のデータ変換」を参照してください。

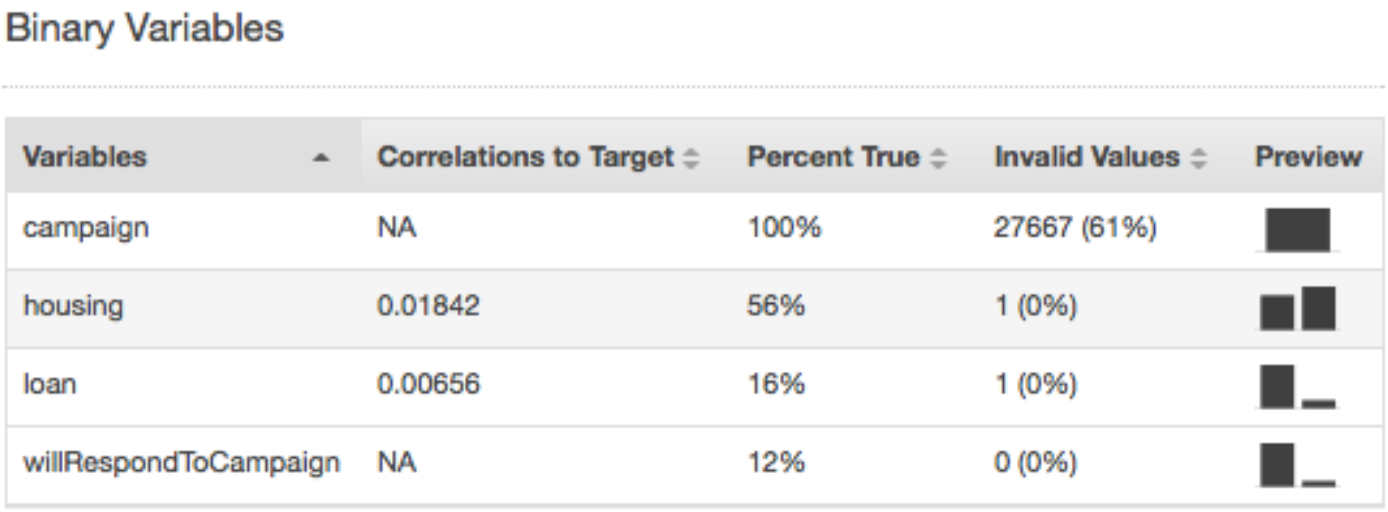
データ型別の属性のサマリー統計

データ洞察レポートでは、次のデータ型で属性サマリー統計を表示できます。

- バイナリ

- カテゴリ
- 数値
- [Text] (テキスト)

バイナリデータ型のサマリー統計には、すべてのバイナリ属性が表示されます。[ターゲットとの相関関係] 列には、ターゲット列と属性列の間で共有される情報が表示されます。[true の割合] 列には、値 1 の観測値の割合が表示されます。[Invalid values (無効な値)] 列には、無効な値の数と各属性の無効な値の割合が表示されます。[プレビュー] 列には、各属性のグラフィカルな分布へのリンクがあります。



カテゴリデータ型のサマリー統計は、一意の値の数、最も頻繁な値、および最も低い頻度の値を持つすべてのカテゴリ属性を示します。[プレビュー] 列には、各属性のグラフィカルな分布へのリンクがあります。

Categorical Variables

Variables	Correlations to Target	Unique Values	Most Frequent	Least Frequent	Preview
campaign	0.00433	49	1	39	
customerId	NA	45211	45211	1	
education	0.00355	5	secondary		
housing	0.01846	4	1		
jobId	0.00671	13	blue-collar		
willRespondToCampaign	NA	3	0		

数値データ型のサマリー統計には、欠落した値の数、無効な値、値の範囲、平均値、および中央値を含むすべての数値属性が表示されます。[プレビュー] 列には、各属性のグラフィカルな分布へのリンクがあります。

Numeric Variables

Variables	Correlations to Target	Missing Values	Invalid Values	Range	Mean	Median	Preview
duration	0.05165	2 (0%)	1 (0%)	0 - 4918	258.1618	180	

テキストデータ型のサマリー統計は、すべてのテキスト属性、属性内の単語の総数、属性内の一意のワードの数、属性内の単語の範囲、語長の範囲、および最も目立つワードを表示します。[プレビュー] 列には、各属性のグラフィカルな分布へのリンクがあります。

Text attributes

Attributes	Correlations to target *	Total words	Unique words	Words in attribute (range)	Word length (range)	Most prominent words
Phrase	0.07118	751741	12811	0 - 48	1 - 18	enters, trust ...

« ‹ 1 - 1 of 1 Attributes › »

** Correlations to Target is an approximate statistic for text attributes.*

次の例は、4 つのレコードを持つレビューというテキスト変数のテキストデータ型統計を示しています。

```
1. The fox jumped over the fence.
2. This movie is intriguing.
3.
4. Fascinating movie.
```

この例の列には、次の情報が表示されます。

- [属性] 列には、変数の名前が表示されます。この例では、この列には「レビュー」と表示されます。
- [ターゲットとの相関関係] 列は、ターゲットが指定されている場合にのみ存在します。相関は、この属性がターゲットに関して提供する情報の量を測定します。相関が高いほど、この属性はターゲットについてより多くの情報を示します。相関は、テキスト属性の簡略化された表現とターゲットとの間の相互情報の観点から測定されます。
- [合計ワード数] 列には、各レコードをトークン化して生成された単語の数が表示され、空白で区切られます。この例では、この列には「12」と表示されます。
- [一意のワード] 列には、属性の一意の単語の数が表示されます。この例では、この列には「10」と表示されます。
- [属性内のワード (範囲)] 列には、属性内の 1 行の単語数が表示されます。この例では、この列には「0 ~ 6」と表示されます。
- [ワードの長さ (範囲)] 列には、単語に含まれる文字の数の範囲が示されます。この例では、この列には「2 ~ 11」と表示されます。
- [最も目立つワード] 列には、属性に表示される単語のランク付けされたリストが表示されます。ターゲット属性がある場合、単語はターゲットとの相関によってランク付けされます。つまり、相関が最も高い単語が最初にリストされます。データ内にターゲットが存在しない場合、単語はそのエントロピーによってランク付けされます。

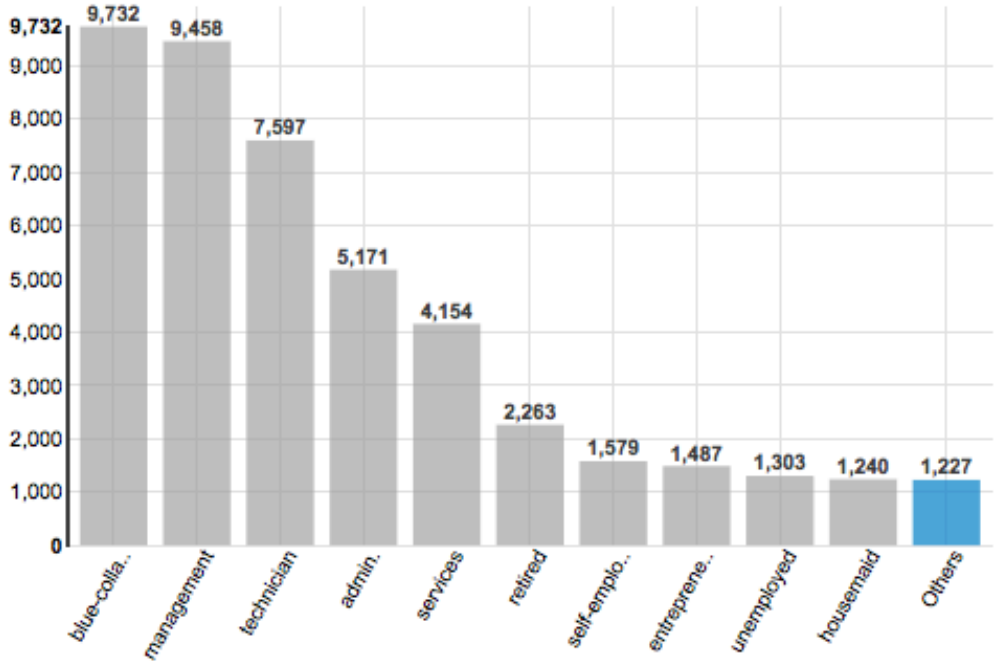
カテゴリ属性とバイナリ属性の分布を理解する

カテゴリまたはバイナリ属性に関連付けられた [プレビュー] リンクをクリックすると、その属性の分布および属性の各カテゴリ値に対する入力ファイルのサンプルデータを表示できます。

たとえば、次のスクリーンショットは、カテゴリ属性 jobId の分布を示しています。分布には上位 10 個のカテゴリ値が表示され、他のすべての値は「その他」にグループ化されています。その値を含む入力ファイル内の観測数と、入力データファイルからのサンプル観測を表示するためのリンクとの上位 10 個のカテゴリ値をランク付けします。

Categorical Variables: jobId

Top 10 jobId



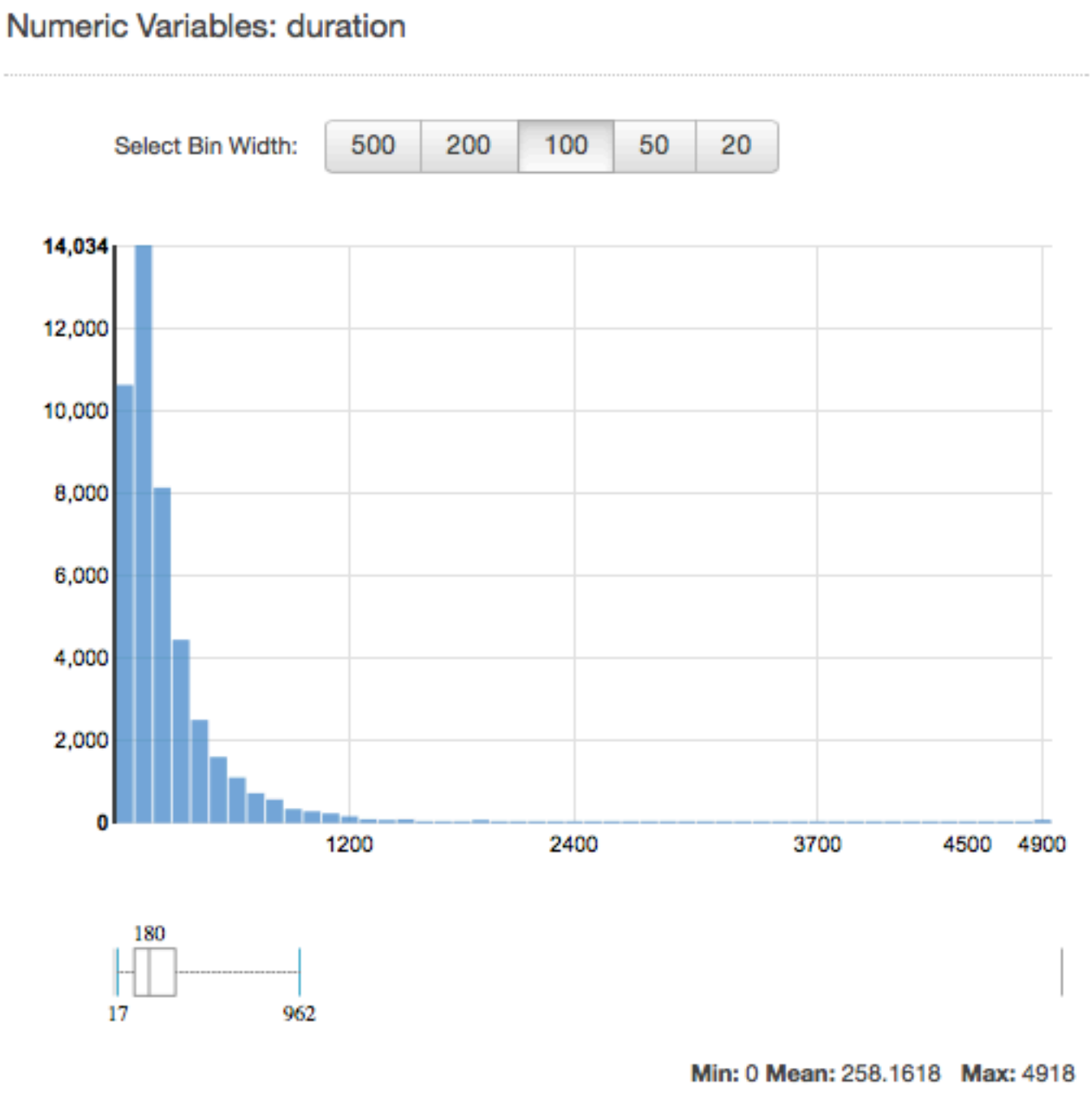
All Categories

Ranking	Category	Count	
1	blue-collar	9732	Sample data
2	management	9458	Sample data
3	technician	7597	Sample data

数値属性の分布について

数値属性の分布を表示するには、属性の [プレビュー] リンクをクリックします。数値属性の分布を表示する際には、500、200、100、50、20 のビンサイズを選択できます。ビンサイズが大きいほど、表示される棒グラフの数は少なくなります。さらに、大きなビンサイズの場合、分布の解像度は粗くなります。反対に、バケットサイズを 20 に設定すると、表示される分布の解像度が向上します。

次のスクリーンショットに示すように、最小値、平均値、最大値も表示されます。



テキスト属性の分布について

テキスト属性の分布を表示するには、属性の [プレビュー] リンクをクリックします。テキスト属性の分布を見ると、次の情報が表示されます。

Text attributes: Phrase

Ranking	Token	Word prominence	Count	
1	enters	0.01105	7	0.0%
2	trust	0.00884	28	0.0%
3	bad	0.00735	833	0.2%
4	film	0.00669	4747	1.3%
5	movie	0.00611	4242	1.2%
6	unwieldy	0.00605	11	0.0%
7	good	0.00574	1620	0.5%
8	ashamed	0.00551	7	0.0%
9	funny	0.00550	1078	0.3%
10	wankery	0.00498	9	0.0%

« ‹ 1 - 10 of 11091 › »

ランキング

テキストトークンは、伝達する情報の量によってランク付けされ、最も有益なものから最も有益なものに分類されます。

Token

トークンは、入力テキストから統計行が表示されている単語を表示します。

ワードプロミネンス

ターゲット属性がある場合、単語はターゲットとの相関によってランク付けされ、相関が最も高いワードが最初にリストされます。データにターゲットが存在しない場合、単語はエントロピー、すなわち通信可能な情報の量によってランク付けされます。

カウント数

カウント数は、トークンが表示した入力レコードの数を示します。

カウント割合

カウント割合は、トークンが表示された入力データ行の割合を示します。

Amazon ML での Amazon S3 の使用

Amazon Simple Storage Service (Amazon S3) は、インターネット用のストレージです。Simple Storage Service (Amazon S3) を使用すると、いつでもウェブ上の任意の場所から任意の量のデータを保存および取得できます。Amazon ML はプライマリデータのリポジトリとして以下のタスクで Amazon S3 を使用します。

- 入力ファイルにアクセスして、ML モデルのトレーニングおよび評価用のデータソースオブジェクトを作成する。
- 入力ファイルにアクセスしてバッチ予測を生成する。
- ML モデルを使用してバッチ予測を生成する際、指定した S3 バケットに予測ファイルを出力する。
- Amazon Redshift または Amazon Relational Database Service (Amazon RDS) に保存したデータを .csv ファイルにコピーして、Amazon S3 にアップロードする。

Amazon ML がこれらのタスクを実行するには、Amazon S3 データにアクセスするためのアクセス権限を Amazon ML に付与する必要があります。

> **Note**
>
> サーバー側の暗号化ファイルのみを受け入れる S3 バケットにバッチ予測ファイルを出力することはできません。リクエストに `Deny` ヘッダーがない場合、ポリシーに `s3:PutObject` アクションの `s3:x-amz-server-side-encryption` 効果が含まれていないことを確認することにより、バケットポリシーが暗号化されていないファイルのアップロードを許可することを確認してください。S3 サーバー側の暗号化バケットポリシーの詳細については、「Amazon Simple Storage Service ユーザーガイド」の「サーバー側の暗号化を使用したデータの保護」を参照してください。

Amazon S3 へのデータのアップロード

Amazon ML は Amazon S3の場所からデータを読み取るため、Amazon Simple Storage Service (Amazon S3) に入力データをアップロードする必要があります。データを Amazon S3 に直接アッ

プロードする (例えば、コンピュータなどから)、または、Amazon ML が Amazon Redshift または Amazon Relational Database Service (RDS) に保存してあるデータを .csv ファイルにコピーして Amazon S3 にアップロードすることができます。

Amazon Redshift または Amazon RDS からのデータのコピーの詳細については、「Amazon ML での Amazon Redshift の使用」または「Amazon ML での Amazon RDS の使用」をそれぞれ参照してください。

このセクションの残りの部分では、入力データをコンピュータから直接 Amazon S3 にアップロードする方法について説明します。このセクションの手順を開始する前に、データを .csv ファイルで持っている必要があります。Amazon ML で使用できるように .csv ファイルを正しくフォーマットする方法の詳細については、「Amazon ML のデータ形式を理解する」を参照してください。

データをコンピュータから Amazon S3 にアップロードするには

1. AWS マネジメントコンソールにサインインして Amazon S3 コンソール (https://console.aws.amazon.com/s3/) を開きます。
2. バケットを作成するか、既存のバケットを選択します。
   a. バケットを作成するには、[バケットの作成] を選択します。バケットに名前を付け、リージョンを選択 (使用可能な任意のリージョンを選択) した後、[作成] を選択します。詳細については、Amazon Simple Storage 入門ガイドの「バケットの作成」を参照してください。
   b. 既存のバケットを使用するには、[すべてのバケット] のリストでバケットを選択することによりバケットを検索します。バケット名が表示されたら、それを選択して、[アップロード] を選択します。
3. [アップロード] ダイアログボックスで、[ファイルを追加] を選択します。
4. 入力データの .csv ファイルを含むフォルダに移動し、[開く] を選択します。

許可

Amazon ML がいずれかの S3 バケットにアクセスするためのアクセス権限を付与するため、バケットポリシーを編集する必要があります。

Amazon ML に Amazon S3 内のバケットからデータを読み込むためのアクセス権限を付与することの詳細については、「Amazon S3 からデータを読み込むためのアクセス許可を Amazon ML に付与する」を参照してください。

Amazon ML に Amazon S3 内のバケットへバッチ予測結果を出力するためのアクセス許可の付与の詳細については、「Amazon S3 に予測を出力するためのアクセス許可を Amazon ML に付与する」を参照してください。

Amazon S3 リソースへのアクセス権限の管理の詳細については、「Amazon S3 開発者ガイド」を参照してください。

Amazon Redshift のデータから Amazon ML データソースを作成する

Amazon Redshift に保存したデータがある場合は、Amazon Machine Learning (Amazon ML) コンソールの [Create Datasource] (データソースの作成) ウィザードを使用して、データソースオブジェクトを作成できます。Amazon Redshift データからデータソースを作成する場合は、データを含むクラスター、および、データを取得する SQL クエリを指定します。クラスターの Amazon Redshift `Unload` コマンドを呼び出すことで Amazon ML はクエリを実行します。Amazon ML は、選択した Amazon Simple Storage Service (Amazon S3) の場所に結果を保存し、Amazon S3 に保存されているデータを使用してデータソースを作成します。データソース、Amazon Redshift クラスター、および S3 バケットはすべて同じリージョンにある必要があります。

> **Note**
>
> Amazon ML はプライベート VPC で Amazon Redshift クラスターからのデータソースの作成をサポートしていません。クラスターにはパブリック IP アドレスが必要です。

トピック

- データソースの作成ウィザードに必要なパラメータ
- Amazon Redshift データ (コンソール) でデータソースを作成する
- Amazon Redshift の問題のトラブルシューティング

データソースの作成ウィザードに必要なパラメータ

Amazon ML が Amazon Redshift データベースに接続し、ユーザーに代わってデータを読み込むのを許可するには、以下を提供する必要があります。

- Amazon Redshift `ClusterIdentifier`
- Amazon Redshift データベースの名前

- Amazon Redshift データベースの認証情報 (ユーザー名とパスワード)
- Amazon ML Amazon Redshift AWS Identity and Access Management (IAM) ロール
- Amazon Redshift SQL クエリ
- (オプション) Amazon ML スキーマの場所
- Amazon S3 のステージング場所 (Amazon ML がデータソースを作成する前にデータを配置した所)

さらに、Amazon Redshift データソースを作成する IAM ユーザーまたはロール (コンソールによるか、または、`CreateDatasourceFromRedshift` アクションを使用するかにかかわらず) が、`iam:PassRole` 権限を持っていることを確認する必要があります。

Amazon Redshift**`ClusterIdentifier`**

この大文字と小文字を区別するパラメータを使用して、Amazon ML がクラスターを検索して接続できるようにします。Amazon Redshift コンソールからクラスター識別子 (名前) を取得できます。クラスターの詳細については、「Amazon Redshift クラスター」を参照してください。

Amazon Redshift データベースの名前

このパラメータを使用して、Amazon Redshift クラスターのどのデータベースにデータソースとして使用するデータが含まれているかを Amazon ML に伝えます。

Amazon Redshift データベース認証情報

これらのパラメータを使用して、セキュリティクエリが実行されるコンテキスト内の Amazon Redshift データベースユーザーのユーザー名とパスワードを指定します。

> **Note**
>
> Amazon ML では、Amazon Redshift データベースに接続するために、Amazon Redshift ユーザー名とパスワードが必要です。データを Amazon S3 にアンロードした後に、Amazon ML がパスワードを再利用する、または、パスワードを保存することはありません。

Amazon ML Amazon Redshift ロール

このパラメータを使用して、Amazon Redshift クラスターのセキュリティグループと Amazon S3 のステージング場所のバケットポリシーを設定するために Amazon ML が使用する IAM ロールの名前を指定します。

Amazon Redshift にアクセスできる IAM ロールがない場合は、Amazon ML がロールを作成できます。Amazon ML がロールを作成すると、顧客管理ポリシーが作成され、IAM ロールにアタッチされます。Amazon ML が作成するポリシーは、指定したクラスターのみにアクセスするためのアクセス権限を Amazon ML に付与します。

Amazon Redshift にアクセスするための IAM ロールがすでにある場合は、ロールの ARN を入力するか、ドロップダウンリストからロールを選択できます。Amazon Redshift アクセス権限を持つ IAM ロールはドロップダウンの上部に表示されます。

IAM ロールには、次の内容が必要です。

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "Service": "machinelearning.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringEquals": { "aws:SourceAccount": "123456789012" },
           "ArnLike": { "aws:SourceArn": "arn:aws:machinelearning:us-
east-1:123456789012:datasource/*" }
        }
    }]
}
```

顧客管理ポリシーの詳細については、「IAM ユーザーガイド」の「顧客管理ポリシー」を参照してください。

Amazon Redshift SQL クエリ

このパラメータを使用して、データを選択するために Amazon Redshift データベースで Amazon ML が実行する SQL SELECT クエリを指定します。Amazon ML は、Amazon Redshift の UNLOAD アクションを使用して、クエリの結果を Amazon S3 の場所に安全にコピーします。

> **Note**
>
> Amazon ML は、入力レコードがランダムな順序 (シャッフル) のときに最も効果的です。Amazon Redshift random() 関数を使用すると、Amazon Redshift SQL クエリの結果を簡単にシャッフルできます。たとえば、これが元のクエリであるとします。

```
"SELECT col1, col2, … FROM training_table"
```

次のようにクエリを更新することでランダムなシャッフルを埋め込むことができます。

```
"SELECT col1, col2, … FROM training_table ORDER BY random()"
```

スキーマの場所 (オプション)

このパラメータを使用して、Amazon ML がエクスポートする Amazon Redshift データのスキーマへの Amazon S3 パスを指定します。

データソースのスキーマを指定しない場合、Amazon ML コンソール は Amazon Redshift SQL クエリのデータスキーマに基づいて Amazon ML スキーマを自動的に作成します。Amazon ML スキーマは データ型が Amazon Redshift スキーマより少ないため、1 対 1 の変換ではありません。Amazon ML コンソールは、次の変換スキームを使用して Amazon Redshift データ型を Amazon ML データ型に変換します。

Amazon Redshift のデータ型	Amazon Redshift エイリアス	Amazon ML のデータ型
SMALLINT	INT2	NUMERIC
INTEGER	INT、INT4	NUMERIC
BIGINT	INT8	NUMERIC
DECIMAL	NUMERIC	NUMERIC
REAL	FLOAT4	NUMERIC
DOUBLE PRECISION	FLOAT8、FLOAT	NUMERIC
BOOLEAN	BOOL	BINARY
CHAR	CHARACTER、NCHAR、BPCHAR	CATEGORICAL
VARCHAR	CHARACTER VARYING、NVARCHAR、TEXT	TEXT

Amazon Redshift のデータ型	Amazon Redshift エイリアス	Amazon ML のデータ型
DATE		TEXT
TIMESTAMP	TIMESTAMP WITHOUT TIME ZONE	TEXT

Amazon ML `Binary` データ型に変換するには、データ内の Amazon Redshift ブール値が Amazon ML バイナリ値をサポートしている必要があります。ブール値のデータ型が、サポートされていない値を持つ場合は、Amazon ML はそれをできるだけ具体的なデータ型に変換します。例えば、Amazon Redshift のブール値が `0`、`1`、および `2` である場合、Amazon ML はブール値を `Numeric` データ型に変換します。サポートされているバイナリ値の詳細については、「AttributeType フィールドの使用」を参照してください。

Amazon ML がデータ型を判別できない場合は、デフォルトの `Text` となります。

Amazon ML がスキーマを変換したら、割り当てられた Amazon ML のデータ型をデータソース作成ウィザードで確認し修正でき、Amazon ML がデータソースを作成する前にスキーマを変更できます。

Amazon S3 のステージング場所

このパラメータを使用して、Amazon ML が Amazon Redshift SQL クエリの結果を保存している Amazon S3 のステージング場所の名前を指定します。データソースを作成した後、Amazon ML はデータを Amazon Redshift に返すのではなく、ステージング場所で使用します。

> **Note**
>
> Amazon ML は Amazon ML Amazon Redshift ロールによって定義された IAM ロールを引き受けるため、Amazon ML は指定された Amazon S3 ステージング場所のすべてのオブジェクトにアクセスできるアクセス許可を持ちます。このため、Amazon S3 のステージング場所には、機密情報が含まれていないファイルのみを保存するようお勧めします。例えば、ルートバケットが `s3://mybucket/` の場合、 Amazon ML がアクセスするファイルだけを保存するための場所を作成することをお勧めします (例: `s3://mybucket/AmazonMLInput/`)。

Amazon Redshift データ (コンソール) でデータソースを作成する

Amazon ML コンソールには、Amazon Redshift データを使用してデータソースを作成する 2 つの方法が用意されています。データソース作成ウィザードを完了して、データソースを作成することも、Amazon Redshift データから作成したデータソースがすでにあれば、元のデータソースをコピーして設定を変更することもできます。データソースをコピーすれば、類似のデータソースを複数作成することが簡単にできます。

API を使用したデータソースの作成の詳細については、「CreateDataSourceFromRedshift」を参照してください。

以下の手順のパラメータの詳細については、データソースの作成ウィザードに必要なパラメータ を参照してください。

トピック

- データソースの作成 (コンソール)
- データソースのコピー (コンソール)

データソースの作成 (コンソール)

Amazon Redshift から Amazon ML データソースへデータをアンロードするには、データソース作成ウィザードを使用します。

Amazon Redshift のデータからデータソースを作成するには

1. Amazon Machine Learning コンソール (https://console.aws.amazon.com/machinelearning/) を開きます。
2. Amazon ML ダッシュボードの [Entities] (エンティティ) の下で、[Create new...] (新規作成...)、[Datasource] (データソース) の順に選択します。
3. [Input data] (入力データ) ページで、[Amazon Redshift] を選択します。
4. データソース作成ウィザードで、[クラスター識別子] にクラスターの名前を入力します。
5. [Database name] (データベース名) には、Amazon Redshift データベースの名前を入力します。
6. [データベースユーザー名] にデータベースのユーザー名を入力します。
7. [データベースパスワード] にデータベースのパスワードを入力します。
8. [IAM ロール] で、IAM ロールを選択します。まだロールがない場合は、[Create a new role] (新しいロールの作成) を選択します。Amazon ML によって自動的に IAM Amazon Redshift ロールが作成されます。

9. Amazon Redshift の設定をテストするには、[Test Access] (アクセスのテスト) ([IAM role] (IAM ロール) の横) を選択します。提供された設定で Amazon ML が Amazon Redshift に接続できない場合は、データソースの作成を継続できません。トラブルシューティングヘルプについては、エラーのトラブルシューティング を参照してください。
10. [SQL クエリ] には、SQL クエリを入力します。
11. [Schema location] (スキーマの場所) では、Amazon ML がスキーマを作成するかどうかを選択します。スキーマを自分で作成した場合は、スキーマファイルに Amazon S3 パスを入力します。
12. [Amazon S3 staging location] (Amazon S3 ステージング場所) には、Amazon ML が Amazon Redshift からアンロードしたデータを配置するバケットへの Amazon S3 パスを入力します。
13. (オプション) [データソース名] には、データソースの名前を入力します。
14. [検証] を選択します。Amazon ML は Amazon Redshift データベースに接続できることを確認します。
15. [スキーマ] ページで、すべての属性のデータ型を確認し、必要に応じて修正します。
16. [Continue] (続行) をクリックします。
17. このデータソースを使用して ML モデルを作成または評価する場合は、[Do you plan to use this dataset to create or evaluate an ML model? (このデータセットを使用して ML モデルを作成または評価することを計画しますか?)] で [Yes (はい)] を選択します。[Yes (はい)] を選択した場合は、ターゲット行を選択します。ターゲットの詳細については、targetAttributeName フィールドの使用 を参照してください。

    予測を作成するために既に作成したモデルと共にこのデータソースを使用する場合は、[No (いいえ)] を選択します。
18. [Continue] (続行) をクリックします。
19. [Does your data contain an identifier? (データには識別子が含まれていますか。)] で、データに行の識別子が含まれていなければ、[No (いいえ)] を選択します。

    データに行の識別子が含まれていれば、[Yes (はい)] を選択します。行の識別子の詳細については、rowID フィールドの使用 を参照してください。
20. [Review] (レビュー) を選択します。
21. [レビュー] ページで、設定を確認し、[完了] を選択します。

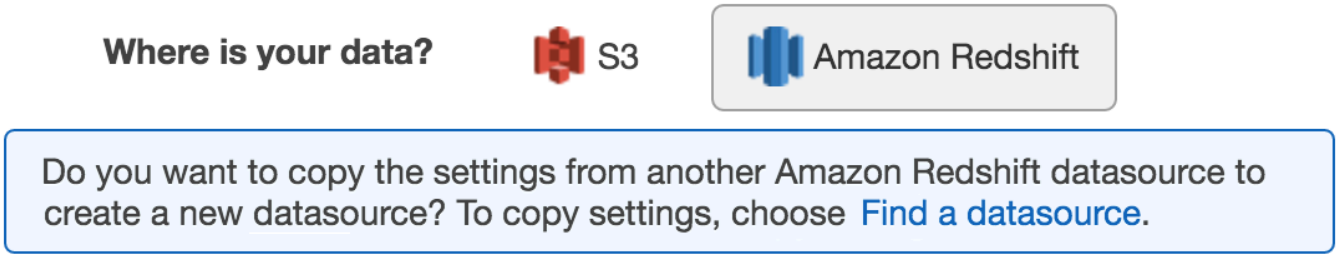
データソースを作成した後、create an ML model に使用できます。すでにモデルを作成している場合は、データソースを evaluate an ML model または generate predictions に使用できます。

データソースのコピー (コンソール)

既存のデータソースに似たデータソースを作成する場合は、Amazon ML コンソールを使用して元のデータソースをコピーし、設定を変更できます。例えば、既存のデータソースから始めて、データスキーマを変更してデータをより一致させる、Amazon Redshift からデータをアンロードするのに使用する SQL クエリを変更する、または、Amazon Redshift クラスターにアクセスする別の AWS Identity and Access Management (IAM) ユーザーを指定することができます。

Amazon Redshift データソースをコピーして変更するには

1. Amazon Machine Learning コンソール (https://console.aws.amazon.com/machinelearning/) を開きます。
2. Amazon ML ダッシュボードの [Entities] (エンティティ) の下で、[Create new...] (新規作成...)、[Datasource] (データソース) の順に選択します。
3. [Input data] (入力データ) ページの、[Where is your data?] (データの場所) で、[Amazon Redshift] を選択します。Amazon Redshift データから作成されたデータソースがすでにある場合は、別のデータソースから設定をコピーすることを選択できます。



   Amazon Redshift データから作成されたデータソースを持っていない場合は、このオプションは表示されません。
4. [Find a datasource (データソースの検索)] を選択します。
5. コピーするデータソースを選択し、[Copy settings] (設定のコピー) を選択します。Amazon ML が、データソース設定のほとんどに、元のデータソースからの設定を自動入力します。データベースパスワード、スキーマの場所、データソース名は元のデータソースからコピーされません。
6. 必要に応じて、自動入力された設定を変更します。例えば、Amazon ML が Amazon Redshift からアンロードするデータを変更する場合は、SQL クエリを変更します。
7. [データベースパスワード] にデータベースのパスワードを入力します。Amazon ML はパスワードの保存や再利用をしないため、常に自分で入力する必要があります。

8. (オプション) [Schema location] (スキーマの場所) では、Amazon MLによって [I want Amazon ML to generate a recommended schema] (Amazon ML が推奨スキーマを生成する) が前もって選択されています。スキーマをすでに作成している場合は、[I want to use the schema that I created and stored in Amazon S3] (Amazon S3 で作成して保存したスキーマを使用する) を選択し、Amazon S3 のスキーマファイルへのパスを入力します。
9. (オプション) [データソース名] には、データソースの名前を入力します。それ以外の場合は、Amazon ML が新しいデータソース名を生成します。
10. [検証] を選択します。Amazon ML は Amazon Redshift データベースに接続できることを確認します。
11. (オプション) Amazon ML が [Schema] (スキーマ) ページでスキーマを推定した場合は、スキーマのすべての属性のデータ型を確認し、必要に応じて修正します。
12. [Continue] (続行) をクリックします。
13. このデータソースを使用して ML モデルを作成または評価する場合は、[Do you plan to use this dataset to create or evaluate an ML model? (このデータセットを使用して ML モデルを作成または評価することを計画しますか?)] で [Yes (はい)] を選択します。[Yes (はい)] を選択した場合は、ターゲット行を選択します。ターゲットの詳細については、targetAttributeName フィールドの使用 を参照してください。

    予測を作成するために既に作成したモデルと共にこのデータソースを使用する場合は、[No (いいえ)] を選択します。
14. [Continue] (続行) をクリックします。
15. [Does your data contain an identifier? (データには識別子が含まれていますか。)] で、データに行の識別子が含まれていなければ、[No (いいえ)] を選択します。

    行の識別子がデータに含まれている場合は、[Yes (はい)] を選択し、識別子として使用する行を選択します。行の識別子の詳細については、rowID フィールドの使用 を参照してください。
16. [Review] (レビュー) を選択します。
17. 設定を確認し、[完了] を選択します。

データソースを作成した後、create an ML model に使用できます。すでにモデルを作成している場合は、データソースを evaluate an ML model または generate predictions に使用できます。

Amazon Redshift の問題のトラブルシューティング

Amazon Redshift データソース、ML モデル、および評価を作成すると、Amazon Machine Learning (Amazon ML) は Amazon ML コンソールで Amazon ML オブジェクトのステータスをレポートしま

す。Amazon ML がエラーメッセージを返す場合は、次の情報とリソースを使用して問題のトラブルシューティングを行います。

Amazon ML に関する一般的な質問への回答は、「Amazon Machine Learning のよくある質問」を参照してください。また、Amazon Machine Learning フォーラムで回答を検索したり、質問を投稿したりすることもできます。

トピック

- エラーのトラブルシューティング
- AWS Support へのお問い合わせ

エラーのトラブルシューティング

ロールの形式が無効です。有効な IAM ロールを指定します。たとえば、arn:aws:iam::YourAccountID:role/YourRedshiftRole とします。

原因

IAM ロールの Amazon リソースネーム (ARN) の形式が正しくありません。

解決策

データソース作成ウィザードで、ロールに合わせて ARN を修正します。ロール ARN のフォーマットの詳細については、「IAM ユーザーガイド」の「IAM ARN」を参照してください。リージョンは、IAM ロール ARN の場合はオプションです。

ロールが無効です。Amazon ML は <role ARN> IAM ロールを引き受けることはできません。有効な IAM ロールを提供し、Amazon ML からアクセスできるようにします。

原因

ロールは、Amazon ML がそれを引き受けることを許可するように設定されていません。

解決策

IAM コンソールで、ロールを編集して、Amazon ML がアタッチされたロールを引き受けることを許可する信頼ポリシーを持つようにします。

この <user ARN> ユーザーには <role ARN> IAM ロール渡す権限がありません。

原因

IAM ユーザーには、Amazon ML にロールを渡すことを許可するアクセス権限ポリシーはありません。

解決策

IAM ユーザーに許可ポリシーを添付して、Amazon ML にロールを渡すことができます。IAM コンソールの IAM ユーザーにアクセス権限ポリシーをアタッチすることができます。

アカウント間で IAM ロールを渡すことはできません。IAM ロールはこのアカウントに属している必要があります。

原因

別の IAM アカウントに属しているロールを渡すことはできません。

解決策

ロールの作成に使用した AWS アカウントにサインインします。IAM コンソールで IAM の役割を確認できます。

定されたロールに操作を実行する権限がありません。Amazon ML に必要なアクセス権限を提供するポリシーを持つロールを提供します。

原因

IAM ロールには、要求された操作を実行する権限がありません。

解決策

IAM コンソールでロールに添付されている権限ポリシーを編集して、必要な権限を与えます。

Amazon ML は、指定された IAM ロールを持つ Amazon Redshift クラスター上のセキュリティグループを設定することはできません。

原因

IAM ロールには、Amazon Redshift セキュリティクラスターの設定に必要な権限がありません。

解決策

IAM コンソールでロールに添付されている権限ポリシーを編集して、必要な権限を与えます。

Amazon ML がクラスター上のセキュリティグループを設定しようとしたときにエラーが発生しました。あとでもう一度試してみてください。

原因

Amazon ML が Amazon Redshift クラスターに接続しようとしたときに問題が発生しました。

解決策

Create Datasource ウィザードで指定した IAM ロールに、必要な権限がすべて含まれていることを確認します。

クラスター ID の形式が無効です。クラスター ID はアルファベット文字で始まり、アルファベット文字とハイフンのみでなければなりません。ハイフンを、2 つ続けて使用したり、文字列の最後に使用したりすることはできません。

原因

Amazon Redshift クラスター ID 形式が正しくありません。

解決策

データソース作成ウィザードでは、クラスター ID に英数字とハイフンのみが含まれ、2 つの連続するハイフンまたはハイフンで終わらないように修正します。

<Amazon Redshift クラスター名> クラスターが存在しないか、クラスターが Amazon ML サービスと同じリージョンに存在しません。この Amazon ML と同じリージョンにクラスターを指定してください。

原因

Amazon ML データソースを作成しているリージョンに Amazon Redshift クラスターは存在しないため、Amazon ML はそれを見つけられません。

解決策

Amazon Redshift コンソールのクラスターページにクラスターが存在し、Amazon Redshift クラスターがあるリージョンと同じリージョンにデータソースを作成していることと、データソース作成ウィザードで指定したクラスター ID が正しいことを確認してください。

Amazon ML は Amazon Redshift クラスター内のデータを読み取ることができません。正しい Amazon Redshift クラスター ID を指定します。

原因

Amazon ML は、ユーザーが指定した Amazon Redshift クラスター内のデータを読み取ることができません。

解決策

データソース作成ウィザードで、正しい Amazon Redshift クラスター ID を指定し、Amazon Redshift クラスターがあるのと同じリージョン内にデータソースを作成していることを確認し、Amazon Redshift の [Clusters] (クラスター) ページにクラスターが一覧表示されていることを確認します。

<Amazon Redshift クラスター名> クラスターはパブリックアクセス可能ではありません。

原因

クラスターはパブリックアクセス可能ではなく、パブリック IP アドレスがないため、Amazon ML はクラスターにアクセスできません。

解決策

クラスターに公開してアクセス可能にし、パブリック IP アドレスを与えます。クラスターをパブリックアクセス可能にする方法については、「Amazon Redshift 管理ガイド」の「クラスターの変更」を参照してください。

<Redshift> クラスターステータスは、Amazon ML では利用できません。Amazon Redshift コンソールを使用して、このクラスターステータスの問題を表示および解決してください。クラスターステータスは「利用可能」でなければなりません。

原因

Amazon ML はクラスターステータスを見ることができません。

解決策

クラスターが利用可能であることを確認します。クラスターステータスの確認については、「Amazon Redshift 管理ガイド」の「クラスターステータスの概要の取得」を参照してください。クラスターを再起動して使用できるようにする方法については、「Amazon Redshift 管理ガイド」の「クラスターの再起動」を参照してください。

このクラスターには、<データベース名> データベースはありません。データベース名が正しいことを確認するか、別のクラスターおよびデータベースを指定してください。

原因

Amazon ML は、指定されたクラスター内の指定されたデータベースを見つけることができません。

解決策

データソース作成ウィザードで入力したデータベース名が正しいことを確認するか、正しいクラスターとデータベース名を指定してください。

Amazon ML はデータベースにアクセスできませんでした。データベースユーザー <ユーザー名> に有効なパスワードを入力します。

原因

Amazon ML が Amazon Redshift データベースにアクセスできるようにデータソース作成ウィザードで指定したパスワードが正しくありません。

解決策

Amazon Redshift データベースユーザーに正しいパスワードを入力してください。

Amazon ML がクエリの検証を試みたときにエラーが発生しました。

原因

SQL クエリに問題があります。

解決策

クエリが有効な SQL であることを確認します。

SQL クエリの実行中にエラーが発生しました。データベース名と指定されたクエリを確認してください。根本原因: {serverMessage}。

原因

Amazon Redshift はクエリを実行できませんでした。

解決策

データソース作成ウィザードで正しいデータベース名を指定し、クエリが有効な SQL であることを確認します。

SQL クエリの実行中にエラーが発生しました。根本原因: {serverMessage}。

原因

Amazon Redshift は指定されたテーブルを見つけることができませんでした。

解決策

データソース作成ウィザードで指定したテーブルが Amazon Redshift クラスターデータベースに存在し、正しいクラスター ID、データベース名、および SQL クエリを入力したことを確認します。

AWS Support へのお問い合わせ

AWS Premium Support を契約している場合は、AWS Support Center で技術サポートケースを作成できます。

Amazon RDS データベースのデータを使用して Amazon ML データソースを作成する

Amazon ML では、Amazon Relational Database Service (Amazon RDS) の MySQL データベースに格納されたデータからデータソースオブジェクトを作成できます。このアクションを実行すると、Amazon ML は指定した SQL クエリを実行する AWS Data Pipeline オブジェクトを作成し、その出力を任意の S3 バケットに配置します。Amazon ML はそのデータを使ってデータソースを作成します。

> **Note**
>
> Amazon ML は、VPC 内の MySQL データベースのみをサポートします。

Amazon ML が入力データを読み取れるようにするには、そのデータを Amazon Simple Storage Service (Amazon S3) にエクスポートしておく必要があります。API を使用して Amazon ML がエクスポートを実行するように設定することができます。(RDS は API に限定されており、コンソールからは利用できません)。

Amazon ML が Amazon RDS で MySQL データベースに接続し、ユーザーに代わってデータを読み込むためには、以下を提供する必要があります。

- RDS DB インスタンス識別子
- MySQL データベース名
- データパイプラインの作成、アクティブ化、および実行に使用される AWS Identity and Access Management (IAM) ロール
- データベースユーザー認証情報
  - [User name] (ユーザー名)

  - [パスワード]
- AWS Data Pipeline セキュリティ情報
  - IAM リソースロール
  - IAM サービスロール
- Amazon RDS セキュリティ情報
  - サブネット ID
  - セキュリティグループ ID
- データソースの作成に使用するデータを指定する SQL クエリ
- クエリの結果を格納するために使用される S3 出力の場所 (バケット)
- (オプション) データスキーマファイルの場所

さらに、CreateDataSourceFromRDS オペレーションを使用して Amazon RDS データソースを作成する IAM ユーザーまたはロールが `iam:PassRole` 権限を持っていることを確認する必要があります。詳細については、「IAM による Amazon ML リソースへのアクセスの制御」を参照してください。

トピック

- RDS データベースインスタンス識別子
- MySQL データベース名
- データベースユーザー認証情報
- AWS Data Pipeline セキュリティ情報
- Amazon RDS セキュリティ情報
- MySQL SQL クエリ
- S3 出力の場所

RDS データベースインスタンス識別子

RDS DB インスタンス識別子は、Amazon RDS を操作するときに Amazon ML が使用するデータベースインスタンスを識別する固有の名前です。RDS DB インスタンス識別子は、Amazon RDS コンソールで確認できます。

MySQL データベース名

MySQL データベース名は、RDS DB インスタンス内の MySQL データベースの名前を指定します。

データベースユーザー認証情報

RDS DB インスタンスに接続するには、提供する SQL クエリを実行するのに十分な権限を持つデータベースユーザーのユーザー名とパスワードを指定する必要があります。

AWS Data Pipeline セキュリティ情報

AWS Data Pipeline の安全なアクセスを有効にするには、IAM リソースロールと IAM サービスロールの名前を指定する必要があります。

EC2 インスタンスは、Amazon RDS から Amazon S3 にデータをコピーするリソースロールを引き受けます。このリソースロールを作成する最も簡単な方法は、DataPipelineDefaultResourceRole テンプレートを使用して、**machinelearning.aws.com** を信頼できるサービスとして一覧表示することです。テンプレートの設定の詳細については、「AWS Data Pipeline 開発者ガイド」の「IAM ロールの設定」を参照してください。

独自のロールを作成する場合は、そのロールには次の内容が必要です。

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "Service": "machinelearning.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringEquals": { "aws:SourceAccount": "123456789012" },
            "ArnLike": { "aws:SourceArn": "arn:aws:machinelearning:us-
east-1:123456789012:datasource/*" }
        }
    }]
}
```

AWS Data Pipeline は、サービスロールを引き受けて、Amazon RDS から Amazon S3 へのデータのコピーの進行をモニタリングします。このリソースロールを作成する最も簡単な方法は、DataPipelineDefaultRole テンプレートを使用して、machinelearning.aws.com を信頼できるサービスとして一覧表示することです。テンプレートの設定の詳細については、「AWS Data Pipeline 開発者ガイド」の「IAM ロールの設定」を参照してください。

Amazon RDS セキュリティ情報

安全な Amazon RDS アクセスを有効にするには、`VPC Subnet ID` と `RDS Security Group IDs` を指定する必要があります。また、`Subnet ID` パラメータが指す VPC サブネットに適切な入力規則を設定し、このアクセス許可を持つセキュリティグループの ID を指定する必要があります。

MySQL SQL クエリ

`MySQL SQL Query` パラメータは、MySQL データベースで実行する SQL SELECT クエリを指定します。クエリの結果は、指定した S3 出力の場所 (バケット) にコピーされます。

> **Note**
>
> 機械学習テクノロジーは、入力レコードがランダムな順序 (シャッフル) で表示されるときに最も効果的です。`rand( )` 関数を使用すると、MySQL SQL クエリの結果を簡単にシャッフルできます。たとえば、これが元のクエリであるとします。
> "SELECT col1, col2, … FROM training_table"
> 次のようにクエリを更新することでランダムなシャッフルを追加できます。
> "SELECT col1, col2, … FROM training_table ORDER BY rand()"

S3 出力の場所

`S3 Output Location` パラメータは、MySQL SQL クエリの結果が出力されるステージングする Amazon S3 の場所の名前を指定します。

> **Note**
>
> Amazon RDS からデータをエクスポートした後、Amazon ML にこの場所のデータを読み取る権限があることを確認する必要があります。これらのアクセス権限の設定の詳細については、Amazon S3 からデータを読み込むための Amazon ML 権限の付与を参照してください。

ML モデルのトレーニング

ML モデルのトレーニングプロセスには、ML アルゴリズム (つまり、学習アルゴリズム) を学習のためのトレーニングデータと共に提供することが含まれます。ML モデル とは、モデルトレーニングプロセスで作成されたモデルアーティファクトを指します。

トレーニングデータには正しい答えが含まれている必要があります。これは、ターゲットまたはターゲット属性として知られています。学習アルゴリズムは入力データ属性をターゲット (予測したい答え) にマッピングするトレーニングデータのパターンを検出し、これらのパターンをキャプチャする ML モデルを出力します。

ML モデルを使用すると、ターゲットがわからない新しいデータでターゲットを予測できます。たとえば、E メールがスパムかどうかを予測する ML モデルをトレーニングすると仮定します。Amazon ML に、ターゲット (E メールがスパムであるかどうかを示すラベル) がわかっている E メールのトレーニングデータを提供します。Amazon ML はこのデータを使用して ML モデルをトレーニングし、新しい E メールがスパムかどうかを予測するモデルになります。

ML モデルと ML アルゴリズムに関する一般的な情報については、「機械学習の概念」を参照してください。

トピック

ML モデルのタイプ

Amazon ML は、バイナリ分類、複数クラス分類、回帰の 3 つのタイプの ML モデルに対応しています。選択するモデルのタイプは、予測したいターゲットのタイプによって異なります。

バイナリ分類のモデル

バイナリ分類問題の ML モデルは、バイナリ結果 (2 つの可能なクラスのうちの 1 つ) を予測します。バイナリ分類モデルをトレーニングするために、Amazon ML はロジスティック回帰として知られる業界標準の学習アルゴリズムを使用します。

バイナリ分類問題の例

- 「このメールはスパムでしょうか」。
- 「顧客はこの製品を購入するでしょうか、それともしないでしょうか」。
- 「この商品は本でしょうか、それとも家畜でしょうか」。
- 「このレビューは顧客によって書かれたものでしょうか、それともロボットによって書かれたものでしょうか」。

複数クラスの分類モデル

複数クラス分類問題の ML モデルを使用すると、複数のクラスの予測を生成できます (2 つ以上の結果の 1 つを予測します)。複数クラスモデルをトレーニングするために、Amazon ML は複数ロジスティック回帰として知られる業界標準の学習アルゴリズムを使用します。

複数クラス問題の例

- 「この製品は書籍、映画、衣類のいずれですか」。
- 「この映画はロマンチックコメディ、ドキュメンタリー、またはスリラーですか」。
- 「この顧客にとって最も関心のある商品のカテゴリはどれですか」。

回帰モデル

回帰問題の ML モデルは数値を予測します。回帰モデルをトレーニングするために、Amazon ML は線形回帰として知られる業界標準の学習アルゴリズムを使用します。

回帰問題の例

- 「明日のシアトルの温度はどうなりますか」。
- 「この製品の販売台数は何台ですか」。
- 「この家はどのような値段で売れるでしょうか」。

トレーニングプロセス

ML モデルをトレーニングするには、次のように指定する必要があります。

- 入力トレーニングデータソース
- 予測ターゲットを含むデータ属性の名前

- 必要なデータ変換手順
- 学習アルゴリズムを制御するトレーニングパラメータ

トレーニングプロセス中、Amazon ML はトレーニングデータソースで指定したターゲットのタイプに基づいて、正しい学習アルゴリズムを自動的に選択します。

トレーニングパラメータ

通常、機械学習アルゴリズムは、トレーニングプロセスおよび結果として生じる ML モデルのプロパティを制御するために使用できるパラメータを受け入れます。Amazon Machine Learning では、これらをトレーニングパラメータと呼びます。Amazon ML コンソール、API、またはコマンドラインインターフェイス (CLI) を使用してこれらのパラメータを設定できます。パラメータを設定しない場合、Amazon ML は、機械学習のさまざまなタスクに適していることが知られているデフォルト値を使用します。

以下のトレーニングパラメータに値を指定できます。

- 最大モデルサイズ
- トレーニングデータへのパスの最大数
- シャッフルタイプ
- 正則化タイプ
- 正則化の量

Amazon ML コンソールで、トレーニングパラメータはデフォルトで設定されています。デフォルトの設定は、ほとんどの ML の問題に適していますが、パフォーマンスを微調整するために別の値を選択できます。学習レートなどのその他のトレーニングパラメータには、データに基づいて設定されているものがあります。

以下のセクションでは、トレーニングパラメータについて詳しく説明します。

最大モデルサイズ

最大モデルサイズは、Amazon ML が ML モデルのトレーニング中に作成するパターンの合計サイズ (バイト単位) です。

デフォルトでは、Amazon ML は 100 MB のモデルを作成します。異なるサイズを指定することで、より小さな、またはより大きなモデルを作成するよう Amazon ML に指示できます。使用可能なサイズの範囲については、「ML モデルのタイプ」を参照してください。

モデルサイズを満たすだけのパターンを Amazon ML が見つけられない場合は、より小さなモデルが作成されます。たとえば、最大モデルサイズとして 100 MB を指定しても、Amazon ML が合計 50 MB のパターンしか見つけられない場合は、結果のモデルは 50 MB となります。Amazon ML が指定したサイズより多くのパターンを見つけた場合は、学習したモデルの品質に最も影響の少ないパターンをトリミングして最大値でカットオフします。

モデルサイズを選択すると、モデルの予測品質と使用コストの間でのトレードオフを制御できます。より小さなモデルでは、最大サイズに収まるように Amazon ML により多くのパターンが削除され、予測の品質に影響を与えます。一方、より大きなモデルでは、リアルタイム予測のクエリのためにコストがより大きくなります。

> **Note**
>
> リアルタイム予測を生成するために ML モデルを使用する場合、モデルサイズにより決まる小さなキャパシティーの予約料金が発生します。詳細については、「Amazon ML の料金」を参照してください。

モデルは入力データではなくパターンを保存するため、入力データセットが大きくなるほどモデルが大きくなるとは限りません。パターンがシンプルであれば、結果として生じるモデルは小さくなります。多くの raw 属性 (入力列) や派生した機能 (Amazon ML データ変換の出力) を持つ入力データは、トレーニングプロセス中により多くのパターンを見つけて保存する可能性が高くなります。データと問題に対する正しいモデルサイズの選択は、いくつかの実験をすることで最も効果的に行えます。Amazon ML モデルのトレーニングログ (コンソールから、または API 経由でダウンロードできる) には、トレーニングプロセス中に発生したトリミングの量 (存在する場合) に関するメッセージが含まれていて、潜在的な予想の的中品質を見積もることができます。

データに対するパスの最大数

最良の結果を得るには、パターンを発見するために Amazon ML がデータを複数回パスする必要があるかもしれません。デフォルトでは、Amazon ML は 10 回のパスを行いますが、100 までの数値を設定することでデフォルトを変更できます。Amazon ML はパターンの品質 (モデルの収束) を追跡し、データポイントやパターンが発見できなくなったら自動的にトレーニングを終了します。例えば、パスの数を 20 に設定していても、15 のパスを終えた時点で新しいパターンはもう見つけられないと Amazon ML が判断すると、15 のパスでトレーニングを終了します。

通常、わずかな観測値しか持たないデータセットでは、より高いモデル品質を得るために、データに対してより多くのパスが必要となります。大きなデータセットには多くの同様のデータポイントが含

まれているため、多数のパスを必要としません。データにより多くのデータパスを選択することは二重の影響があります。モデルトレーニングにはより時間がかかり、コストも高くなります。

トレーニングデータのシャッフルタイプ

Amazon ML では、トレーニングデータをシャッフルする必要があります。シャッフルは、データの順序をミックスして、SGD アルゴリズムがあまりに多くの連続した観測で 1 つのタイプのデータに遭遇することがないようにします。たとえば、ML モデルをトレーニングして製品タイプを予測するとき、トレーニングデータに映画、玩具、ビデオゲームの製品タイプが含まれている場合、アップロードする前に製品タイプの列でデータを並べ替えた場合、アルゴリズムは製品タイプごとのアルファベット順にデータを見ていきます。アルゴリズムは、映画のすべてのデータを最初に見ていき、ML モデルは映画のパターンを学習し始めます。次に、モデルが玩具のデータに遭遇したとき、アルゴリズムが行うすべての更新は、その更新が映画に適したパターンを劣化させるとしても、モデルを玩具の製品タイプに適合させようとします。この映画から玩具のタイプへの突然の切り替えにより、製品タイプについての精度の高い予測を学習できないモデルが生成されます。

入力データソースをトレーニングと評価の部分に分割するときにランダム分割オプションを選択した場合でも、トレーニングデータをシャッフルする必要があります。ランダム分割の方法では、各データソースのデータのランダムなサブセットが選択されますが、データソース内の行の順序は変更されません。データ分割の詳細については、「データの分割」を参照してください。

コンソールを使用して ML モデルを作成すると、Amazon ML はデフォルトで、擬似乱数シャッフルの手法を使ってデータをシャッフルします。リクエストされたパスの数にかかわらず、Amazon ML は ML モデルをトレーニングする前にデータを 1 回だけシャッフルします。Amazon ML にデータを提供する前にシャッフルし、Amazon ML でデータを再度シャッフルしたくない場合は、[Shuffle type] (シャッフルタイプ) を none に設定できます。例えば、 Amazon S3 にアップロードする前に .csv ファイル内のレコードをランダムにシャッフルした場合、Amazon RDS からデータソースを作成する際に MySQL SQL クエリの `rand()` 関数を使用した場合、または、Amazon Redshift からデータソースを作成する際に Amazon Redshift SQL クエリの `random()` 関数を使用した場合は、[Shuffle type] (シャッフルタイプ) を none に設定しても、ML モデルの予測精度には影響しません。データを 1 回シャッフルするだけで、ML モデルを作成するための実行時間とコストが削減されます。

> **Important**
>
> Amazon ML API を使用して ML モデルを作成すると、Amazon ML はデフォルトではデータをシャッフルしません。コンソールではなく API を使用して ML モデルを作成する場合

は、sgd.shuffleType パラメータを auto に設定することで、データをシャッフルすることを強くお勧めします。

正則化のタイプと量

データに含まれるパターンが多すぎると、複雑な ML モデル (入力属性が多いモデル) の予測パフォーマンスが低下します。パターンの数が増えると、モデルが、実際のデータパターンではなく、意図しないデータアーティファクトを学習する可能性も高くなります。そのような場合、モデルはトレーニングデータではうまくいきますが、新しいデータではうまく一般化できません。この現象はトレーニングデータのオーバーフィットとして知られています。

正則化は、極端なウェイト値にペナルティを課すことによって、線形モデルがトレーニングデータの例にオーバーフィットするのを防ぎます。L1 正則化は、さもなければ非常に小さなウェイトを持つ機能のウェイトを 0 にすることによって、モデルで使用される機能の数を減らします。L1 正則化は、まばらなモデルを生成し、モデル内のノイズの量を低減します。L2 正則化により、全体のウェイトの値が小さくなり、機能間の相関性が高い場合にウェイトを安定させます。Regularization amount パラメータを使用して、L1 または L2 正則化の量を調整できます。非常に大きな Regularization amount 値を指定すると、すべての機能のウェイトがゼロになる可能性があります。

最適な正則化の値を選択して調整することは、機械学習の分野で活発に研究されている課題です。Amazon ML コンソールのデフォルトである適度な量の L2 正則化を選択することにはメリットがあるでしょう。上級ユーザーは、正則化の 3 つのタイプ (none、L1、または L2) と量を選択できます。正則化の詳細については、「正則化 (数学)」を参照してください。

トレーニングパラメータ: タイプとデフォルト値

次の表に、Amazon ML トレーニングパラメータと、それぞれのデフォルト値と許容範囲を示します。

トレーニングパラメータ	タイプ	[Default Value] (デフォルト値)	説明
maxMLModelSizeInBytes	整数	100,000,000 バイト (100 MiB)	許容範囲:100,000 (100 KiB) ~ 2,147,483,648 (2 GiB)

トレーニングパラメータ	タイプ	[Default Value] (デフォルト値)	説明
			入力データによっては、モデルのサイズがパフォーマンスに影響する可能性があります。
sgd.maxPasses	整数	10	許容範囲: 1 ~ 100
sgd.shuffleType	文字列	auto	許容範囲: auto または none
sgd.l1RegularizationAmount	ダブル	0 (デフォルト、L1 は使用されません)	許容範囲: 0 ~ MAX_DOUBLE L1 の値を 1E-4 と 1E-8 の間にすると良好な結果が得られることが分かっています。それより大きな値では、役立つモデルが生成される可能性はあまりありません。 L1 と L2 の両方を設定することはできません。どちらかを選択する必要があります。
sgd.l2RegularizationAmount	ダブル	1E-6 (デフォルト、L2 はこの量の正則化を使用します)	許容範囲: 0 ~ MAX_DOUBLE L2 の値を 1E-2 と 1E-6 の間にすると良好な結果が得られることが分かっています。それより大きな値では、役立つモデルが生成される可能性はあまりありません。 L1 と L2 の両方を設定することはできません。どちらかを選択する必要があります。

ML モデルの作成

データソースを作成したら、ML モデルを作成できます。Amazon Machine Learning コンソールを使用してモデルを作成する場合は、デフォルト設定を使用するか、カスタムオプションを適用してモデルをカスタマイズするかを選択できます。

カスタムオプションは次のとおりです。

- 評価設定: Amazon ML に入力データの一部を保持させ、ML モデルの予測品質を評価させることができます。評価の詳細については、「ML モデルの評価」を参照してください。
- レシピ: レシピは Amazon ML に、モデルトレーニングに使用できる属性と属性変換を知らせます。Amazon ML レシピの詳細については、「データレシピを使用した機能変換」を参照してください。
- トレーニングパラメータ: パラメータは、トレーニングプロセスおよび結果として生じる ML モデルの特定のプロパティを制御します。トレーニングパラメータの詳細については、「トレーニングパラメータ」を参照してください。

これらの設定の値を選択または指定するには、ML モデル作成ウィザードを使用するときに [カスタム] オプションを選択します。Amazon ML にデフォルト設定を適用する場合は、[Default] (デフォルト) を選択します。

ML モデルを作成すると、Amazon ML はターゲット属性の属性タイプに基づいて、使用する学習アルゴリズムのタイプを選択します。(ターゲット属性とは「正しい」回答を含む属性のことです。) ターゲット属性がバイナリの場合、Amazon ML はロジスティック回帰アルゴリズムを使用するバイナリ分類モデルを作成します。ターゲット属性がカテゴリの場合、Amazon ML は多項ロジスティック回帰アルゴリズムを使用する複数クラスモデルを作成します。ターゲット属性が数値の場合、Amazon ML は直線回帰アルゴリズムを使用する回帰モデルを作成します。

トピック

前提条件

Amazon ML コンソールを使用して ML モデルを作成する前に、モデルのトレーニング用とモデル評価用の 2 つのデータソースを作成する必要があります。2 つのデータソースをまだ作成していない

場合は、「ステップ 2: トレーニングデータソースを作成する」のチュートリアルを参照してください。

デフォルトオプションで ML モデルを作成する

Amazon ML で以下のことを行う場合は、[Default] (デフォルト) オプションを選択します。

- 入力データを分割して最初の 70% をトレーニングに使用し、残りの 30% を評価に使用します
- トレーニングデータソースで収集された統計 (入力データソースの 70%) に基づいてレシピを提案します
- デフォルトのトレーニングパラメータを選択します

デフォルトのオプションを選択するには

1. Amazon ML コンソールで [Amazon Machine Learning] を選択してから、[ML models] (ML モデル) を選択します。
2. [ML モデル] の概要ページで、[新しい ML モデルを作成] を選択します。
3. [入力データ] ページで、[S3 データを指すデータソースを既に作成しました] が選択されていることを確認します。
4. 表からデータソースを選択し、[続行] を選択します。
5. [ML モデル設定] ページの [ML モデル名] に ML モデルの名前を入力します。
6. [トレーニングおよび評価設定] で、[デフォルト] が選択されていることを確認します。
7. [Name this evaluation] (この評価に名前を設定) で、評価の名前を入力して [Review] (確認) を選択します。Amazon ML により残りのウィザードはスキップされ、[Review] (確認) ページに移動します。
8. データを確認し、モデルと評価に適用しないデータソースからコピーしたタグをすべて削除し、[完了] を選択します。

カスタムオプションで ML モデルを作成する

ML モデルのカスタマイズを行うと以下のことができます。

- 独自のレシピを提供します。独自のレシピを提供する方法の詳細については、「レシピ形式のリファレンス」を参照してください。
- トレーニングパラメータを選択します。トレーニングパラメータの詳細については、「トレーニングパラメータ」を参照してください。

- デフォルトの 70/30 以外のトレーニング/評価分割比を選択するか、または、評価のために準備した別のデータソースを提供してください。分割方法の詳細については、「データの分割」を参照してください。

これらの設定のデフォルト値を選択することもできます。

デフォルトのオプションを使用してすでにモデルを作成していて、モデルの予測パフォーマンスを向上させたい場合は、[カスタム] オプションを使用してカスタマイズされた設定で新しいモデルを作成します。たとえば、機能変換をレシピに追加する、または、トレーニングパラメータのパスの数を増やすことができます。

カスタムオプションでモデルを作成するには

1. Amazon ML コンソールで [Amazon Machine Learning] を選択してから、[ML models] (ML モデル) を選択します。
2. [ML モデル] の概要ページで、[新しい ML モデルを作成] を選択します。
3. データソースをすでに作成している場合は、[入力データ] ページで、[S3 データを指すデータソースを既に作成しました] を選択します。表からデータソースを選択し、[続行] を選択します。

   データソースを作成する必要がある場合は、[データは S3 にあり、データソースを作成する必要があります] を選択した後、[続行] を選択します。[Create a Datasource (データソースの作成)] ウィザードにリダイレクトされます。データが [S3] または [Redshift] にあるかを指定し、[検証] を選択します。データソースを作成する手順を完了します。

   データソースを作成したら、[Create ML Model (ML モデルの作成)] ウィザードの次のステップにリダイレクトされます。
4. [ML モデル設定] ページの [ML モデル名] に ML モデルの名前を入力します。
5. [Select training and evaluation settings (トレーニングおよび評価設定の選択)] で、[カスタム] を選択した後、[続行] を選択します。
6. [レシピ] ページで、customize a recipe を行えます。レシピをカスタマイズしない場合は、Amazon ML がレシピを提案します。[Continue] (続行) をクリックします。
7. [詳細設定] ページで、[最大 ML モデルサイズ]、[データパスの最大数]、[トレーニングデータのシャッフルタイプ]、[正則化タイプ]、および [正則化の量] を指定します。これらを指定しない場合、Amazon ML はデフォルトのトレーニングパラメータを使用します。

これらのパラメータおよびデフォルトの詳細については、「トレーニングパラメータ」を参照してください。

[Continue] (続行) をクリックします。

8. [評価] ページで、すぐに ML モデルを評価するかどうかを指定します。ML モデルをすぐに評価しない場合は、[Review (レビュー)] を選択します。

   ML モデルを今すぐ評価する場合

   a. [この評価に名前を設定] に、評価の名前を入力します。
   b. [Select evaluation data] (評価データを選択) で、Amazon ML が評価のために入力データの一部を保持するかどうかを選択して、そうするのであれば、データソースの分割方法、または評価のために異なるデータソースを提供することを選択します。
   c. [Review] (レビュー) を選択します。

9. [Review (レビュー)] ページで、選択を編集し、モデルと評価に適用しないデータソースからコピーしたタグをすべて削除して、[終了] を選択します。

モデルを作成した後は、「ステップ 4: ML モデルの予測パフォーマンスを確認し、スコアのしきい値を設定する」を参照してください。

機械学習のデータ変換

データは、機械学習モデルのトレーニングに使用されます。良いトレーニングデータの重要な特徴は、学習と一般化のために最適化された方法で提供されることです。この最適なフォーマットでデータをまとめるプロセスは、業界では機能変換として知られています。

トピック

機能変換の重要性

クレジットカードの取引が不正であるかどうかを判断することを目的とした機械学習モデルを考えてみましょう。アプリケーションの背景知識とデータ分析に基づいて、入力データに含めることが重要なデータフィールド (または機能) を決定できます。たとえば、取引金額、販売者名、住所、クレジットカード所有者の住所を学習プロセスに提供するのは重要です。一方、ランダムに生成されたトランザクション ID には情報が含まれておらず (実際にランダムであることが分かっている場合)、有用ではありません。

どのフィールドを含めるかを決めたら、これらの機能を変換して学習プロセスに役立てます。変換により、入力データにバックグラウンド経験が追加され、機械学習モデルはこの経験からのメリットを得られます。たとえば、次の販売者住所は文字列として表されます。

「123 Main Street, Seattle, WA 98101」

これ自体では、これとまったく同じ住所に関連したパターンの学習にしか役立たないので、表現力が限られています。しかし、それを構成部分に分割すると、「住所」 (123 Main Street)、「市」 (Seattle)、「州」 (WA)、および、「郵便番号」 (98101) などの追加機能を作成できます。これで、学習アルゴリズムは、より多くの異なるトランザクションをグループ化し、より広範なパターンを発見することができます。たとえば、一部の販売者の郵便番号では、他よりも多くの不正行為を経験しているかもしれません。

機能変換のアプローチと処理の詳細については、「機械学習の概念」を参照してください。

データレシピを使用した機能変換

Amazon ML で ML モデルを作成する前に機能を変換する方法は 2 つあります。入力データを Amazon ML に表示する前に直接変換するか、Amazon ML の組み込みデータ変換を使用します。Amazon ML のレシピを使用することができます。これは、事前にフォーマットされた一般的な変換のための手順です。レシピでは、以下を実行できます。

- 組込みの共通マシン学習変換のリストから選択し、個々の変数または変数のグループに適用します
- 機械学習プロセスで使用できる入力変数と変換の選択を選択します

Amazon ML のレシピを使用すると、いくつかの利点があります。Amazon ML はデータ変換を実行するので、自分で実装する必要はありません。さらに、Amazon ML は入力データを読み取っている間に変換を適用し、結果をディスクに保存する途中で学習プロセスに結果を提供するため、高速です。

レシピ形式リファレンス

Amazon ML レシピには、機械学習プロセスの一部としてデータを変換する手順が含まれています。レシピは JSON に似た構文を使用して定義されていますが、通常の JSON の制限の上に追加の制限があります。レシピには以下のセクションがあり、ここに示す順序で表示される必要があります。

- グループ は、複数の変数をグループ化でき、変換の適用がしやすくなります。たとえば、ウェブページのフリーテキスト部分 (タイトル、本文) と関係があるすべての変数のグループを作成し、これらのすべての部分を一度に変換することができます。
- 割り当ては、処理中に再利用できる中間の名前付き変数の作成ができます。
- 出力は、学習プロセスで使用される変数と、これらの変数に適用される変換 (存在する場合) を定義します。

グループ

グループ内のすべての変数を一括して変換したり、これらの変数を変換せずに機械学習に使用するために、変数のグループを定義できます。デフォルトでは、Amazon ML は以下のグループを作成します。

ALL_TEXT、ALL_NUMERIC、ALL_CATEGORICAL、ALL_BINARY –データソーススキーマで定義された変数に基づくタイプ固有のグループ。

> **Note**
>
> ALL_INPUTS でグループを作成することはできません。

これらの変数は、定義されていないレシピの出力セクションで使用できます。また、既存のグループに変数を追加または削除することで、または、変数のコレクションから直接、カスタムグループを作成することもできます。次の例では、3 つすべてのアプローチとグループ化の割り当ての構文を示します。

```
"groups": {

"Custom_Group": "group(var1, var2)",
"All_Categorical_plus_one_other": "group(ALL_CATEGORICAL, var2)"

}
```

グループ名はアルファベット文字で始まる必要があり、長さは 1 ～ 64 文字です。グループ名がアルファベット文字で始まらない場合、または特殊文字 (, ' " \t \r \n ( ) \) が含まれている場合は、その名前を引用符で囲んでレシピに含める必要があります。

割り当て

利便性と可読性のために、1 つまたは複数の変換を中間変数に割り当てることができます。たとえば、email_subject という名前のテキスト変数があり、それに小文字の変換を適用すると、結果の変数に email_subject_lowercase という名前を付けることができ、レシピ内のどこでもその変数を簡単に追跡することができます。また、割り当てを連鎖させて、指定した順序で複数の変換を適用することもできます。次の例は、レシピ構文の単一および連鎖の割り当てを示しています。

```
"assignments": {

"email_subject_lowercase": "lowercase(email_subject)",

"email_subject_lowercase_ngram":"ngram(lowercase(email_subject), 2)"
```

```
}
```

中間変数はアルファベット文字で始まる必要があり、長さは 1 ～ 64 文字です。名前がアルファベットで始まらない場合、または特殊文字 (, ' " \t \r \n ( ) \) が含まれている場合は、その名前を引用符で囲んでレシピに含める必要があります。

[Outputs] (出力)

出力セクションは、どの入力変数が学習プロセスに使用されるか、どの変換が適用されるかを制御します。空の、または、存在しない出力セクションは、学習プロセスにデータが渡されないため、エラーとなります。

最も単純な出力セクションには、定義済みの ALL_INPUTS グループが含まれていて、Amazon ML が学習用にデータソースで定義されているすべての変数を使用するように指示します。

```
"outputs": [

"ALL_INPUTS"

]
```

出力セクションは、Amazon ML にこれらのグループのすべての変数を使用するように指示することで、他の定義済みグループを参照することもできます。

```
"outputs": [

"ALL_NUMERIC",

"ALL_CATEGORICAL"

]
```

出力セクションはカスタムグループを参照することもできます。次の例では、前の例のグループ割り当てセクションで定義されたカスタムグループの 1 つのみが機械学習に使用されます。他のすべての変数は削除されます。

```
"outputs": [

"All_Categorical_plus_one_other"
```

```
]
```

出力セクションは、割り当てセクションで定義された変数割り当てを参照することもできます。

```
"outputs": [

"email_subject_lowercase"

]
```

また、入力変数または変換は、出力セクションで直接定義できます。

```

"outputs": [

"var1",

"lowercase(var2)"

]
```

出力は、学習プロセスで使用できると予想されるすべての変数と変換された変数を明示的に指定する必要があります。たとえば、var1 と var2 のデカルト積を出力に含めるとします。raw 変数 var1 と var2 の両方も含める場合は、出力セクションに raw 変数を追加する必要があります。

```

"outputs": [

"cartesian(var1,var2)",

"var1",

"var2"

]
```

出力には、変数とともにコメントテキストを追加して、読みやすくするためのコメントを含めることができます。

```
"outputs": [

"quantile_bin(age, 10) //quantile bin age",

"age // explicitly include the original numeric variable along with the
binned version"

]
```

出力セクション内でこれらのアプローチをすべて組み合わせて利用することができます。

> **Note**
>
> レシピを追加するときに Amazon ML コンソールでコメントを入力することはできません。

完全なレシピの例

以下の例は、前述の例で紹介されたいくつかのビルトインデータプロセッサを示しています。

```
{

"groups": {

"LONGTEXT": "group_remove(ALL_TEXT, title, subject)",

"SPECIALTEXT": "group(title, subject)",

"BINCAT": "group(ALL_CATEGORICAL, ALL_BINARY)"

},

"assignments": {

"binned_age" : "quantile_bin(age,30)",

"country_gender_interaction" : "cartesian(country, gender)"

},

"outputs": [
```

```
"lowercase(no_punct(LONGTEXT))",

"ngram(lowercase(no_punct(SPECIALTEXT)),3)",

"quantile_bin(hours-per-week, 10)",

"hours-per-week // explicitly include the original numeric variable
along with the binned version",

"cartesian(binned_age, quantile_bin(hours-per-week,10)) // this one is
critical",

"country_gender_interaction",

"BINCAT"

]

}
```

推奨レシピ

Amazon ML で新しいデータソースを作成し、そのデータソースの統計が計算されると、Amazon ML はデータソースから新しい ML モデルを作成するために使用できる推奨レシピを作成します。提案されたデータソースは、データに存在するデータ属性とターゲット属性に基づいており、ML モデルの作成とチューニングの出発点となります。

Amazon ML コンソールで提案されたレシピを使用するには、[Create new] (新規作成) ドロップダウンリストから [Datasource] (データソース) または [Datasource and ML model] (データソースおよび ML モデル) を選択します。ML モデルの設定では、[Create ML Model] (ML モデルの作成) ウィザードの [ML Model Setting] (ML モデルの設定) のステップで、デフォルトまたはカスタムのトレーニングと評価の設定を選択できます。[Default] デフォルトオプションを選択すると、Amazon ML は推奨されたレシピを自動的に使用します。[Custom] カスタムオプションを選択すると、次のステップのレシピエディタに推奨レシピが表示され、必要に応じて確認または変更できます。

> **Note**
>
> Amazon ML では、統計情報の計算が完了する前に、データソースを作成してすぐに ML モデルを作成することができます。この場合、カスタムオプションで提案されたレシピを表示

することはできませんが、そのステップを進んで、Amazon ML にモデルトレーニングのデフォルトレシピを使用させることができます。

Amazon ML API で提案されたレシピを使用するには、Recipe API と RecipeUri API の両方のパラメータで空の文字列を渡すことができます。Amazon ML API を使用して推奨されたレシピを取得することはできません。

データ変換リファレンス

トピック

nグラム変換

nグラム変換は、テキスト変数を入力として受け取り、(ユーザが設定可能な) n ワードのウィンドウをスライドさせることに対応する文字列を生成し、プロセス内の出力を生成します。たとえば、「この本を本当に楽しんで読みました」という文字列を考えてみましょう。

ウィンドウサイズ = 1 で nグラム変換を指定すると、その文字列内の個々の単語がすべて得られます。

```
{"I", "really", "enjoyed", "reading", "this", "book"}
```

ウィンドウサイズ = 2 でnグラム変換を指定すると、すべての 2 語の組み合わせと 1 語の組み合わせが得られます。

```
{"I really", "really enjoyed", "enjoyed reading", "reading this", "this
```

```
book", "I", "really", "enjoyed", "reading", "this", "book"}
```

ウィンドウサイズ = 3 で nグラム変換を指定すると、このリストに 3 語の組み合わせが追加され、次の結果が得られます。

```
{"I really enjoyed", "really enjoyed reading", "enjoyed reading this",
"reading this book", "I really", "really enjoyed", "enjoyed reading",
"reading this", "this book", "I", "really", "enjoyed", "reading",
"this", "book"}
```

n-grams は、2〜10 語の範囲のサイズでリクエストできます。サイズ 1 の n-grams は、データスキーマでタイプがテキストとしてマークされているすべての入力に対して暗黙的に生成されるため、入力する必要はありません。最後に、n-grams は空白文字の入力データを分割することによって生成されることに注意してください。これは、たとえば、句読点文字が単語トークンの一部とみなされることを意味します。文字列「red, green, blue」をウィンドウ 2 で n-grams を生成すると、{"red,", "green,", "blue,", "red, green", "green, blue"} となります。これを希望しない場合は、句読記号を削除するために、句読点除去プロセッサ (この資料で後述) を使用して句読記号を削除することができます。

変数 var1 のウィンドウサイズ 3 の n-grams を計算するには。

```
"ngram(var1, 3)"
```

直角のスパースなバイグラム (OSB) 変換

OSB 変換はテキスト文字列解析を支援することを目的としており、バイグラム変換 (ウィンドウサイズ 2 の nグラム) に代わるものです。OSB は、テキスト上で n サイズのウィンドウをずらし、ウィンドウの最初の単語を含む、各単語ペアを出力することにより生成されます。

各 OSB を構築するために、その構成要素の単語は「_」(アンダースコア) 文字で結合され、スキップされたすべてのトークンは OSB にアンダースコアを追加することによって示されます。したがって、OSB は、ウィンドウ内に見られるトークンだけでなく、同じウィンドウ内でスキップされたトークンの数もエンコードします。

例として、文字列「quick brown fox dog」、OSB サイズ 4 を考えます。文字列の最後から 6 つの 4 単語のウィンドウと最後の 2 つの短いウィンドウと、同様にそれぞれから生成された OSB が次の例に示されています。

ウィンドウ {OSB 生成}

```
"The quick brown fox", {The_quick, The__brown, The___fox}

"quick brown fox jumps", {quick_brown, quick__fox, quick___jumps}

"brown fox jumps over", {brown_fox, brown__jumps, brown___over}

"fox jumps over the", {fox_jumps, fox__over, fox___the}

"jumps over the lazy", {jumps_over, jumps__the, jumps___lazy}

"over the lazy dog", {over_the, over__lazy, over___dog}

"the lazy dog", {the_lazy, the__dog}

"lazy dog", {lazy_dog}
```

直交スパースバイグラムは、nグラムの代替品で状況によっては快適に動作する可能性があります。データに大きなテキストフィールド (10 単語以上) がある場合は、どちらがよく動作するか実験してみてください。何が大きなテキストフィールドとなるかは、状況によって異なる可能性があることに注意してください。ただし、大きなテキストフィールドの場合、経験的には OSB が、特別な省略記号 (下線) により、一意にテキストを表すことが示されています。

入力テキスト変数の OSB 変換では、ウィンドウサイズを 2〜10 にリクエストできます。

変数 var1 のウィンドウサイズ 5 の OSB を計算するには。

「osb(var1, 5)」

小文字変換

小文字変換プロセッサは、テキスト入力を小文字に変換します。たとえば、入力が「The Quick Brown Fox Jumps Over the Lazy Dog」の場合、プロセッサは「the quick brown fox jumps over the lazy dog」を出力します。

変数 var1 に小文字変換を適用するには。

「lowercase(var1)」

句読点除去変換

Amazon ML は、空白のデータスキーマのテキストとしてマークされた入力を暗黙的に分割します。文字列の句読点は、隣接する単語トークンで終わるか、それを囲む空白に応じて別のトークンとして完全に終わります。これが望ましくない場合、句読点除去変換を使用して、生成された機能から句読記号を除去することができます。「Welcome to AML - please fasten your seat-belts!」という文字列を指定すると、次のトークンセットが暗黙的に生成されます。

```
{"Welcome", "to", "Amazon", "ML", "-", "please", "fasten", "your", "seat-belts!"}
```

句読点除去プロセッサをこの文字列に適用すると、次のようになります。

```
{"Welcome", "to", "Amazon", "ML", "please", "fasten", "your", "seat-belts"}
```

プレフィックスとサフィックスの句読記号だけが削除されることに注意してください。トークンの途中に現れる句読点、たとえば「seat-belts」のハイフンは削除されません。

変数 var1 に句読点除去を適用するには。

「no_punct(var1)」

四分位ビニング変換

四分位ビニングプロセッサは、数値変数と bin 番号と呼ばれるパラメータの 2 つの入力を受け取り、カテゴリ変数を出力します。目的は、観測値をグループ化して変数の分布における非直線性を発見することです。

多くの場合、数値変数とターゲットの関係は線形ではありません (数値変数の値は単調に増減しません)。そのような場合、数値機能のさまざまな範囲を表すカテゴリ機能に数値機能を格納すると便利です。各カテゴリ機能値 (bin) は、ターゲットとのそれ自身の線形関係を持つものとしてモデル化することができます。たとえば、継続的な数値機能 account_age が書籍を購入する可能性と直線的に相関していないことがわかったとします。ターゲットとの関係をより正確に把握できるようなカテゴリ機能に分類することができます。

四分位ビニングプロセッサを使用して、Amazon ML に、age 変数のすべての入力値の分布に基づいて等しいサイズの n ビンを確立し、各ビンを含むテキストトークンで置き換えるよう指示できます。数値変数の最適なビン数は、変数の特性とターゲットとの関係に依存します。これは、実験を通じて最もよく決定されます。Amazon ML では、推奨レシピのデータ統計に基づいて数値機能の最適なビン数を示唆しています。

任意の数値入力変数に対して 5 から 1000 個の分位ビンを計算して要求することができます。

次の例は、数値変数 var1 の代わりに 50 ビンを計算して使用する方法を示しています。

「quantile_bin(var1, 50)」

正規化変換

正規化変換は、平均値がゼロで分散が 1 になるように数値変数を正規化します。数値変数の正規化は、数値的変数間の距離の差が非常に大きい場合に学習プロセスを助けることができます。なぜなら、その特徴がターゲットに対して有益であるかどうかにかかわらず、最大の変数が ML モデルを支配するからです。

この変換を数値変数 var1 に適用するには、これをレシピに追加します。

normalize(var1)

この変換は、ユーザー定義の数値変数グループまたはすべての数値変数 (ALL_NUMERIC) の事前定義グループを入力として受け取ることもできます。

normalize(ALL_NUMERIC)

[Note] (メモ)

正規化プロセッサを数値変数に使用することは必須ではありません。

デカルト積変換

デカルト変換は、2 つ以上のテキストまたはカテゴリ入力変数の順列を生成します。この変換は、変数間の相互作用が疑われる場合に使用されます。たとえば、チュートリアル :Amazon ML を使用してマーケティングオファーへの応答を予測するのに使用される銀行マーケティングデータセットを考えてみましょう。このデータセットを使用して、経済的および人口統計的情報に基づいて、人が銀行のプロモーションに積極的に反応するかどうかを予測します。当社では、個人の仕事の種類が重要であると考えるかもしれません (特定の分野で雇用されて利用可能なお金を手に入れること)、最高水準の教育の獲得も重要です。当社はまた、これらの 2 つの変数の相互作用に強いシグナルがあるという、より深い直感を持っているかもしれません - たとえば、昇進は大学の学位を取得した起業家である顧客に特に適しています。

デカルト積変換は、カテゴリ変数またはテキストを入力として受け取り、これらの入力変数間の相互作用を取得する新しい機能を生成します。具体的には、トレーニング例ごとに機能の組み合わせを作成し、スタンドアロン機能として追加します。たとえば、簡略化した入力行が次のようになっているとします。

ターゲット、教育、仕事

0、university.degree、技術者

0、high.school、サービス

1、university.degree、管理者

デカルト変換をカテゴリ変数の教育分野と職種分野に適用するように指定すると、結果の機能 education_job_interaction は次のようになります。

ターゲット、education_job_interaction

0、university.degree_technician

0、high.school_services

1、university.degree_admin

デカルト変換は、引数の 1 つが暗黙的である場合、または明示的にトークンに分割されるテキスト変数である場合、一連のトークンを処理するときにさらに強力になります。たとえば、書籍が教科書かどうかを分類するタスクについて検討します。直感的には、教科書であることを示す書籍のタイトル (教科書のタイトルでは特定の単語が頻繁に出現する可能性がある) があると思われるかもしれません。また、本の装丁で予測できると考えるかもしれませんが (教科書はハードカバーになる可能性が高いため)、実際にはタイトルと装丁についてのいくつかの単語の組み合わせで予測できます。実際の例で、次の表は、入力変数 binding および title にデカルトプロセッサを適用した結果を示しています。

教科書	タイトル	装丁	no_punct (タイトル) と装丁のデカルト積
1	経済学: 原則、問題、ポリシー	ハードカバー	{"Economics_Hardcover", "Principles_Hardcover", "Problems_Hardcover", "Policies_Hardcover"}
0	The Invisible Heart: An Economics Romance	ソフトカバー	{"The_Softcover", "Invisible_Softcover", "Heart_Softcover", "An_Softcover", "Economics_Softcover", "Romance_Softcover"}
0	Fun With Problems	ソフトカバー	{"Fun_Softcover", "With_Softcover", "Problems_Softcover"}

次の例は、デカルト変換を var1 と var2 に適用する方法を示しています。

cartesian(var1, var2)

データ再配置

データ再配置機能を使用すると、入力データの一部にのみ基づいてデータソースを作成できます。例えば、Amazon ML コンソールの [Create ML Model] (ML モデルの作成) ウィザードを使用して ML モデルを作成し、デフォルトの評価オプションを選択すると、Amazon ML は自動的に ML モデル評価のためにデータの 30% を予約して、残りの 70% をトレーニングに使用します。この機能は、Amazon ML のデータ再編成機能によって有効になります。

Amazon ML API を使用してデータソースを作成する場合は、入力データのどの部分に新しいデータソースが基づいているかを指定できます。これを行うには、`DataRearrangement` パラメータの指示を `CreateDataSourceFromS3`、`CreateDataSourceFromRedshift` API、または `CreateDataSourceFromRDS` API に渡します。DataRearrangement 文字列の内容は、データの開始位置と終了位置を含む JSON 文字列で、パーセンテージ、補完フラグ、および分割戦略で表されます。たとえば、次の DataRearrangement 文字列は、最初の 70% のデータがデータソースの作成に使用されることを指定します。

```
{
    "splitting": {
        "percentBegin": 0,
        "percentEnd": 70,
        "complement": false,
        "strategy": "sequential"
    }
}
```

DataRearrangement パラメータ

Amazon ML がデータソースを作成する方法を変更するには、以下のパラメータを使用します。

PercentBegin (オプション)

`percentBegin` を使用して、データソースのデータの開始位置を指定します。`percentBegin` と `percentEnd` を指定しなければ、データソースの作成時に Amazon ML にすべてのデータが含まれます。

有効な値は `0` ~ `100` です。

PercentEnd (オプション)

percentEnd を使用して、データソースのデータの終了位置を指定します。percentBegin と percentEnd を指定しなければ、データソースの作成時に Amazon ML にすべてのデータが含まれます。

有効な値は 0 ~ 100 です。

補完 (オプション)

complement パラメータは、Amazon ML が percentBegin から percentEnd の範囲に含まれていないデータを使用してデータソースを作成するようにします。complement パラメータは、トレーニングと評価のための補完的なデータソースを作成する必要がある場合に便利です。補完的なデータソースを作成するには、percentBegin パラメータで percentEnd および complement と同じ値を使用します。

たとえば、次の 2 つのデータソースはデータを共有せず、モデルをトレーニングおよび評価するために使用できます。最初のデータソースには 25% のデータがあり、2 番目のデータソースは 75% のデータがあります。

評価のためのデータソース

```
{
    "splitting":{
        "percentBegin":0,
        "percentEnd":25
    }
}
```

トレーニングのためのデータソース

```
{
    "splitting":{
        "percentBegin":0,
        "percentEnd":25,
        "complement":"true"
    }
}
```

有効な値は、true および false です。

戦略 (オプション)

Amazon ML がデータソースのデータをどのように分割するかを変更するには、strategy パラメータを使用します。

strategy パラメータのデフォルト値は sequential です。つまり、Amazon ML は、レコードが入力データに表示される順序で、データソースの percentBegin と percentEnd 間のすべてのデータレコードを取得します。

次の 2 つの DataRearrangement 行は、順番に順序付けられたトレーニングと評価のデータソースの例です。

評価のためのデータソース。{"splitting":{"percentBegin":70, "percentEnd":100, "strategy":"sequential"}}

トレーニングのためのデータソース。{"splitting":{"percentBegin":70, "percentEnd":100, "strategy":"sequential", "complement":"true"}}

データをランダムに選択してデータソースを作成するには、strategy パラメータを random に設定し、ランダムデータ分割のシード値として使用する文字列を指定します (たとえば、データへの S3 パスをランダムなシード文字列として使用できます)。ランダムな分割戦略を選択した場合、Amazon ML は各データ行に擬似乱数を割り当て、percentBegin と percentEnd の間に割り当てられた数を持つ行を選択します。バイトオフセットをシードとして使用して疑似乱数が割り当てられるため、データを変更すると異なる分割が発生します。既存の順序はすべて保存されます。ランダムな分割戦略により、トレーニングデータと評価データの変数が同様に分散されます。入力データに暗黙的な並べ替え順序が含まれている場合に役立ちます。そうでない場合は、類似しないデータレコードを含むトレーニングおよび評価データソースが作成されます。

次の 2 つの DataRearrangement 行は、非連続的なトレーニングと評価のデータソースの例です。

評価のためのデータソース

```
{
    "splitting":{
        "percentBegin":70,
        "percentEnd":100,
        "strategy":"random",
        "strategyParams": {
            "randomSeed":"RANDOMSEED"
        }
```

```
    }
}
```

トレーニングのためのデータソース

```
{
    "splitting":{
        "percentBegin":70,
        "percentEnd":100,
        "strategy":"random",
        "strategyParams": {
            "randomSeed":"RANDOMSEED"
        }
        "complement":"true"
    }
}
```

有効な値は、`sequential` および `random` です。

(オプション) 戦略。RandomSeed

Amazon ML は [randomSeed] を使用してデータを分割します。API のデフォルトのシードは空の文字列です。ランダムな分割戦略のシードを指定するには、文字列を渡します。random Seed について詳しくは、「Amazon Machine Learning デベロッパーガイド」の「データのランダムな分割」を参照してください。

Amazon ML でクロスバリデーションを使用する方法を示すサンプルコードについては、「Github Machine Learning Samples」を参照してください。

ML モデルの評価

新しいデータと将来のデータでターゲットを予測がうまくいくかどうかを判断するために、常にモデルを評価する必要があります。将来のインスタンスには不明なターゲット値があるため、ターゲットの回答をすでに知っているデータで ML モデルの精度メトリクスを確認し、この評価を将来のデータの予測精度のプロキシとして使用する必要があります。

モデルを適切に評価するには、トレーニングデータソースのターゲット (グランドトゥルース) でラベル付けされたデータのサンプルを提出します。トレーニングに使用されたのと同じデータを持つ ML モデルの予測精度を評価することは有用ではありません。なぜなら、トレーニングデータを一般化するのではなく、トレーニングデータを「覚える」モデルに報いるからです。ML モデルのトレーニングが終了したら、ターゲット値を知っている提出された観測値をモデルに送信します。次に、ML モデルによって返された予測と既知のターゲット値を比較します。最後に、予測された値と真の値がどれくらい一致しているかを示すサマリーメトリクスを計算します。

Amazon ML では、評価を作成して、ML モデルを評価します。ML モデルの評価を作成するには、評価する ML モデルが必要であり、トレーニングに使用されなかったラベル付きデータが必要です。まず、提出されたデータを持つ Amazon ML データソースを作成して、評価用のデータソースを作成します。評価に使用するデータは、トレーニングで使用されたデータと同じスキーマを持ち、ターゲット変数の実際の値を含んでいる必要があります。

すべてのデータが単一のファイルまたはディレクトリにある場合は、Amazon ML コンソールを使用してデータを分割できます。ML モデルの作成ウィザードのデフォルトパスは入力データソースを分割して、最初の 70% をトレーニングデータソースに使用し、残りの 30% を評価データソースに使用します。ML モデルの作成ウィザードのカスタムオプションを使用して、分割比率をカスタマイズすることもできます。ウィザードでは、トレーニング用にランダムな 70% のサンプルを選択し、残りの 30% を評価に使用できます。カスタム分割比率をさらに指定するには、データソースの作成 API でデータ再配置文字列を使用します。評価データソースと ML モデルを取得したら、評価を作成して評価結果を確認することができます。

トピック

- ML モデルインサイト
- バイナリモデルインサイト
- 複数モデルクラスの洞察
- 回帰モデルの洞察
- オーバーフィッティングの防止

- 交差検証
- 評価アラート

ML モデルインサイト

ML モデルを評価すると、Amazon ML は業界標準のメトリクスと多くの洞察を提供して、モデルの予測精度を確認します。Amazon ML では、評価の結果には次のものが含まれます。

- モデルの全体的な成功をレポートする予測精度メトリクス
- 予測精度メトリクスを超えてモデルの正確性を調べるための視覚化
- スコアのしきい値の設定の影響を確認する機能 (バイナリ分類の場合のみ)
- 評価の有効性をチェックする基準に関するアラート

メトリクスと視覚化の選択は、評価している ML モデルのタイプによって異なります。これらの視覚化を確認して、モデルがビジネス要件に合った十分なパフォーマンスを発揮しているかどうかを判断することが重要です。

バイナリモデルインサイト

予測の解釈

多くのバイナリ分類アルゴリズムの実際の出力は予測スコアです。スコアは、指定された観測が正のクラスに属しているというシステムの確実性を示します (実際のターゲット値は 1)。Amazon ML バイナリ分類モデルは、0 から 1 の範囲のスコアを出力します。このスコアのコンシューマーとして、観察を 1 または 0 に分類するかどうかを決定するには、分類しきい値を選択してスコアを解釈するか、カットオフして、それに対するスコアを比較します。カットオフよりも高いスコアを持つ監視はターゲット = 1 として予測されます。カットオフより低いスコアを持つ監視はターゲット = 0 として予測されます。

Amazon ML でのデフォルトのスコアカットオフは 0.5 です。このカットオフをビジネスニーズに合わせて更新することができます。コンソールの可視化を使用して、カットオフの選択がアプリケーションにどのように影響するかを理解することができます。

ML モデルの正確性の測定

Amazon ML は、(Receiver Operating Characteristic) 曲線下面積 (AUC) と呼ばれるバイナリ分類モデルの業界標準の正確性メトリクスを提供します。AUC は、モデルの能力を測定して、正の例につい

てより高いスコアを予測し負の例と比較します。スコアカットオフから独立しているため、しきい値を選択せずに AUC メトリクスからモデルの予測精度を知ることができます。

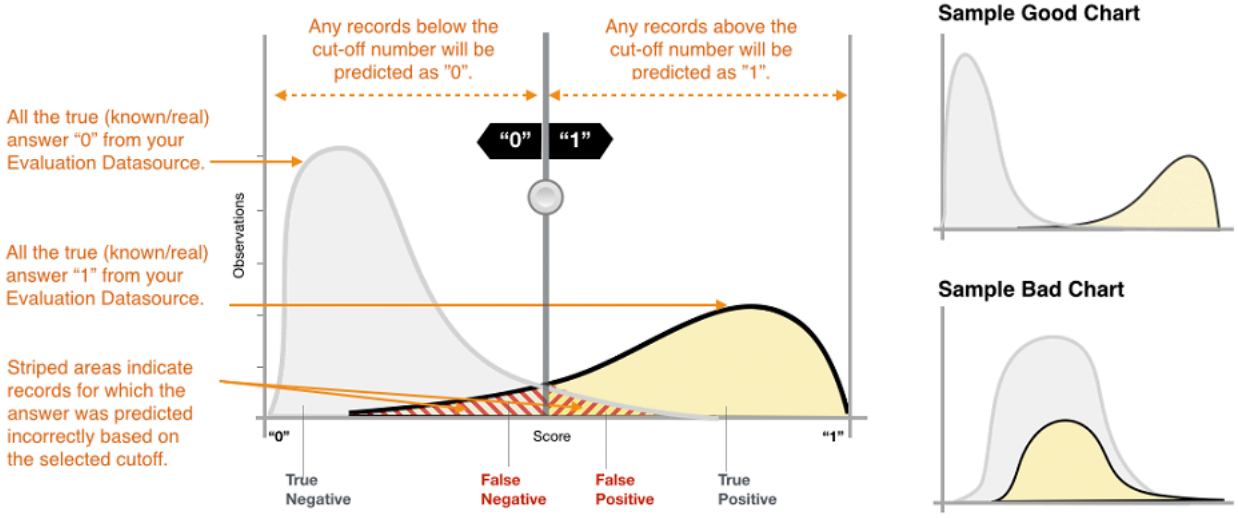
AUC のメトリクスは 0 から 1 の 10 進値を返します。1 に近い AUC 値は、極めて正確な ML モデルであることを示します。0.5 付近の値は、機械学習モデルがランダムな推測を上回っていないことを示します。0 に近い値は一般的ではありません。通常、データに問題があることを示します。基本的に、0 に近い AUC は、ML モデルが正しいパターンを学習したが、現実から反転された予測を行うためにそれらを使用していることを示しています (0 は 1 と予測され、逆も同様)。AUC の詳細については、Wikipedia の「Receiver operating characteristic」のページを参照してください。

バイナリモデルのベースライン AUC メトリクスは 0.5 です。これは 1 または 0 の答えをランダムに予測する仮想 ML モデルの値です。バイナリ ML モデルが価値あるものになるためには、パフォーマンスはこの値よりも優れている必要があります。

パフォーマンスの可視化の使用

ML モデルの正確性を調べるには、Amazon ML コンソールの [Evaluation] (評価) ページのグラフを参照してください。このページには、a) 実際の正 (ターゲットは 1) のスコアのヒストグラムと、b) 評価データの実際の負 (ターゲットは 0) のスコアのヒストグラムの 2 つのヒストグラムが表示されます。

予測精度が良好な ML モデルは、実際の 1 に高いスコアを、実際の 0 に低いスコアを予測します。完全なモデルは、x 軸の 2 つの異なる端に 2 つのヒストグラムを持ち、実際の正がすべて高い得点を示し、実際の負がすべて低い得点を示します。しかし、ML モデルは間違いを引き起こし、典型的なグラフは、2 つのヒストグラムが特定のスコアで重なっていることを示します。極端にパフォーマンスの低いモデルでは、正と負のクラスを区別できず、どちらのクラスもほとんど重複するヒストグラムとなります。



可視化を使用すると、2 つのタイプの正しい予測と 2 つのタイプの誤った予測に分類される予測の数を特定できます。

正しい予測

- 正しい検出 (TP) : Amazon ML はその値を 1 と予測し、真の値は 1 です。
- 正しい非検出 (TN) : Amazon ML はその値を 0 と予測し、真の値は 0 です。

誤った予測

- 誤検出 (FP) : Amazon ML はその値を 1 と予測しますが、真の値は 0 です。
- 検出漏れ (FN) : Amazon ML はその値を 0 と予測しますが、真の値は 1 です。

> **Note**
>
> TP、TN、FP、および FN の数は、選択したスコアのしきい値に依存し、これらの数値のいずれかを最適化することは、他のスコアとのトレードオフを意味します。高い数の TP は通常、高い FP 数および低い数の TN となります。

スコアカットオフの調整

ML モデルは、数値予測スコアを生成し、これらのスコアをバイナリ 0/1 ラベルに変換するカットオフを適用することによって機能します。スコアのカットオフを変更することで、失敗したときにモデルの動作を調整できます。Amazon ML コンソールの [Evaluation] (評価) ページでは、さまざまなスコアカットオフの影響を確認し、モデルに使用するスコアカットオフを保存できます。

スコアのカットオフしきい値を調整するときは、2 種類のエラーのトレードオフを確認します。カットオフを左に移動すると、より正しい検出が得られますが、誤検出エラーの数が増加する可能性もあります。右に移動すると誤検出エラーは少なくなりますが、正しい検出を見逃す可能性があります。予測アプリケーションでは、適切なカットオフのスコアを選択することで、どのような種類のエラーがより許容できるかを判断します。

高度なメトリクスの確認

Amazon ML は、ML モデルの予測精度 (正確性、精度、リコール、および誤検出率) を測定するために、以下の追加のメトリクスを提供します。

Accuracy

Accuracy (ACC) は正しい予測の割合を測定します。範囲は 0 ~ 1 です。値が大きいほど予測精度が良いことを示します。

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

精度

Precision は、正と予測されるこれらの例の中で実際の正の割合を測定します。範囲は 0 ~ 1 です。値が大きいほど予測精度が良いことを示します。

$$Precision = \frac{TP}{TP + FP}$$

リコール

Recall は、正と予測される実際の正の割合を測定します。範囲は 0 ~ 1 です。値が大きいほど予測精度が良いことを示します。

$$Recall = \frac{TP}{TP + FN}$$

誤検出率

誤検出率 (FPR) は、誤ったアラームの割合または正と予測された実際の負の割合を測定します。範囲は 0 ~ 1 です。値が小さいほど予測精度が良いことを示します。

$$FPR = \frac{FP}{FP + TN}$$

ビジネス上の問題によっては、これらのメトリクスの特定のサブセットでうまくいくモデルにもっと興味があるかもしれません。たとえば、2 つのビジネスアプリケーションで、ML モデルの要件が非常に異なる場合があります。

- 一方のアプリケーションでは、正の予測が実際に正 (高精度) であると確認し、いくつかの正な例を負 (中程度のリコール) として誤分類する可能性があります。
- 別のアプリケーションでは、可能な限り多くの正の例を正しく予測する必要があるかもしれないため (高いリコール)、正として間違って分類されるいくつかの負の例を受け入れます (中程度の精度)。

Amazon ML では、先行するいずれかの高度なメトリクスの特定の値に対応するスコアのカットオフを選択できます。また、1 つのメトリクスを最適化する際に生じるトレードオフも示しています。たとえば、高精度に対応するカットオフを選択した場合、通常、それをより低いリコールでトレードオフする必要があります。

> **Note**
>
> 将来の予測を ML モデルで分類するには、スコアカットオフを保存する必要があります。

複数モデルクラスの洞察

予測の解釈

複数クラス分類アルゴリズムの実際の出力は、一連の予測スコアです。スコアは、指定された観測がそれぞれのクラスに属しているというモデルの確実性を示します。バイナリ分類問題の場合とは異なり、予測を行うのにスコアのカットオフを選択する必要はありません。予測される回答は、予測スコアが最も高いクラス (たとえば、ラベル) です。

ML モデルの正確性の測定

複数クラスで使用される一般的なメトリクスは、すべてのクラスで平均した後のバイナリ分類のケースで使用されるメトリクスと同じです。Amazon ML では、マクロ平均 F1 スコアを使用して、複数クラスメトリクスの予測の精度を評価します。

マクロ平均 F1 スコア

F1 スコアは、バイナリメトリクスの正確性とリコールの両方を考慮するバイナリ分類メトリクスです。正確性とリコールを組み合わせた手法です。範囲は 0 ~ 1 です。値が大きいほど予測精度が良いことを示します。

$$F1\ score = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

マクロ平均 F1 スコアは、複数クラスケースのすべてのクラスでの F1 スコアの非加重平均です。評価データセット内でのクラスの発生頻度は考慮されていません。値が大きいほど予測精度が良いことを示します。以下の例に示しているのは、評価データソースの K クラスです。

$$Macro\ average\ F1\ score = \frac{1}{K}\sum_{k=1}^{K} F1\ score\ for\ class\ k$$

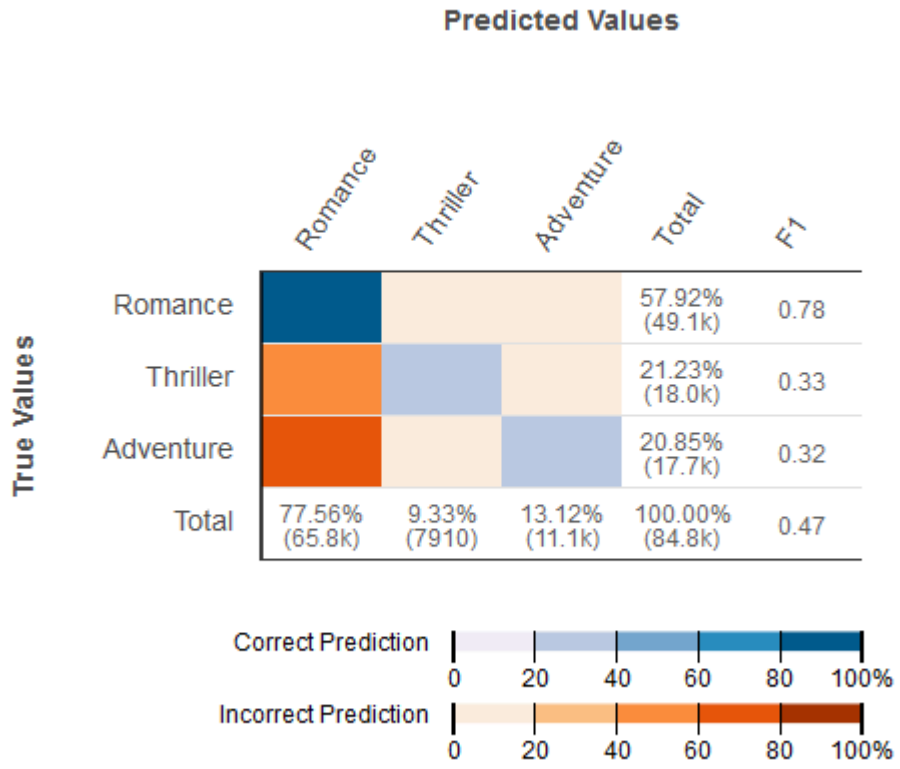
ベースラインマクロ平均 F1 スコア

Amazon ML には、複数クラスモデルのベースラインメトリクスが用意されています。これは、最も頻繁なクラスを回答として予測する仮の複数クラスモデルのマクロ平均 F1 スコアです。たとえば、映画のジャンルを予測するとき、トレーニングデータでの最も一般的なジャンルがロマンスであれば、ベースラインモデルは常にジャンルをロマンスと予測します。ML モデルをこのベースラインと比較することで、常にこの定数を回答として予測する ML モデルよりも自分の ML モデルが優れているかどうかを検証できます。

パフォーマンスの可視化の使用

Amazon ML では、複数クラス分類の予測モデルの精度を可視化する方法として、混合行列を提供しています。混同行列は、観測の予測クラスと真のクラスを比較することによって、各クラスの正しい予測と誤った予測の数または割合 (%) を表に示します。

たとえば、映画をジャンルに分類しようとしている場合、予測モデルはそのジャンル (クラス) がロマンスであると予測するかもしれません。しかし、実際のジャンルはサスペンスである場合があります。複数クラス分類 ML モデルの精度を評価するとき、Amazon ML はこれらの誤分類を識別し、次の図に示すように結果を混同行列に表示します。



次の情報が混合行列に表示されます。

- 各クラスの正しい予測と誤った予測の数: 混同行列の各行は、真のクラスのメトリクスの 1 つに対応します。たとえば、最初の行は、実際にロマンスのジャンルにあるムービーについて、複数クラス ML モデルは、80% 以上のケースで正しい予測をしたことを示しています。20% 以下のケースでジャンルをサスペンスと誤って予測し、20% 以下のケースでアドベンチャーとしました。
- クラス対応の F1 スコア: 最後の列は各クラスの F1 スコアを示しています。
- 評価データでの真のクラス頻度: 最後から 2 番目の列には、評価データセットで、57.92% の評価データが口マンス、21.23% がサスペンス、20.85% がアドベンチャーであることを示しています。
- 評価データの予測クラス頻度: 最後の行は、予測における各クラスの頻度を示しています。観測値の 77.56% は口マンスと予測され、9.33% はサスペンスと予測され、13.12% はアドベンチャーとして予測されています。

Amazon ML コンソールには、混合行列に最大 10 のクラスを、評価データの中で最も頻繁なクラスから最も頻度の低いクラスの順にリストできるビジュアル表示が用意されています。評価データに 10 以上のクラスがある場合は、混合行列の中で最も頻発する上位 9 つのクラスが表示され、他のすべてのクラスは「その他」というクラスにまとめられます。Amazon ML は、複数クラスの可視化ページのリンクから混合行列をすべてダウンロードする機能も提供しています。

回帰モデルの洞察

予測の解釈

回帰 ML モデルの出力は、ターゲットのモデル予測の数値です。たとえば、住宅価格を予測している場合、モデルの予測は 254,013 などの値になります。

> **Note**
>
> 予測の範囲が、トレーニングデータのターゲット範囲と異なる場合があります。たとえば、住宅価格を予測していると仮定します。トレーニングデータに含まれていたターゲット値の範囲は 0 から 450,000 です。予測するターゲットが同じ範囲である必要はなく、大きな正の値 (450,000 より大きい) または負の値 (0 未満) かもしれません。アプリケーションで受け入れる範囲外の予測値が得られた場合の対処方法を計画することは重要です。

ML モデルの正確性の測定

回帰タスクの場合、Amazon ML は業界標準の二乗平均平方根誤差 (RMSE) メトリックスを使用します。これは、予測された数値ターゲットと実際の数値解の間の距離を測定することです (グランドトゥルース)。RMSE の値が小さいほど、モデルの予測の正確性が高くなります。完全に正しい予測モデルでは、RMSE は 0 です。以下の例は、N レコードが保存されている評価データを示します。

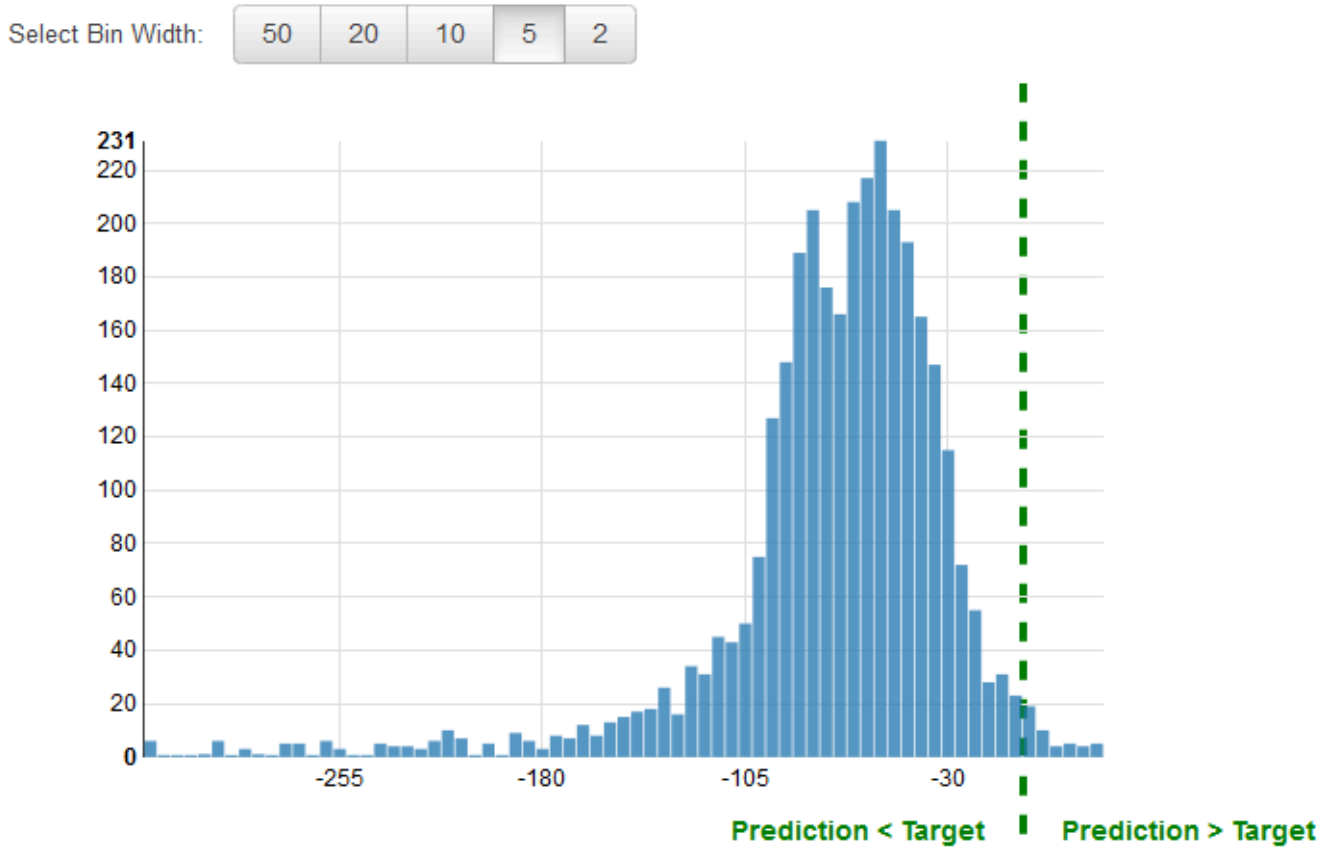
$$RMSE = \sqrt{1/N \sum_{i=1}^{N} (actual\ target - predicted\ target)^2}$$

ベースライン RMSE

Amazon ML には回帰モデルのベースラインメトリックスが用意されています。これは、常にターゲットの平均値を予測の回答とする架空の回帰モデルの RMSE です。たとえば、家の購入者の年齢を予測していて、トレーニングデータの観測の平均年齢が 35 の場合、ベースラインモデルは常に回答として 35 を予測します。ML モデルをこのベースラインと比較することで、常にこの定数を回答として予測する ML モデルよりも自分の ML モデルが優れているかどうかを検証できます。

パフォーマンスの可視化の使用

回帰問題では残余をレビューするのが一般的な方法です。評価データの観測の残余とは、真のターゲットと予測されたターゲットの違いを意味しています。残余は、モデルが予測できないターゲットの部分を表しています。正の残余は、モデルがターゲットを過少評価している (実際のターゲットが予測ターゲットより大きい) ことを示します。負の残余は、モデルがターゲットを過大評価している (実際のターゲットが予測ターゲットより小さい) ことを示します。評価データの残余のヒストグラムが、ゼロを中心とするベル形状で分布している場合、モデルがランダムにミスを犯していて、ターゲット値の特定の範囲で体系的に過大予測または過小予測していないことを示します。残余がゼロを中心としたベル形状にならない場合、モデルの予測エラーに何かの構造が存在しています。モデルに変数を追加すると、現在のモデルでキャプチャしていないパターンをモデルがキャプチャする役に立つかもしれません。次の図に、ゼロが中心とならない残余を示します。



オーバーフィッティングの防止

ML モデルを作成してトレーニングする場合、目標は最良の予測を行うモデルを選択することです。これは、最適な設定 (ML モデル設定またはハイパーパラメータ) でモデルを選択することを意味します。Amazon Machine Learning には、パス数、正規化、モデルサイズ、シャッフルタイプの 4 つのハイパーパラメータを設定できます。ただし、評価データで「最良の」予測パフォーマンスを生成するモデルパラメータ設定を選択すると、モデルがオーバーフィットする可能性があります。モデルがトレーニングと評価のデータソースで発生するパターンを記憶しているが、データのパターンを一般化することができなかった場合、オーバーフィッティングが発生します。これはトレーニングデータに、評価で使用されたすべてのデータが含まれている場合によく発生します。オーバーフィッティングされたモデルは、評価中はうまくいきますが、見えないデータについて正確な予測をすることはできません。

オーバーフィッティングされたモデルを最良のモデルとして選択するのを避けるために、追加のデータを予約して ML モデルのパフォーマンスを検証することができます。たとえば、データをトレーニング用に 60%、評価用に 20%、検証用に 20% に分割することができます。評価データに適したモデルパラメータを選択した後、検証データを使用して 2 番目の評価を実行して、ML モデルが検証

データに対してどれだけうまく実行するかを確認します。モデルが検証データに対する期待値を満たしていれば、モデルはデータにオーバーフィッティングしていません。

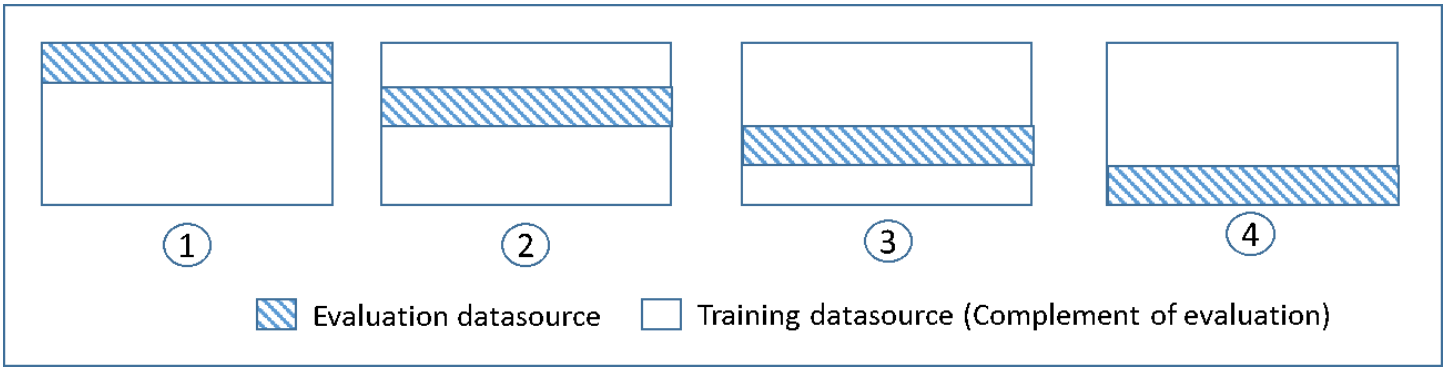
検証のために第 3 のデータセットを使用すると、適切な ML モデルパラメータを選択でき、オーバーフィッティングを防止できます。しかし、評価と検証の両方のためにトレーニングプロセスからのデータを提出すると、トレーニングのために利用可能なデータは少なくなります。トレーニングのためにはできるだけ多くのデータを使用するのが常にベストであるため、これは、小さなデータセットの場合は特に問題です。この問題を解決するには、相互検証を実行できます。相互検証の詳細については、交差検証 を参照してください。

交差検証

交差検証は、使用可能な入力データのサブセットでいくつかの ML モデルをトレーニングし、また、それらをデータの補完サブセットで評価することにより ML モデルを評価する手法です。交差検証を使用して、パターン生成の失敗などのオーバーフィットを検出します。

Amazon ML は、K 分割交差検証のメソッドを使用して交差検証を実行できます。K 分割交差検証では、入力データを K 個のデータのサブセットに分割します。 1 つ以外すべてのサブセット (k-1) で ML モデルをトレーニングし、トレーニングに使用しなかったサブセットでモデルを評価します。このプロセスが k 回、評価のために取り分けられた (トレーニングから除外された) 毎回異なるサブセットで繰り返されます。

次の図では、4 分割交差検証の時に作成されトレーニングされた 4 つのモデルそれぞれのために生成されたトレーニングサブセットと補完的な評価サブセットの例を示しています。モデル 1 は、最初の 25% のデータを評価に、残りの 75% をトレーニングに使用しています。モデル 2 は、2 番目のサブセットの 25 パーセント (25% ~ 50%) を評価に、残りの 3 つのサブセットをトレーニングに使用していて、以降同様に続きます。



各モデルは、補完的なデータソースを使用してトレーニングされ、評価されます。評価データソースのデータには、とレーニンデータソースにないデータのみがすべて含まれています。これらの各サ

ブセットのデータソースを作成するには、DataRearrangement、createDatasourceFromS3、および createDatasourceFromRedShift API の createDatasourceFromRDS パラメータを使用します。DataRearrangement パラメータでは、各セグメントの開始位置と終了位置を指定することで、データソースに含めるデータのサブセットを指定します。4k 分割の交差検証に必要な補完データソースを作成するには、以下の例に示すように DataRearrangement パラメータを指定します。

モデル 1:

評価のためのデータソース

```
{"splitting":{"percentBegin":0, "percentEnd":25}}
```

トレーニングのためのデータソース

```
{"splitting":{"percentBegin":0, "percentEnd":25, "complement":"true"}}
```

モデル 2:

評価のためのデータソース

```
{"splitting":{"percentBegin":25, "percentEnd":50}}
```

トレーニングのためのデータソース

```
{"splitting":{"percentBegin":25, "percentEnd":50, "complement":"true"}}
```

モデル 3:

評価のためのデータソース

```
{"splitting":{"percentBegin":50, "percentEnd":75}}
```

トレーニングのためのデータソース

```
{"splitting":{"percentBegin":50, "percentEnd":75, "complement":"true"}}
```

モデル 4:

評価のためのデータソース

```
{"splitting":{"percentBegin":75, "percentEnd":100}}
```

トレーニングのためのデータソース

```
{"splitting":{"percentBegin":75, "percentEnd":100, "complement":"true"}}
```

4 分割交差検証を実行すると、4 つのモデル、モデルをトレーニングするための 4 つのデータソース、モデルを評価するための 4 つのデータソース、および各モデルに 1 つずつの 4 つの評価が生成されます。Amazon ML では、評価ごとにモデルパフォーマンスメトリクスを生成します。たとえば、バイナリ分類問題の 4 分割交差検証では、それぞれの評価は曲線下面積 (AUC) メトリクスを報告します。全体的なパフォーマンスの測定値を取得するには、4 つの AUC のメトリクスの平均を計算します。AUC メトリクスの詳細については、「ML モデルの正確性の測定」を参照してください。

交差検証とモデルスコアの平均を作成する方法を示すサンプルコードについては、「Amazon ML サンプルコード」を参照してください。

モデルの調整

モデルの交差検証をした後は、モデルのパフォーマンスが期待にそぐわない場合に、次のモデルの設定を調整できます。オーバーフィットの詳細については、「モデルフィット: アンダーフィットとオーバーフィット」を参照してください。正則化の詳細については、「正則化」を参照してください。正則化の設定の変更の詳細については、「カスタムオプションで ML モデルを作成する」を参照してください。

評価アラート

Amazon ML は、モデルを正しく評価したかどうかを検証するための洞察を与えます。評価でいずれかの検証基準が満たされない場合、Amazon ML コンソールは、違反した検証基準を次のように表示することによって警告します。

- 保有データで ML モデルの評価が行われました

  Amazon ML はトレーニングと評価に同じデータソースを使用する場合に警告します。Amazon ML を使用してデータを分割する場合は、この検証基準を満たします。Amazon ML を使用してデータを分割しない場合は、トレーニングデータソース以外のデータソースで ML モデルを評価していることを確認します。
- 予測モデルの評価に十分なデータが使用されました

Amazon ML は、評価データの観測数/レコード数がトレーニングデータソースの観測数の 10% 未満である場合に警告します。モデルを適切に評価するには、十分に大きなデータサンプルを提供することが重要です。この基準は、使用しているデータが少なすぎるかどうかをチェックし、知らせます。ML モデルを評価するために必要なデータの量は主観的です。ここでは、より良い基準がない場合に 10% が一時的なものとして選択されています。

- 一致したスキーマ

  Amazon ML はトレーニングと評価のデータソースのスキーマが同じでない場合に警告します。評価データソースに存在しない特定の属性がある場合、または追加の属性がある場合、Amazon ML はこのアラートを表示します。

- 評価ファイルのすべてのレコードが予測モデルパフォーマンスの評価に使用されました

  評価のために提供されたすべてのレコードがモデルを評価するために実際に使用されたかどうかを知ることは重要です。評価データソースの一部のレコードが無効で、精度メトリクス計算に含まれていない場合、Amazon ML は警告を表示します。例えば、評価データソースの観測値の一部にターゲット変数がない場合、Amazon ML は、これらの観測値に対する ML モデルの予測が正しいかどうかをチェックできません。この場合、不足しているターゲット値を持つレコードは無効と見なされます。

- ターゲット変数の分布

  Amazon ML はトレーニングと評価のデータソースからターゲット属性の分布を表示するので、ターゲットが両方のデータソースで同様に分布しているかどうかを確認できます。モデルが、評価データ上のターゲット分布とは異なるターゲット分布のあるトレーニングデータでトレーニングされた場合、非常に異なる統計を持つデータに関して計算されているので、評価の質が損なわれる可能性があります。トレーニングおよび評価データソースでデータが同じように分布していて、予測を作成するときにモデルが直面するデータにできる限りデータセットを似せるのが最善です。

  このアラートがトリガーされる場合は、ランダムスプリット戦略を使用して、データをトレーニングおよび評価データソースに分割してみてください。まれに、データをランダムに分割してもターゲット分布の違いについてこの警告が誤って出されることがあります。Amazon ML は、おおまかなデータ統計を使用してデータの分布を評価していて、このアラートを誤ってトリガーすることがあります。

予測の生成と解釈

Amazon ML には、非同期 (バッチベース) と同期 (一度に 1 つ) という予測を生成する 2 つのメカニズムがあります。

多数の観測値を持ち、観測値の予測をまとめて取得する場合は、非同期予測、またはバッチ予測を使用します。プロセスはデータソースを入力として使用し、選択した S3 バケットに格納された .csv ファイルに予測を出力します。予測結果にアクセスする前に、バッチ予測プロセスが完了するまで待つ必要があります。Amazon ML がバッチファイルで処理できるデータソースの最大サイズは 1 TB (約 1 億レコード) です。データソースが 1 TB より大きい場合、ジョブは失敗し、Amazon ML は エラーコードを返します。これを防ぐには、データを複数のバッチに分割します。レコードが通常長い場合、1 億のレコードが処理される前に 1 TB の制限に達します。この場合、AWS support に連絡して、バッチ予測のジョブサイズを増やすことをお勧めします。

低いレイテンシーで予測を取得する場合は、同期、またはリアルタイム予測、を使用します。リアルタイム予測 API は、JSON 文字列としてシリアル化された単一の入力観測を受け入れ、予測と関連するメタデータを API 応答の一部として同期的に返します。API を複数回呼び出し、並行して同期予測を取得することができます。リアルタイム予測 API のスループット制限の詳細については、Amazon ML API リファレンスのリアルタイム予測制限を参照してください 。

トピック

- バッチ予測の作成
- バッチ予測メトリクスの確認
- バッチ予測出力ファイルの読み込み
- リアルタイム予測のリクエスト

バッチ予測の作成

バッチ予測を作成するには、Amazon Machine Learning (Amazon ML) コンソールまたは API を使用して `BatchPrediction` オブジェクトを作成します。`BatchPrediction` オブジェクトは、ML モデルと一連の入力観測を使用して Amazon ML が生成する一連の予測を記述します。`BatchPrediction` オブジェクトを作成すると、Amazon ML は予測を計算する非同期ワークフローを開始します。

バッチ予測を取得するために使用するデータソースと、予測をクエリする ML モデルを訓練するために使用したデータソースに同じスキーマを使用する必要があります。唯一の例外は、Amazon ML は

ターゲットを予測するため、バッチ予測のデータソースにターゲット属性を含める必要がないことです。ターゲット属性を指定すると、Amazon ML はその値を無視します。

バッチ予測の作成 (コンソール)

Amazon ML コンソールを使用してバッチ予測を作成するには、バッチ予測の作成ウィザードを使用します。

バッチ予測を作成するには (コンソール)

1. AWS Management Console にサインインし、Amazon Machine Learning コンソール (https://console.aws.amazon.com/machinelearning/) を開きます。
2. Amazon ML ダッシュボードの [Objects] (オブジェクト) の下で、[Create new...] (新規作成...)、[Batch prediction] (バッチ予測) の順に選択します。
3. バッチ予測の作成に使用する Amazon ML モデルを選択します。
4. このモデルを使用することを確認するには、[続行] を選択します。
5. 予測を作成するデータソースを選択します。データソースに、ターゲット属性を含める必要はありませんが、モデルと同じスキーマを持たなければなりません。
6. [Continue] (続行) をクリックします。
7. [S3 の宛先] には、S3 バケットの名前を入力します。
8. [Review] (レビュー) を選択します。
9. 設定を確認し、[バッチ予測の生成] を選択します。

バッチ予測の作成 (API)

Amazon ML API を使用して `BatchPrediction` オブジェクトを作成するには、次のパラメータを指定する必要があります。

データソース ID

予測が必要な観測値を指し示すデータソースの ID。たとえば、`s3://examplebucket/input.csv` というファイル内のデータの予測を行う場合は、データファイルを指すデータソースオブジェクトを作成し、このパラメータを使用してそのデータソースの ID を渡します。

BatchPrediction ID

バッチ予測に割り当てる ID。

ML モデル ID

Amazon ML が予測を照会する ML モデルの ID。

出力 URI

予測の出力を格納する S3 バケットの URI。Amazon ML には、このバケットにデータを書き込む権限が必要です。

`OutputUri` パラメータは、次の例に示すように、スラッシュ ("/") で終わる S3 パスを参照する必要があります。

s3://examplebucket/examplepath/

S3 アクセス権限の設定については、 Amazon S3 に予測を出力するために Amazon ML のアクセス許可を得る を参照してください。

(オプション) BatchPrediction 名

(オプション) バッチ予測のための人間が読み取れる名前。

バッチ予測メトリクスの確認

Amazon Machine Learning (Amazon ML) がバッチ予測を作成したら、2 つのメトリクス、`Records seen` および `Records failed to process` が提供されます。`Records seen` は、Amazon ML がバッチ予測を実行したときに調べたレコードの数を示します。`Records failed to process` は、Amazon ML が処理できなかったレコードの数を示します。

Amazon ML が失敗したレコードを処理できるようにするには、データソースの作成に使用されたデータのレコードのフォーマットをチェックし、必要なすべての属性が存在し、すべてのデータが正しいことを確認します。データを修正したら、バッチ予測を再作成するか、または、失敗したレコードで新しいデータソースを作成した後、新しいデータソースを使用して新しいバッチ予測を作成することができます。

バッチ予測メトリクスの確認 (コンソール)

Amazon ML コンソールでメトリクスを表示するには、[Batch prediction summary] (バッチ予測の概要) ページを開き、[Processed Info] (処理済みの情報) セクションを確認します。

バッチ予測メトリクスと詳細の確認 (API)

Amazon ML API を使用して、レコードメトリクスを含む、BatchPrediction オブジェクトの詳細を取得できます。Amazon ML では、以下のバッチ予測 API コールを提供します。

- CreateBatchPrediction
- UpdateBatchPrediction
- DeleteBatchPrediction
- GetBatchPrediction
- DescribeBatchPredictions

詳細については、「Amazon ML API リファレンス」を参照してください。

バッチ予測出力ファイルの読み込み

バッチ予測出力ファイルを取得するには、次の手順を実行します。

1. バッチ予測のマニフェストファイルを見つけます。
2. マニフェストファイルを読み、出力ファイルの場所を決定します。
3. 予測を含む出力ファイルを取得します。
4. 出力ファイルの内容を解釈します。コンテンツは、予測を生成するために使用された ML モデルのタイプによって異なります。

次のセクションでは、手順について詳しく説明します。

バッチ予測のマニフェストファイルを見つける

バッチ予測のマニフェストファイルには、入力ファイルを予測出力ファイルにマップする情報が含まれています。

マニフェストファイルを見つけるには、バッチ予測オブジェクトを作成したときに指定した出力場所から開始します。完了したバッチ予測オブジェクトをクエリして、Amazon ML API または https://console.aws.amazon.com/machinelearning/ のいずれかを使用して、このファイルの S3 の場所を取得できます。

マニフェストファイルは、出力場所とマニフェストファイルの名前 (拡張子 /batch-prediction/ が追加されたバッチ予測の ID) に追加される静的な文字列 .manifest で構成されるパスの出力場所にあります。

たとえば、ID bp-example でバッチ予測オブジェクトを作成し、S3 の場所 s3://examplebucket/output/ を出力場所として指定すると、ここにマニフェストファイルを見つけられます。

s3://examplebucket/output/batch-prediction/bp-example.manifest

マニフェストファイルの読み込み

マニフェストファイルのコンテンツは JSON マップとしてエンコードされます。キーは S3 入力データファイルの名前の文字列で、値は関連するバッチ予測結果ファイルの文字列です。各入力/出力ファイルのペアには 1 つのマッピング行があります。これまでの例を引き続き使用します。BatchPrediction オブジェクトの作成の入力が、s3://examplebucket/input/ にある data.csv という単一のファイルで構成されている場合、次のようなマッピング文字列が表示されることがあります。

```
{"s3://examplebucket/input/data.csv":"
s3://examplebucket/output/batch-prediction/result/bp-example-data.csv.gz"}
```

BatchPrediction オブジェクトの作成への入力が、data1.csv、data2.csv、および data3.csv という 3 つのファイルで構成され、それらがすべて S3 の場所 s3://examplebucket/input/ に格納されている場合は、マッピング文字列は次のようになります。

```
{"s3://examplebucket/input/data1.csv":"s3://examplebucket/output/batch-prediction/
result/bp-example-data1.csv.gz",

"s3://examplebucket/input/data2.csv":"
s3://examplebucket/output/batch-prediction/result/bp-example-data2.csv.gz",

"s3://examplebucket/input/data3.csv":"
s3://examplebucket/output/batch-prediction/result/bp-example-data3.csv.gz"}
```

バッチ予測出力ファイルの取得

マニフェストマッピングから取得した各バッチ予測ファイルをダウンロードし、ローカルで処理することができます。ファイル形式は CSV で、gzip アルゴリズムで圧縮されています。そのファイル内には、対応する入力ファイルの入力観測ごとに 1 行があります。

予測をバッチ予測の入力ファイルと結合するには、2 つのファイルのレコードごとの簡単なマージを実行します。バッチ予測の出力ファイルには、常に予測入力ファイルと同じ数のレコードが同じ順序で含まれています。入力観測が処理に失敗し、予測を生成できない場合、バッチ予測の出力ファイルは、対応する場所に空白行を持ちます。

バイナリ分類 ML モデルのバッチ予測ファイルのコンテンツの解釈

バイナリ分類モデルのバッチ予測ファイルの列は、bestAnswer および score と呼ばれます。

bestAnswer 列には、予測スコアをカットオフスコアと比較して得られた予測ラベル (「1」または「0」) が含まれます。カットオフスコアの詳細については、「スコアカットオフの調整」を参照してください。ML モデルのカットオフスコアは Amazon ML API または Amazon ML コンソールのモデル評価機能のいずれかを使用して設定します。カットオフスコアを設定しない場合、Amazon ML ではデフォルト値の 0.5 が使用されます。

スコア列には、この予測のために ML モデルによって割り当てられた未加工の予測スコアが含まれています。Amazon ML はロジスティック回帰モデルを使用するため、このスコアは真の (「1」) 値に対応する監視の確率をモデル化しようとします。スコアは科学的表記で報告されるので、次の例の最初の行では、値 8.7642E-3 は 0.0087642 に等しいことに注意してください。

たとえば、ML モデルのカットオフスコアが 0.75 の場合、バイナリ分類モデルのバッチ予測出力ファイルのコンテンツは次のようになります。

```
bestAnswer,score

0,8.7642E-3

1,7.899012E-1


0,6.323061E-3

0,2.143189E-2


1,8.944209E-1
```

入力ファイルの 2 番目と 5 番目の観測値は 0.75 を超える予測スコアを受けているため、これらの観測値の bestAnswer 列は値「1」を示し、他の観測値は値「0」を示します。

複数クラスの分類 ML モデルのバッチ予測ファイルのコンテンツの解釈

複数クラスモデルのバッチ予測ファイルには、トレーニングデータに含まれるクラスごとに 1 つの列が含まれています。バッチ予測ファイルのヘッダー行に列名が表示されます。

複数クラスモデルから予測をリクエストすると、Amazon ML は、入力ファイルの各観測について、(入力データセットで定義されている各クラスに 1 つずつ) いくつかの予測スコアを計算します。これは、他のクラスと対照して、この観察がこのクラスに当てはまる確率 (0 と 1 の間で測定される) を尋ねているのと同等です。各スコアは、「観察がこのクラスに属する確率」と解釈することができます。予測スコアは、任意のクラスに属する観測の基盤となる確率をモデル化するため、行全体のすべての予測スコアの合計は 1 です。モデルの予測クラスとして 1 つのクラスを選択する必要があります。最も一般的には、最も高い確率を持つクラスをベストアンサーとして選択します。

たとえば、1~5 の星スケールに基づいて、製品の顧客の評価を予測しようとします。クラスの名前が 1_star、2_stars、3_stars、4_stars、および 5_stars である場合、複数クラスの予測出力ファイルは次のようになります。

```
1_star, 2_stars, 3_stars, 4_stars, 5_stars

8.7642E-3, 2.7195E-1, 4.77781E-1, 1.75411E-1, 6.6094E-2

5.59931E-1, 3.10E-4, 2.48E-4, 1.99871E-1, 2.39640E-1

7.19022E-1, 7.366E-3, 1.95411E-1, 8.78E-4, 7.7323E-2

1.89813E-1, 2.18956E-1, 2.48910E-1, 2.26103E-1, 1.16218E-1

3.129E-3, 8.944209E-1, 3.902E-3, 7.2191E-2, 2.6357E-2
```

この例では、最初の観測値は 3_stars クラスの予測スコア (予測スコア = 4.77781E-1) が最も高いため、結果は 3_stars クラスがこの観測のベストアンサーであると解釈します。予測スコアは科学的表記で報告されるので、4.77781E-1 の予測スコアは 0.477781 に等しいことに注意してください。

確率が最も高いクラスを選択したくない場合があります。たとえば、最小しきい値を確立するため、その値以下では、予測スコアが最も高い場合でもそのクラスをベストアンサーとみない場合があります。映画をジャンルに分類し、そのジャンルをベストアンサーと宣言する前に予測スコアを少なくとも 5E-1 にしたいとします。コメディに対して 3E-1、ドラマに対して 2.5E-1、ドキュメンタリーに対して 2.5E-1、およびアクション映画に対して 2E-1 の予測スコアを得るとします。この場合、ML モデルはコメディが最も可能性の高い選択肢だと予測しますが、それをベストアンサーとして選択し

ないことにします。予測スコアはどれもベースライン予測スコア 5E-1 を上回っていないため、その予測がそのジャンルを確実に予測するには不十分であると判断し、何か他のものを選択することになります。アプリケーションでは、このムービーのジャンルフィールドは「不明」として扱われます。

回帰 ML モデルのバッチ予測ファイルのコンテンツの解釈

回帰モデルのバッチ予測ファイルには、score という名前の列が 1 つ含まれています。この列には、入力データ内の各観測の未加工の数値予測が含まれます。値は科学的表記で報告されるので、-1.526385E1 という score は次の例の最初の行で -15.26835 に等しくなります。

この例では、回帰モデルで実行されるバッチ予測の出力ファイルを示しています。

```
score

-1.526385E1

-6.188034E0

-1.271108E1

-2.200578E1

8.359159E0
```

リアルタイム予測のリクエスト

リアルタイム予測は、Amazon Machine Learning (Amazon ML) への同期呼び出しです。Amazon ML がリクエストを受け取ったときに予測が作成され、すぐにレスポンスが返されます。リアルタイム予測は一般的に、インタラクティブなウェブアプリケーション、モバイルアプリケーション、デスクトップアプリケーションで予測機能を有効にするのに使用されます。低レイテンシー `Predict` API を使用して、Amazon ML で作成された ML モデルに対してリアルタイム予測のためのクエリを実行します。この `Predict` オペレーションでは、リクエストペイロードで 1 つの入力観測を受け入れ、レスポンスで同期的に予測を返します。これは、入力観測の位置を示す Amazon ML の ID データソースオブジェクトで呼び出され、すべての観測の予測を含むファイルの URI を非同期に返す、バッチ予測 API とは別のものです。Amazon ML はほとんどのリアルタイム予測リクエストに 100 ミリ秒以内に応答します。

Amazon ML コンソールの料金を発生させずにリアルタイム予測を試すことができます。リアルタイム予測を使用すると決定した場合、最初にリアルタイム予測生成のエンドポイントを作成します。こ

れは Amazon ML コンソールまたは `CreateRealtimeEndpoint` API を使用して実行できます。エンドポイントの作成後、リアルタイム予測 API を使用してリアルタイム予測を生成します。

> **Note**
>
> モデルのリアルタイムエンドポイントを作成した後、モデルのサイズに応じたキャパシティー予約の料金が発生し始めます。詳細については、「 の料金」を参照してください。リアルタイムエンドポイントをコンソールで作成する場合、コンソールにはエンドポイントで継続的に発生する予想請求額の内訳が表示されます。そのモデルによるリアルタイム予測の取得が不要になった場合に料金を発生させないためには、コンソールまたは `DeleteRealtimeEndpoint` オペレーションを使用して、リアルタイムエンドポイントを削除します。

`Predict` のリクエストとレスポンスの例については、「Amazon Machine Learning API リファレンス」の「Predict」を参照してください。自分のモデルを使用する正確なレスポンス形式の例を参照するには、「リアルタイム予測の試用」を参照してください。

トピック

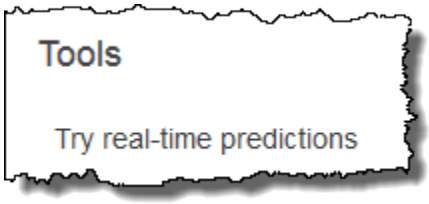
- リアルタイム予測の試用
- リアルタイムエンドポイントの作成
- リアルタイム予測エンドポイント (コンソール) を見つける
- リアルタイム予測エンドポイント (API) を見つける
- リアルタイム予測リクエストの作成
- リアルタイムエンドポイントの削除

リアルタイム予測の試用

リアルタイム予測を有効にするかどうかを判断するのを助けるために、リアルタイム予測エンドポイントの設定に関係する追加料金を発生させずに、Amazon ML で単一のデータレコードの予測の生成を実際に試すことができます。リアルタイム予測をテストするには、ML モデルを持っている必要があります。大きな規模でリアルタイム予測を作成するには、「Amazon Machine Learning API リファレンス」の Predict API を使用します。

リアルタイム予測を試用するには

1. AWS Management Console にサインインし、Amazon Machine Learning コンソール (https://console.aws.amazon.com/machinelearning/) を開きます。
2. ナビゲーションバーの [Amazon Machine Learning] ドロップダウンで [ML モデル] を選択します。
3. チュートリアルから `Subscription propensity model` など、リアルタイム予測に試用するモデルを選択します。
4. [ML モデルレポート] ページで、[予測] から [概要] を選択し、次に [リアルタイム予測の試用] を選択します。



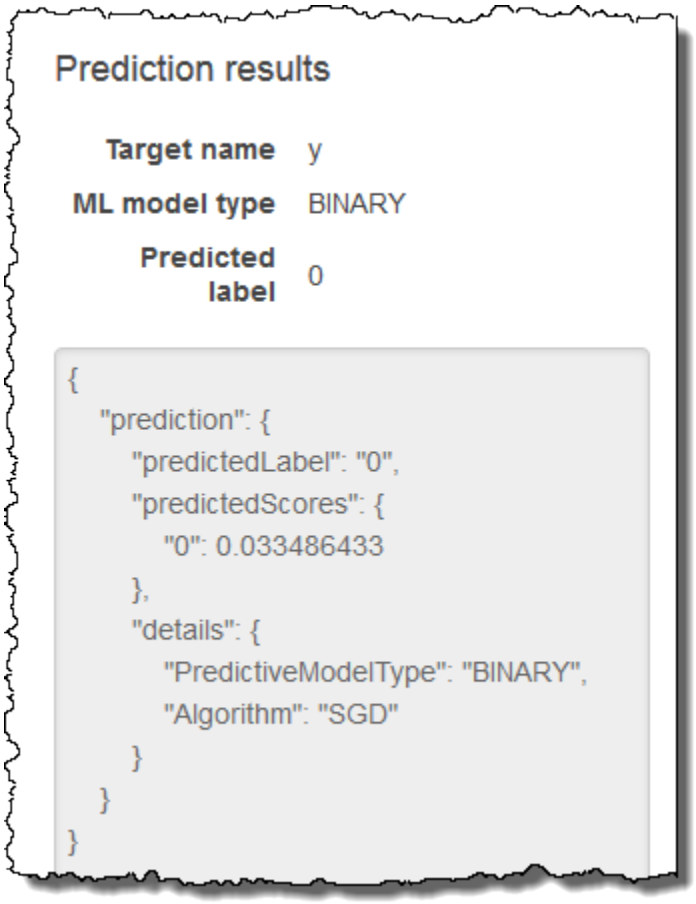
   Amazon ML は、Amazon ML がモデルのトレーニングに使用したデータレコードに含まれていた変数のリストを表示します。
5. 続行するには、フォームの各フィールドにデータを入力するか、1 つのデータレコードを CSV 形式でテキストボックスに貼り付けます。

   フォームを使用するには、各 [値] フィールドに、リアルタイム予測のテストに使用したいデータを入力します。入力しているデータレコードに 1 つまたは複数のデータ属性の値が含まれていない場合、入力フィールドは空白のままにします。

   データレコードを提供する場合、[レコードの貼り付け] を選択します。テキストフィールドに CSV 形式の 1 つのデータ行を貼り付け、[Submit] (送信) を選択します。Amazon ML が [Value] (値) フィールドに自動的に入力します。

   > **Note**
   >
   > データレコードのデータの列の数はトレーニングデータと同じで、同じ順序に配置されている必要があります。唯一の例外は、ターゲット値を省略する必要があることです。ターゲット値を含めた場合、Amazon ML はそれを無視します。
6. ページの下部で、[予測の作成] を選択します。Amazon ML はすぐに予測を返します。

[予測結果] ペインに、Predict API コールが返した予測オブジェクトに加えて、ML モデルタイプ、ターゲット変数の名前、予測されたクラスまたは値が表示されます。結果の解釈の詳細については、「バイナリ分類 ML モデルのバッチ予測ファイルのコンテンツの解釈」を参照してください。



リアルタイムエンドポイントの作成

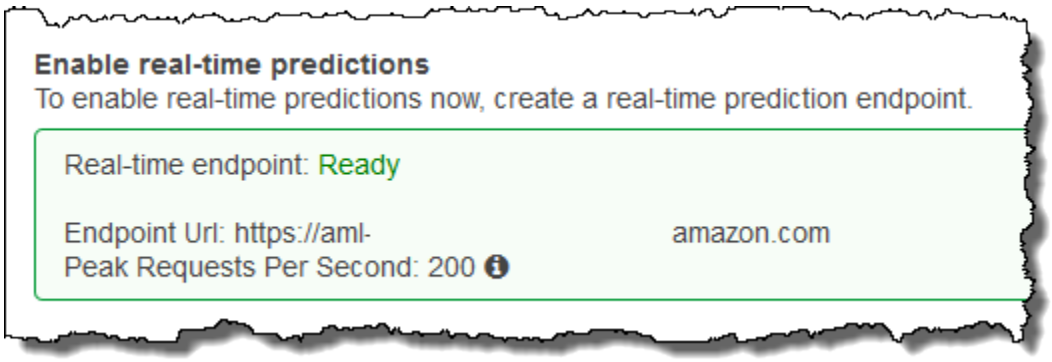
リアルタイム予測を生成するには、リアルタイムエンドポイントを作成する必要があります。リアルタイムエンドポイントを作成するには、リアルタイム予測を生成する ML モデルを既に持っている必要があります。リアルタイムエンドポイントは、Amazon ML コンソールを使用するか、CreateRealtimeEndpoint API を呼び出して作成できます。CreateRealtimeEndpoint API の使用の詳細については、「Amazon Machine Learning API リファレンス」の「https://docs.aws.amazon.com/machine-learning/latest/APIReference/API_CreateRealtimeEndpoint.html」を参照してください。

リアルタイムエンドポイントを作成するには

1. AWS Management Console にサインインし、Amazon Machine Learning コンソール (https://console.aws.amazon.com/machinelearning/) を開きます。
2. ナビゲーションバーの [Amazon Machine Learning] ドロップダウンで [ML モデル] を選択します。
3. リアルタイム予測を生成したいモデルを選択します。
4. [ML モデルの要約] ページの [予測] で、[リアルタイムエンドポイントの作成] を選択します。

   リアルタイム予測の課金方法を説明するダイアログボックスが表示されます。
5. [Create] (作成) を選択します。リアルタイムエンドポイントのリクエストが Amazon ML に送信され、キューに入ります。リアルタイムエンドポイントのステータスは [更新中] になります。

6. リアルタイムエンドポイントの準備ができたら、ステータスは [Ready] (準備完了) に変化し、Amazon ML はエンドポイントの URL を表示します。エンドポイントの URL を使用して、`Predict` API のリアルタイム予測リクエストを作成します。`Predict` API の使用の詳細については、「Amazon Machine Learning API リファレンス」の「https://docs.aws.amazon.com/machine-learning/latest/APIReference/API_Predict.html」を参照してください。



リアルタイム予測エンドポイント (コンソール) を見つける

Amazon ML コンソールを使用して ML モデルの URL エンドポイントを見つけるには、モデルの [ML model summary] (ML モデルの要約) ページに移動します。

リアルタイムエンドポイントの URL を見つけるには

1. AWS Management Console にサインインし、Amazon Machine Learning コンソール (https://console.aws.amazon.com/machinelearning/) を開きます。
2. ナビゲーションバーの [Amazon Machine Learning] ドロップダウンで [ML モデル] を選択します。
3. リアルタイム予測を生成したいモデルを選択します。
4. [ML モデルの要約] ページで、[予測] セクションが表示されるまで下へスクロールします。
5. モデルのエンドポイントの URL は [リアルタイム予測] に表示されます。リアルタイム予測の呼び出でこの URL を、[エンドポイント URL] URL として使用します。エンドポイントを使用して予測を生成する方法については、「Amazon Machine Learning API リファレンス」の「https://docs.aws.amazon.com/machine-learning/latest/APIReference/API_Predict.html」を参照してください。

リアルタイム予測エンドポイント (API) を見つける

`CreateRealtimeEndpoint` オペレーションを使用してリアルタイムエンドポイントを作成すると、レスポンスで URL とエンドポイントのステータスが返されます。コンソールを使用してリアルタイムエンドポイントを作成した場合、または以前に作成したエンドポイントの URL や状態を取得する場合、リアルタイム予測用にクエリしたいモデルの ID を指定して `GetMLModel` オペレーションを呼び出します。エンドポイント情報は、レスポンスの `EndpointInfo` セクションに含まれています。リアルタイムエンドポイントが関連付られているモデルの場合、`EndpointInfo` は次のようになります。

```
"EndpointInfo":{
    "CreatedAt": 1427864874.227,
    "EndpointStatus": "READY",
    "EndpointUrl": "https://endpointUrl",
    "PeakRequestsPerSecond": 200
}
```

リアルタイムエンドポイントがないモデルは次のように返します。

```
EndpointInfo":{
    "EndpointStatus": "NONE",
    "PeakRequestsPerSecond": 0
}
```

リアルタイム予測リクエストの作成

サンプル Predict リクエストペイロードは次のようになります。

```
{
    "MLModelId": "model-id",
    "Record":{
        "key1": "value1",
        "key2": "value2"
    },
    "PredictEndpoint": "https://endpointUrl"
}
```

[PredictEndpoint] フィールドは EndpointInfo 構造の [EndpointUrl] フィールドに対応している必要があります。Amazon ML は、リアルタイム予測フリートの適切なサーバーにリクエストをルーティングするためにこのフィールドを使用します。

MLModelId は、以前にトレーニングを受けた、リアルタイムエンドポイントがあるモデルの識別子です。

Record は変数名と変数値のマップです。各ペアは観測を表します。Record マップには、Amazon ML モデルへの入力が含まれます。これは、ターゲット変数がないトレーニングデータセットの 1 行のデータに相当します。トレーニングデータの値のタイプにかかわらず、Record は文字列から文字列へのマッピングを含みます。

> **Note**
>
> 値を持っていない変数を省略できますが、その場合予測精度が低下することがあります。変数が多いほど、モデルはより正確になります。

Predict リクエストによって返されるレスポンスのフォーマットは、予測のためにクエリされているモデルのタイプによって異なります。いずれの場合も、[details] フィールドには予測リクエストに関する情報が保存され、特に [PredictiveModelType] フィールドにはモデルタイプが含まれます。

以下はバイナリモデルのレスポンスの例です。

```
{
    "Prediction":{
```

```
        "details":{
            "PredictiveModelType": "BINARY"
        },
        "predictedLabel": "0",
        "predictedScores":{
            "0": 0.47380468249320984
        }
    }
}
```

[predictedLabel] フィールドに予測されたラベルが含まれていることに注意してください。この場合は 0 になります。Amazon ML は、予測スコアを分類カットオフと比較することで、予測されたラベルを計算します。

- ML モデルに現在関連付けられている分類カットオフは、ScoreThreshold フィールドを GetMLModel オペレーションのレスポンスで調べることで、または Amazon ML コンソールでモデル情報を表示して取得できます。スコアしきい値を設定しない場合、Amazon ML はデフォルト値の 0.5 を使用します。
- バイナリ分類モデルの正確な予測スコアを取得するには、predictedScores マップを調べます。このマップ内では、予測されたラベルは正確な予測スコアとペアになっています。

バイナリ予測の詳細については、「予測の解釈」を参照してください。

以下は回帰モデルのレスポンスの例です。予測された数値が [predictedValue] フィールドにあることに注意してください。

```
{
    "Prediction":{
        "details":{
            "PredictiveModelType": "REGRESSION"
        },
        "predictedValue": 15.508452415466309
    }
}
```

以下は複数クラスモデルのレスポンスの例です。

```
{
    "Prediction":{
        "details":{
```

```
            "PredictiveModelType": "MULTICLASS"
        },
        "predictedLabel": "red",
        "predictedScores":{
            "red": 0.12923571467399597,
            "green": 0.08416014909744263,
            "orange": 0.22713537514209747,
            "blue": 0.1438363939523697,
            "pink": 0.184102863073349,
            "violet": 0.12816807627677917,
            "brown": 0.10336143523454666
        }
    }
}
```

バイナリ分類モデルと同様に、予測されるラベル/クラスは [predictedLabel] フィールドにあります。予測が各クラスとどれほど強く関連しているかは、predictedScores マップを見ることでさらに理解できます。このマップ内のクラスのスコアが大きいほど、予測はクラスと強く関連していて、最終的に最大値が predictedLabel として選択されます。

複数クラス予測の詳細については、「複数モデルクラスの洞察」を参照してください。

リアルタイムエンドポイントの削除

リアルタイム予測が完成した場合、リアルタイムエンドポイントを削除して追加料金の発生を抑えます。エンドポイントを削除するとすぐに料金がかからなくなります。

リアルタイムエンドポイントを削除するには

1. AWS Management Console にサインインし、Amazon Machine Learning コンソール (https://console.aws.amazon.com/machinelearning/) を開きます。
2. ナビゲーションバーの [Amazon Machine Learning] ドロップダウンで [ML モデル] を選択します。
3. リアルタイム予測が不要になったモデルを選択します。
4. [ML モデルレポート] ページの [予測] で、[概要] を選択します。
5. [リアルタイムエンドポイントの削除] を選択します。
6. [リアルタイムエンドポイントの削除] ダイアログボックスで、[削除] を選択します。

Amazon ML オブジェクトの管理

Amazon ML は、Amazon ML コンソールまたは Amazon ML API を通じて管理できる 4 つのオブジェクトを提供します。

- データソース
- ML モデル
- 評価
- バッチ予測

各オブジェクトは、機械学習アプリケーションを構築するライフサイクルにおいて異なる目的を果たし、各オブジェクトには、そのオブジェクトにのみ適用される特定の属性および機能があります。これらの違いにもかかわらず、同じような方法でオブジェクトを管理します。たとえば、オブジェクトのリスト、説明の取得、更新または削除にほぼ同じプロセスを使用します。

以下のセクションでは、4 つのオブジェクトすべてに共通の管理操作について説明し、相違点について説明します。

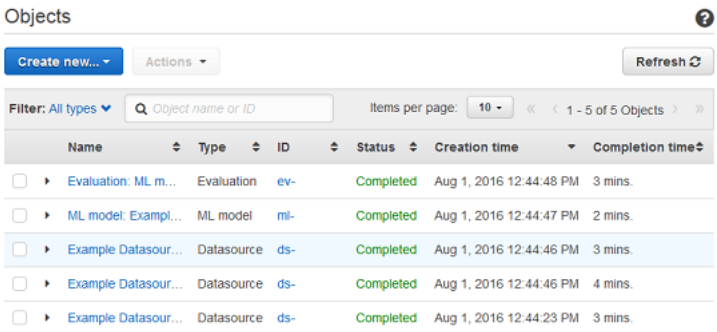
トピック

オブジェクトのリスト作成

Amazon Machine Learning (Amazon ML) データソース、ML モデル、評価、およびバッチ予測の詳細な情報について、一覧表示します。各オブジェクトについて、名前、タイプ、ID、ステータスコード、作成時刻が表示されます。また、特定のオブジェクトタイプに固有の詳細も表示されます。たとえば、データソースのデータ洞察も表示されます。

オブジェクトのリスト作成 (コンソール)

お客様が作成した最後の 1,000 個のオブジェクトのリストを表示するには、Amazon ML コンソールで [Objects] (オブジェクト) ダッシュボードを開きます。[Objects] (オブジェクト) ダッシュボードを表示するには、Amazon ML コンソールにログインします。



そのオブジェクトに固有の詳細を含むオブジェクトの詳細を表示するには、オブジェクトの名前または ID を選択します。たとえば、データソースの [データ洞察] を表示するには、データソース名を選択します。

[オブジェクト] ダッシュボードの列には、各オブジェクトに関する次の情報が表示されます。

名前

オブジェクトの名前。

タイプ

オブジェクトのタイプ。有効な値には、[データソース]、[ML モデル]、[評価]、および [バッチ予測] が含まれます。

> **Note**
>
> モデルがリアルタイム予測をサポートするように設定されたかどうかを調べるには、モデル名または ID を選択して [ML モデルの要約] ページに移動します。

ID

オブジェクトの ID。

ステータス

オブジェクトのステータス。値には、[保留]、[進行中]、[完了]、および [失敗] が含まれます。ステータスが [失敗] の場合は、データを確認してもう一度試してください。

作成時刻

Amazon ML がこのオブジェクトの作成を終了した日時。

完了時間

Amazon ML がこのオブジェクトを作成するのにかかった時間の長さ。モデルの完了時間を使用して、新しいモデルのトレーニング時間を見積もることができます。

データソース ID

モデルおよび評価などデータソースを使用して作成されたオブジェクトの場合、データソースの ID。データソースを削除すると、そのデータソースを使用して作成された ML モデルで予測の作成ができなくなります。

列ヘッダーの横にある 2 重三角形のアイコンを選択することで、いずれかの列でソートします。

オブジェクトのリスト作成 (API)

Amazon ML API では、次のオペレーションを使用してタイプ別にオブジェクトを一覧表示できます。

- `DescribeDataSources`
- `DescribeMLModels`
- `DescribeEvaluations`
- `DescribeBatchPredictions`

各オペレーションは、オブジェクトの長いリストを介して、フィルタリング、ソート、およびページ分割のパラメータを含みます。API でアクセスできるオブジェクト数に制限はありません。リストのサイズを制限するには [`Limit`] パラメータを使用し、その最大値は 100 です。

`Describe*` コマンドに対する API レスポンスには、必要に応じてページ分割トークン (`nextPageToken`) および各オブジェクトの簡単な説明が含まれています。オブジェクトの説明に

は、オブジェクトタイプに固有の詳細を含め、コンソールに表示される各オブジェクトタイプごとの同じ情報が含まれます。

> **Note**
>
> 応答が指定された制限よりも少ないオブジェクトしか含まない場合でも、さらに多くの結果があることを示す [nextPageToken] が含まれることがあります。応答が 0 個のアイテムの場合でも [nextPageToken] が含まれる可能性があります。

詳細については、「Amazon ML API リファレンス」を参照してください。

オブジェクトの説明の取得

コンソールまたは API により、オブジェクトの詳細な説明を表示できます。

コンソールでの詳細説明

コンソールで説明を表示するには、特定のタイプのオブジェクト (データソース、ML モデル、評価、バッチ予測) のリストに移動します。次に、リストを参照するか、名前または ID を検索することで、オブジェクトに対応するテーブルの行を見つけます。

API からの詳細説明

各オブジェクトタイプには、Amazon ML オブジェクトのすべての詳細を取得するための操作があります。

- GetDataSource
- GetMLModel
- GetEvaluation
- GetBatchPrediction

各オペレーションはちょうど 2 つのパラメータを使用します。オブジェクト ID と Verbose (詳細) と呼ばれるブーリアン型フラグです。true に設定された Verbose (詳細) での呼び出しには、オブジェクトについてのより詳細な情報が含まれ、レイテンシーおよびレスポンスのサイズが大きくなります。Verbose (詳細) フラグを設定することで含まれるフィールドの詳細については、Amazon ML API リファレンスを参照してください。

オブジェクトの更新

各オブジェクトタイプには、Amazon ML オブジェクトの詳細を更新する操作があります (「Amazon ML API リファレンス」を参照)。

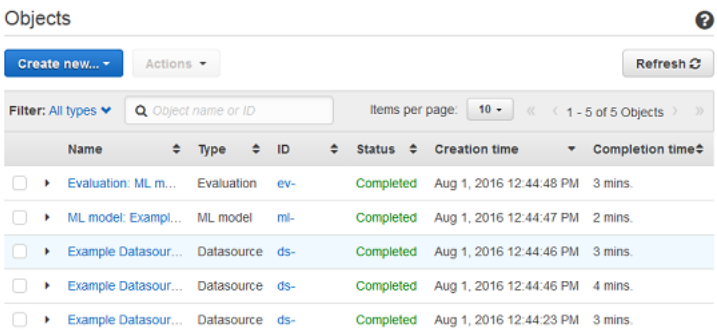
- UpdateDataSource
- UpdateMLModel
- UpdateEvaluation
- UpdateBatchPrediction

各オペレーションでは、どのオブジェクトが更新されているかを指定するためにオブジェクトの ID が必要です。すべてのオブジェクトの名前を更新できます。データソース、評価、およびバッチ予測のオブジェクトの他のプロパティは更新できません。ML モデルでは、ML モデルにリアルタイム予測エンドポイントが関連付けられていない限り、ScoreThreshold フィールドを更新できます。

オブジェクトの削除

ML モデル、データソース、評価、およびバッチ予測が必要なくなった場合、削除することができます。完了した後にバッチ予測以外の Amazon ML オブジェクトを保持しても追加料金はありませんが、オブジェクトを削除すると、管理しやすい整ったワークスペースが保持されます。1 つまたは複数のオブジェクトを削除するには、Amazon Machine Learning (Amazon ML) コンソールまたは API を使用します。

> **Warning**
>
> Amazon ML のオブジェクトを削除すると、効果は即時で、永続的で、元に戻せません。



オブジェクトの削除 (コンソール)

Amazon ML コンソールを使用して、モデルを含むオブジェクトを削除できます。モデルを削除するために使用する手順は、モデルを使ってリアルタイム予測を生成しているかどうかによって異なります。リアルタイム予測を生成するために使用されるモデルを削除するには、最初にリアルタイムエンドポイントを削除します。

Amazon ML のオブジェクトを削除するには (コンソール)

1. AWS Management Console にサインインし、Amazon Machine Learning コンソール (https://console.aws.amazon.com/machinelearning/) を開きます。
2. 削除する Amazon ML オブジェクトを選択します。複数のオブジェクトを選択するには、Shift キーを使用します。すべての選択オブジェクトの選択を解除するには、
または

ボタンを使用します。
3. [Actions] (アクション) として、[Delete] (削除) を選択します。
4. ダイアログボックスで、[削除] を選択してモデルを削除します。

リアルタイムエンドポイントがある Amazon ML モデルを削除するには (コンソール)

1. AWS Management Console にサインインし、Amazon Machine Learning コンソール (https://console.aws.amazon.com/machinelearning/) を開きます。
2. 削除するモデルを選択します。
3. [アクション] で [リアルタイムエンドポイントの削除] を選択します。
4. [削除] を選択して、エンドポイントを削除します。
5. モデルを再度選択します。
6. [Actions] (アクション) として、[Delete] (削除) を選択します。
7. [削除] を選択してモデルを削除します。

オブジェクトの削除 (API)

次の API コールを使用して、Amazon ML オブジェクトを削除することができます。

- `DeleteDataSource` - は `DataSourceId` パラメータを取ります。
- `DeleteMLModel` - は `MLModelId` パラメータを取ります。
- `DeleteEvaluation` - は `EvaluationId` パラメータを取ります。
- `DeleteBatchPrediction` - は `BatchPredictionId` パラメータを取ります。

詳細については、「Amazon 機械学習 API リファレンス」を参照してください。

Amazon ML と Amazon CloudWatch メトリックスのモニタリング

Amazon ML は、Amazon CloudWatch に自動的にメトリックスを送信し、ML モデルの使用統計を収集、分析できます。たとえば、バッチ予測およびリアルタイム予測の追跡に、RequestMode ディメンションに従って PredictCount メトリックスをモニタリングできます。メトリクスは自動的に収集され、5 分毎に Amazon CloudWatch に送られます。Amazon CloudWatch コンソール、AWS CLI、または AWS SDK を使用して、これらのメトリクスをモニタリングできます。

CloudWatch を経由して報告される Amazon ML メトリクスには料金はかかりません。メトリクスにアラームを設定している場合は、CloudWatch の標準料金が請求されます。

詳細については、Amazon CloudWatch 開発者ガイドの「Amazon CloudWatch の名前空間、ディメンション、メトリックスのリファレンス」の Amazon ML リストを参照してください。

AWS CloudTrail での Amazon ML API コールのログ記録

Amazon Machine Learning (Amazon ML) は、AWS CloudTrail と統合されています。これは、Amazon ML のユーザー、ロール、または AWS サービスで実行されたアクションを記録するためのサービスです。CloudTrail は、Amazon ML へのすべての API コールをイベントとしてキャプチャします。キャプチャされた呼び出しには、Amazon ML コンソールからの呼び出しと、Amazon ML API オペレーションへのコード呼び出しが含まれます。証跡を作成する場合は、Amazon ML のイベントなど、Amazon S3 バケットへの CloudTrail イベントの継続的な配信を有効にすることができます。追跡を設定しない場合でも、CloudTrail コンソールの [Event history] (イベント履歴) で最新のイベントを表示できます。CloudTrail で収集された情報を使用して、Amazon ML 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。

設定や有効化の方法など、CloudTrail の詳細については、「AWS CloudTrail ユーザーガイド」を参照してください。

CloudTrail 内の Amazon ML 情報

AWS アカウントを作成すると、そのアカウントに対して CloudTrail が有効になります。Amazon ML でサポートされているイベントアクティビティが発生すると、そのアクティビティは [Event history] (イベント履歴) の他の AWS のサービスのイベントとともに CloudTrail 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、「Viewing Events with CloudTrail Event History」(CloudTrail イベント履歴でのイベントの表示) を参照してください。

Amazon ML のイベントなど、AWS アカウント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。追跡により、CloudTrail はログファイルを Simple Storage Service (Amazon S3) バケットに配信できます。デフォルトでは、コンソールで作成した追跡がすべての AWS リージョンに適用されます。追跡は、AWS パーティションのすべてのリージョンからのイベントを口グに記録し、指定した Simple Storage Service (Amazon S3) バケットにログファイルを配信します。さらに、CloudTrail ログで収集したイベントデータをより詳細に分析し、それに基づく対応するためにその他の AWS の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。

- 追跡を作成するための概要
- CloudTrail のサポート対象サービスと統合
- Amazon SNS の CloudTrail の通知の設定
- 「複数のリージョンから CloudTrail ログファイルを受け取る」および「複数のアカウントから CloudTrail ログファイルを受け取る」

Amazon ML は、CloudTrail ログファイルのイベントとして以下のアクションのログ付けをサポートします。

- AddTags
- CreateBatchPrediction
- CreateDataSourceFromRDS
- CreateDataSourceFromRedshift
- CreateDataSourceFromS3
- CreateEvaluation
- CreateMLModel
- CreateRealtimeEndpoint
- DeleteBatchPrediction
- DeleteDataSource
- DeleteEvaluation
- DeleteMLModel
- DeleteRealtimeEndpoint
- DeleteTags
- DescribeTags
- UpdateBatchPrediction
- UpdateDataSource
- UpdateEvaluation
- UpdateMLModel

次の Amazon ML オペレーションでは、認証情報を含むリクエストパラメータが使用されます。これらのリクエストが CloudTrail に送信される前に、認証情報は 3 つのアスタリスク (「***」) に置き換えられます。

- CreateDataSourceFromRDS
- CreateDataSourceFromRedshift

次の Amazon ML オペレーション が Amazon ML コンソールで実行される場合、属性 `ComputeStatistics` は CloudTrail ログの `RequestParameters` コンポーネントに含まれません。

- CreateDataSourceFromRedshift
- CreateDataSourceFromS3

各イベントまたはログエントリには、リクエストの生成者に関する情報が含まれます。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、ルート認証情報と AWS Identity and Access Management (IAM) ユーザー認証情報のどちらを使用して送信されたか。
- リクエストがロールまたはフェデレーティッド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 のサービスによって送信されたかどうか。

詳細については、「CloudTrail userIdentity エレメント」を参照してください。

例: Amazon ML ログファイルのエントリ

「トレイル」は、指定した Simple Storage Service (Amazon S3) バケットにイベントをログファイルとして配信するように設定できます。CloudTrail のログファイルには、単一か複数のログエントリがあります。イベントはあらゆる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、リクエストのパラメータなどの情報が含まれます。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、アクションを示す CloudTrail ログエントリです。

```
{
    "Records": [
        {
            "eventVersion": "1.03",
            "userIdentity": {
                "type": "IAMUser",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::012345678910:user/Alice",
                "accountId": "012345678910",
                "accessKeyId": "EXAMPLE_KEY_ID",
                "userName": "Alice"
            },
            "eventTime": "2015-11-12T15:04:02Z",
```

```
            "eventSource": "machinelearning.amazonaws.com",
            "eventName": "CreateDataSourceFromS3",
            "awsRegion": "us-east-1",
            "sourceIPAddress": "127.0.0.1",
            "userAgent": "console.amazonaws.com",
            "requestParameters": {
                "data": {
                    "dataLocationS3": "s3://aml-sample-data/banking-batch.csv",
                    "dataSchema": "{\"version\":\"1.0\",\"rowId\":null,\"rowWeight
\":null,
                        \"targetAttributeName\":null,\"dataFormat\":\"CSV\",
                        \"dataFileContainsHeader\":false,\"attributes\":[
                          {\"attributeName\":\"age\",\"attributeType\":\"NUMERIC\"},
                          {\"attributeName\":\"job\",\"attributeType\":\"CATEGORICAL
\"},
                          {\"attributeName\":\"marital\",\"attributeType\":
\"CATEGORICAL\"},
                          {\"attributeName\":\"education\",\"attributeType\":
\"CATEGORICAL\"},
                          {\"attributeName\":\"default\",\"attributeType\":
\"CATEGORICAL\"},
                          {\"attributeName\":\"housing\",\"attributeType\":
\"CATEGORICAL\"},
                          {\"attributeName\":\"loan\",\"attributeType\":\"CATEGORICAL
\"},
                          {\"attributeName\":\"contact\",\"attributeType\":
\"CATEGORICAL\"},
                          {\"attributeName\":\"month\",\"attributeType\":\"CATEGORICAL
\"},
                          {\"attributeName\":\"day_of_week\",\"attributeType\":
\"CATEGORICAL\"},
                          {\"attributeName\":\"duration\",\"attributeType\":\"NUMERIC
\"},
                          {\"attributeName\":\"campaign\",\"attributeType\":\"NUMERIC
\"},
                          {\"attributeName\":\"pdays\",\"attributeType\":\"NUMERIC\"},
                          {\"attributeName\":\"previous\",\"attributeType\":\"NUMERIC
\"},
                          {\"attributeName\":\"poutcome\",\"attributeType\":
\"CATEGORICAL\"},
                          {\"attributeName\":\"emp_var_rate\",\"attributeType\":
\"NUMERIC\"},
                          {\"attributeName\":\"cons_price_idx\",\"attributeType\":
\"NUMERIC\"},
```

```
                            {\"attributeName\":\"cons_conf_idx\",\"attributeType\":
\"NUMERIC\"},
                            {\"attributeName\":\"euribor3m\",\"attributeType\":\"NUMERIC
\"},
                            {\"attributeName\":\"nr_employed\",\"attributeType\":
\"NUMERIC\"}
                          ],\"excludedAttributeNames\":[]}"
                },
                "dataSourceId": "exampleDataSourceId",
                "dataSourceName": "Banking sample for batch prediction"
            },
            "responseElements": {
                "dataSourceId": "exampleDataSourceId"
            },
            "requestID": "9b14bc94-894e-11e5-a84d-2d2deb28fdec",
            "eventID": "f1d47f93-c708-495b-bff1-cb935a6064b2",
            "eventType": "AwsApiCall",
            "recipientAccountId": "012345678910"
        },
        {
            "eventVersion": "1.03",
            "userIdentity": {
                "type": "IAMUser",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::012345678910:user/Alice",
                "accountId": "012345678910",
                "accessKeyId": "EXAMPLE_KEY_ID",
                "userName": "Alice"
            },
            "eventTime": "2015-11-11T15:24:05Z",
            "eventSource": "machinelearning.amazonaws.com",
            "eventName": "CreateBatchPrediction",
            "awsRegion": "us-east-1",
            "sourceIPAddress": "127.0.0.1",
            "userAgent": "console.amazonaws.com",
            "requestParameters": {
                "batchPredictionName": "Batch prediction: ML model: Banking sample",
                "batchPredictionId": "exampleBatchPredictionId",
                "batchPredictionDataSourceId": "exampleDataSourceId",
                "outputUri": "s3://EXAMPLE_BUCKET/BatchPredictionOutput/",
                "mLModelId": "exampleModelId"
            },
            "responseElements": {
                "batchPredictionId": "exampleBatchPredictionId"
```

```
            },
            "requestID": "3e18f252-8888-11e5-b6ca-c9da3c0f3955",
            "eventID": "db27a771-7a2e-4e9d-bfa0-59deee9d936d",
            "eventType": "AwsApiCall",
            "recipientAccountId": "012345678910"
        }
    ]
}
```

Amazon ML オブジェクトのタグ付け

Amazon Machine Learning (Amazon ML) オブジェクトに、タグ付きのメタデータを割り当てて、それらを整理して管理します。タグは、オブジェクトに対して定義するキーと値のペアです。

タグを使用して Amazon ML オブジェクトを整理して管理するだけでなく、それらを使用して AWS コストを分類して追跡することもできます。AWS オブジェクト (ML モデルなど) にタグを適用すると、AWS のコスト配分レポートに、タグ別に集計された使用状況とコストが表示されます。自社のカテゴリ (たとえばコストセンター、アプリケーション名、所有者) を表すタグを適用すると、複数のサービスにわたってコストを分類することができます。詳細については、AWS Billing ユーザーガイドのコスト配分タグを使用したカスタム請求レポートを参照してください。

目次

タグの基本

タグを使用してオブジェクトを分類すると、オブジェクトを簡単に管理できます。たとえば、目的、所有者、環境などに基づいてオブジェクトを分類できます。次に、所有者と、関連するアプリケーションに基づいてモデルを追跡するのに役立つタグのセットを定義できます。次にいくつかの例を示します。

- プロジェクト: プロジェクト名
- 所有者: 名前
- 目的 : マーケティング予測
- アプリケーション: アプリケーション名
- 環境: 本稼働

Amazon ML コンソール、または API を使用して、以下のタスクを実行します。

- オブジェクトにタグを追加します

- オブジェクトのタグを表示します
- オブジェクトのタグを編集します
- オブジェクトからタグを削除します

デフォルトでは、Amazon ML オブジェクトに適用されたタグは、そのオブジェクトを使用して作成されたオブジェクトにコピーされます。例えば、Amazon Simple Storage Service (Amazon S3) データソースに「マーケティングコスト: ターゲットを絞ったマーケティングキャンペーン」タグがある場合、そのデータソースを使用して作成されたモデルには、モデルの評価として、「マーケティングコスト: ターゲットを絞ったマーケティングキャンペーン」タグが付きます。これにより、タグを使用して、マーケティングキャンペーンに使用されるすべてのオブジェクトなどの関連オブジェクトを追跡できます。「マーケティングコスト : ターゲットを絞ったマーケティングキャンペーン」というタグが付いたモデルや「マーケティングコスト : ターゲットを絞ったマーケティングの顧客」というタグのあるデータソースなど、タグソース間に競合がある場合、Amazon ML はモデルからタグを適用します。

タグの制限

タグには次の制限があります。

基本制限:

- オブジェクトあたりのタグの最大数は 50 です。
- タグのキーと値は大文字と小文字が区別されます。
- 削除されたオブジェクトのタグを変更または編集することはできません。

タグキーの制限:

- 各タグキーは一意である必要があります。既に使用されているキーを含むタグを追加すると、新しいタグで、そのオブジェクトの既存のキーと値のペアが上書きされます。
- `aws:` は AWS が使用するように予約されているため、このプレフィックスを含むタグキーで開始することはできません。AWS ではユーザーの代わりにこのプレフィックスで始まるタグを作成しますが、ユーザーはこれらのタグを編集または削除することはできません。
- タグキーの長さは 1 ~ 128 文字 (Unicode) にする必要があります。
- タグキーは、次の文字で構成する必要があります。Unicode 文字、数字、空白、特殊文字 (`_ . / = + - @`)。

タグ値の制限:

- タグ値の長さは 0 ~ 255 文字 (Unicode) にする必要があります。
- タグ値は空白にすることができます。空白にしない場合は、次の文字で構成する必要があります。Unicode 文字、数字、空白、特殊文字 (_ . / = + - @)。

Amazon ML オブジェクトのタグ付け (コンソール)

Amazon ML コンソールを使用してタグを表示、追加、編集、および削除できます。

オブジェクトのタグを表示するには (コンソール)

1. AWS Management Console にサインインし、Amazon Machine Learning コンソール (https://console.aws.amazon.com/machinelearning/) を開きます。
2. ナビゲーションバーで、リージョンセレクターを展開し、リージョンを選択します。
3. [オブジェクト] ページで、オブジェクトを選択します。
4. 選択したオブジェクトの [タグ] セクションにスクロールします。そのオブジェクトのタグは、セクションの下部に一覧表示されます。

オブジェクトにタグを追加するには (コンソール)

1. AWS Management Console にサインインし、Amazon Machine Learning コンソール (https://console.aws.amazon.com/machinelearning/) を開きます。
2. ナビゲーションバーで、リージョンセレクターを展開し、リージョンを選択します。
3. [オブジェクト] ページで、オブジェクトを選択します。
4. 選択したオブジェクトの [タグ] セクションにスクロールします。そのオブジェクトのタグは、セクションの下部に一覧表示されます。
5. [タグの追加または編集] を選択します。
6. [タグの追加] で [キー] フィールドにタグキーを指定して、オプションで [値] フィールドにタグ値を指定し、[変更の適用] を選択します。

   [変更の適用] ボタンが有効でない場合は、指定したタグキーまたはタグ値のいずれかがタグの制限を満たしていません。詳細については、「タグの制限」を参照してください。
7. [タグ] セクションのリストに新しいタグを表示するには、ページを更新します。

タグを編集するには (コンソール)

1. AWS Management Console にサインインし、Amazon Machine Learning コンソール (https://console.aws.amazon.com/machinelearning/) を開きます。
2. ナビゲーションバーで、リージョンセレクターを展開し、リージョンを選択します。
3. [オブジェクト] ページで、オブジェクトを選択します。
4. 選択したオブジェクトの [タグ] セクションにスクロールします。そのオブジェクトのタグは、セクションの下部に一覧表示されます。
5. [タグの追加または編集] を選択します。
6. [適用したタグ] で、[値] フィールドでタグ値を編集し、[変更の適用] を選択します。

   [適用の変更] ボタンが有効でない場合は、指定したタグ値がタグの制限を満たしていません。詳細については、「タグの制限」を参照してください。
7. [タグ] セクションのリストに更新されたタグを表示するには、ページを更新します。

オブジェクトからタグを削除するには (コンソール)

1. AWS Management Console にサインインし、Amazon Machine Learning コンソール (https://console.aws.amazon.com/machinelearning/) を開きます。
2. ナビゲーションバーで、リージョンセレクターを展開し、リージョンを選択します。
3. [オブジェクト] ページで、オブジェクトを選択します。
4. 選択したオブジェクトの [タグ] セクションにスクロールします。そのオブジェクトのタグは、セクションの下部に一覧表示されます。
5. [タグの追加または編集] を選択します。
6. [適用した変更] で、削除するタグを選択し、[変更の適用] を選択します。

Amazon ML オブジェクトのタグ付け (API)

Amazon ML API を使用してタグの追加、一覧表示、および削除を行うことができます。例については、次のドキュメントを参照してください。

AddTags

指定したオブジェクトのタグを追加または編集します。

DescribeTags

指定したオブジェクトのタグを一覧表示します。

DeleteTags

指定されたオブジェクトからタグを削除します。

Amazon Machine Learning のリファレンス

トピック

Amazon S3 からデータを読み込むための Amazon ML アクセス許可の取得

Amazon S3 で入力データからデータソースオブジェクトを作成するには、入力データが格納されている S3 の場所に対して、Amazon ML に以下のアクセス許可を与える必要があります。

- S3 バケットおよびプレフィックスの GetObject アクセス許可。
- S3 バケットの ListBucket アクセス許可。他のアクションと異なり、ListBucket はバケット全体のアクセス許可 (プレフィックスにではなく) が付与されている必要があります。ただし、Condition 句を使用してアクセス権限を特定のプレフィックスにスコープできます。

Amazon ML コンソールを使用してデータソースを作成する場合、これらのアクセス許可をバケットに追加することができます。ウィザードの手順を完了したときにそれらを追加するかどうかを確認するプロンプトが表示されます。次のポリシーの例は、サンプルの場所 s3://*examplebucket*/*exampleprefix* からデータを読み取る Amazon ML のアクセス権限を付与する方法を示しています。ListBucket アクセス許可の適用範囲は *exampleprefix* 入力パスにのみ絞り込みます。

```
{
```

```
    "Version": "2008-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": { "Service": "machinelearning.amazonaws.com" },
        "Action": "s3:GetObject",
        "Resource": "arn:aws:s3:::examplebucket/exampleprefix/*"
        "Condition": {
            "StringEquals": { "aws:SourceAccount": "123456789012" }
            "ArnLike": { "aws:SourceArn": "arn:aws:machinelearning:us-
east-1:123456789012:*" }
        }
    },
    {
        "Effect": "Allow",
        "Principal": {"Service": "machinelearning.amazonaws.com"},
        "Action": "s3:ListBucket",
        "Resource": "arn:aws:s3:::examplebucket",
        "Condition": {
            "StringLike": { "s3:prefix": "exampleprefix/*" }
            "StringEquals": { "aws:SourceAccount": "123456789012" }
            "ArnLike": { "aws:SourceArn": "arn:aws:machinelearning:us-
east-1:123456789012:*" }
        }
    }]
}
```

このポリシーをデータに適用するには、データが保存されている S3 バケットに関連付けられたポリシーステートメントを編集する必要があります。

S3 バケットのアクセス権限ポリシーを編集するには (古いコンソールを使用)

1. AWS Management Console にサインインし、Amazon S3 コンソール https://console.aws.amazon.com/s3/ を開きます。
2. データが置かれているバケット名を選択します。
3. [Properties] (プロパティ) を選択します。
4. [バケットポリシーの編集] を選択します。
5. 上記のポリシーを入力し、ニーズに合わせてカスタマイズしてから、[保存] を選択します。
6. [Save (保存)] を選択します。

S3 バケットのアクセス権限ポリシーを編集するには (新しいコンソールを使用)

1. AWS Management Console にサインインし、Amazon S3 コンソール https://console.aws.amazon.com/s3/ を開きます。
2. バケット名、[アクセス権限] の順に選択します。
3. [バケットポリシー] を選択します。
4. 上記のポリシーを入力し、ニーズに合わせてカスタマイズします。
5. [Save (保存)] を選択します。

Amazon S3 に予測を出力するために Amazon ML のアクセス許可を得る

バッチ予測オペレーションの結果を Amazon S3 に出力するには、バッチ予測生成オペレーションの入力として提供された出力場所に対する以下のアクセス権限を Amazon ML に付与する必要があります。

- S3 バケットおよびプレフィックスの GetObject アクセス許可。
- S3 バケットおよびプレフィックスの PutObject アクセス許可。
- S3 バケットおよびプレフィックスの PutObjectAcl アクセス許可。
  - オブジェクトが作成された後で既定の ACL bucket-owner-full-control アクセス許可をユーザーの AWS アカウントに与えるために、Amazon ML にはこれらのアクセス許可が必要です。
- S3 バケットの ListBucket アクセス許可。他のアクションと異なり、ListBucket はバケット全体のアクセス許可 (プレフィックスにではなく) が付与されている必要があります。ただし、条件句を使用してアクセス権限を特定のプレフィックスにスコープできます。

Amazon ML コンソールを使用してバッチ予測リクエストを作成する場合、これらのアクセス許可をバケットに追加することができます。ウィザードの手順を完了する時に、それらを追加するかどうかを確認するプロンプトが表示されます。

以下のポリシーの例は、サンプルの場所 s3://examplebucket/exampleprefix にデータを書き込むアクセス許可を Amazon ML に付与する方法を示しています。ListBucket アクセス許可の適用範囲は exampleprefix 入力パスにのみに絞り込まれ、出力プレフィックスの put object ACL のアクセス許可を Amazon ML に付与しています。

```
{
    "Version": "2008-10-17",
```

```
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": { "Service": "machinelearning.amazonaws.com"},
        "Action": [
            "s3:GetObject",
            "s3:PutObject"
       ],
        "Resource": "arn:aws:s3:::examplebucket/exampleprefix/*"
        "Condition": {
            "StringEquals": { "aws:SourceAccount": "123456789012" }
            "ArnLike": { "aws:SourceArn": "arn:aws:machinelearning:us-
east-1:123456789012:*" }
        }
    },
    {
        "Effect": "Allow",
        "Principal": { "Service": "machinelearning.amazonaws.com"},
        "Action": "s3:PutObjectAcl",
        "Resource": "arn:aws:s3:::examplebucket/exampleprefix/*",
        "Condition": {
            "StringEquals": { "s3:x-amz-acl":"bucket-owner-full-control" }
            "StringEquals": { "aws:SourceAccount": "123456789012" }
           "ArnLike": { "aws:SourceArn": "arn:aws:machinelearning:us-
east-1:123456789012:*" }
       }
    },
    {
       "Effect": "Allow",
        "Principal": {"Service": "machinelearning.amazonaws.com"},
       "Action": "s3:ListBucket",
        "Resource": "arn:aws:s3:::examplebucket",
        "Condition": {
            "StringLike": { "s3:prefix": "exampleprefix/*" }
            "StringEquals": { "aws:SourceAccount": "123456789012" }
            "ArnLike": { "aws:SourceArn": "arn:aws:machinelearning:us-
east-1:123456789012:*" }
        }
    }]
}
```

このポリシーをデータに適用するには、データが保存されている S3 バケットに関連付けられたポリシーステートメントを編集する必要があります。

S3 バケットのアクセス権限ポリシーを編集するには (古いコンソールを使用)

1. AWS Management Console にサインインし、Amazon S3 コンソール https://console.aws.amazon.com/s3/ を開きます。
2. データが置かれているバケット名を選択します。
3. [Properties] (プロパティ) を選択します。
4. [バケットポリシーの編集] を選択します。
5. 上記のポリシーを入力し、ニーズに合わせてカスタマイズしてから、[保存] を選択します。
6. [Save (保存)] を選択します。

S3 バケットのアクセス権限ポリシーを編集するには (新しいコンソールを使用)

1. AWS Management Console にサインインし、Amazon S3 コンソール https://console.aws.amazon.com/s3/ を開きます。
2. バケット名、[アクセス権限] の順に選択します。
3. [バケットポリシー] を選択します。
4. 上記のポリシーを入力し、ニーズに合わせてカスタマイズします。
5. [Save (保存)] を選択します。

IAM による Amazon ML リソースへのアクセスの制御

AWS Identity and Access Management (IAM) を利用すると、AWS のサービスおよびリソースに対するお客様のユーザーのアクセスを安全にコントロールすることができます。IAM を使用すると、AWS ユーザー、グループ、およびロールを作成および管理し、アクセス権を使用して AWS リソースへのアクセスを許可および拒否できます。IAM を Amazon Machine Learning (Amazon ML) で使用すると、組織内のユーザーが特定の AWS リソースを使用できるかどうか、および特定の Amazon ML API アクションを使用してタスクを実行できるかどうかを制御できます。

IAM を使用して、以下を行えます。

- お客様の AWS アカウントでユーザーとグループを作成する。
- お客様の AWS アカウントでユーザーごとに固有のセキュリティ認証情報を割り当てる
- AWS のリソースを使用してタスクを実行するために各ユーザーのアクセス権限を制御する
- お客様の AWS アカウントのリソースを AWS アカウント内のユーザー間で共有する

- AWS アカウントにロールを作成し、そのアクセス権限を管理して、それを行えるユーザーまたはサービスを定義する
- IAM でロールを作成し、権限を管理することで、そのロールを適用するエンティティまたは AWS のサービスによって実行可能なオペレーションをコントロールする。ロールをどのエンティティに適用するかについても定義できます。

組織が既に IAM アイデンティティを持っている場合、AWS のリソースを使用してタスクを実行するためのアクセス権限を、それを使用して付与することができます。

IAM の詳細については、IAM ユーザーガイドを参照してください。

IAM ポリシー構文

IAM ポリシーは 1つ以上のステートメントで構成されるJSONドキュメントです。各ステートメントは次のような構成です。

```
{
    "Statement":[{
        "Effect":"effect",
        "Action":"action",
        "Resource":"arn",
        "Condition":{
            "condition operator":{
                "key":"value"
            }
        }
    }]
}
```

ポリシーステートメントには以下の要素が含まれます。

- Effect: ステートメントの後半で指定するリソースおよび API アクションを使用するためのアクセス許可を制御します。有効な値は、`Allow` および `Deny` です。デフォルトでは、IAM ユーザーはリソースおよび API アクションを使用するアクセス許可がないため、リクエストはすべて拒否されます。明示的な `Allow` はデフォルトに優先します。明示的な `Deny` は、すべての `Allows` に優先します。
- Action: アクセス許可を付与または拒否する対象となる、特定の API アクションです。
- [Resource] (リソース): アクションによって影響を及ぼされるリソースです。ステートメントでリソースを指定するには、Amazon リソースネーム (ARN) を使用します。

- Condition (オプション): ポリシーが有効になるタイミングを制御します。

IAM ポリシーの作成および管理を簡素化するために、AWS Policy Generator と IAM Policy Simulator を使用できます。

Amazon MLAmazon ML の IAM ポリシーアクションの指定

IAM ポリシーステートメントで、IAM をサポートするすべてのサービスの任意の API アクションを指定できます。Amazon ML API アクションのポリシーステートメントを作成する場合、次の例に示すように、API アクションの名前の前に `machinelearning:` を追加します。

- `machinelearning:CreateDataSourceFromS3`
- `machinelearning:DescribeDataSources`
- `machinelearning:DeleteDataSource`
- `machinelearning:GetDataSource`

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、アクション間をコンマで区切ります。

```
"Action": ["machinelearning:action1", "machinelearning:action2"]
```

ワイルドカードを使用して複数のアクションを指定することもできます。たとえば、名前が「Get」という単語で始まるすべてのアクションを指定できます。

```
"Action": "machinelearning:Get*"
```

Amazon ML アクションをすべて指定するには、* ワイルドカードを使用します。

```
"Action": "machinelearning:*"
```

すべての Amazon ML API アクションの一覧については、「Amazon Machine Learning API リファレンス」を参照してください。

IAM ポリシーで Amazon ML リソースの ARN を指定する

IAM ポリシーステートメントは 1 つまたは複数のリソースに適用されます。ARN でポリシーのリソースを指定します。

Amazon ML リソースの ARN を指定するには、次の形式を使用します。

"Resource": `arn:aws:machinelearning:region:account:resource-type/identifier`

次の例は、共通 ARN を指定する方法を示しています。

データソース ID: `my-s3-datasource-id`

```
"Resource":
arn:aws:machinelearning:<region>:<your-account-id>:datasource/my-s3-datasource-id
```

ML モデル ID: `my-ml-model-id`

```
"Resource":
arn:aws:machinelearning:<region>:<your-account-id>:mlmodel/my-ml-model-id
```

バッチ予測 ID: `my-batchprediction-id`

```
"Resource":
arn:aws:machinelearning:<region>:<your-account-id>:batchprediction/my-batchprediction-
id
```

評価 ID: `my-evaluation-id`

```
"Resource": arn:aws:machinelearning:<region>:<your-account-id>:evaluation/my-
evaluation-id
```

Amazon ML のポリシーの例

例 1: 機械学習リソースメタデータの読み取りをユーザーに許可する

次のポリシーでは、ユーザーまたはグループが、DescribeDataSources、DescribeMLModels、DescribeBatchPredictions、DescribeEvaluations、GetDat および GetEvaluation アクションを特定のリソースで実行して、データソースのメタデータ、ML モデル、バッチ予測、および評価を読み取ることを許可します。Describe * オペレーションのアクセス権限を特定のリソースに制限することはできません。

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "machinelearning:Get*"
        ],
        "Resource": [
            "arn:aws:machinelearning:<region>:<your-account-id>:datasource/S3-DS-ID1",
            "arn:aws:machinelearning:<region>:<your-account-id>:datasource/REDSHIFT-DS-
ID1",
            "arn:aws:machinelearning:<region>:<your-account-id>:mlmodel/ML-MODEL-ID1",
            "arn:aws:machinelearning:<region>:<your-account-id>:batchprediction/BP-
ID1",
            "arn:aws:machinelearning:<region>:<your-account-id>:evaluation/EV-ID1"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "machinelearning:Describe*"
        ],
        "Resource": [
            "*"
        ]
    }]
}
```

例 2: 機械学習リソースの作成をユーザーに許可する

次のポリシーでは、ユーザーまたはグループが CreateDataSourceFromS3、CreateDataSourceFromRedshift、CreateDataSourceFromRDS、Cr および CreateEvaluation アクションを実行して、機械学習データソース、ML モデル、バッチ予測および評価を作成することを許可します。これらのアクションのアクセス権限を特定のリソースに制限することはできません。

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "machinelearning:CreateDataSourceFrom*",
            "machinelearning:CreateMLModel",
            "machinelearning:CreateBatchPrediction",
            "machinelearning:CreateEvaluation"
```

```
        ],
        "Resource": [
            "*"
        ]
    }]
}
```

例 3: リアルタイムエンドポイントの作成と削除、および ML モデルによるリアルタイム予測の実行をユーザーに許可する

次のポリシーでは、ユーザーまたはグループが CreateRealtimeEndpoint、DeleteRealtimeEndpoint および Predict アクションを特定の ML モデルで実行して、リアルタイムエンドポイントの作成や削除、およびリアルタイム予測を実行することを許可します。

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "machinelearning:CreateRealtimeEndpoint",
            "machinelearning:DeleteRealtimeEndpoint",
            "machinelearning:Predict"
        ],
        "Resource": [
            "arn:aws:machinelearning:<region>:<your-account-id>:mlmodel/ML-MODEL"
        ]
    }]
}
```

例 4: 特定のリソースの更新と削除をユーザーに許可する

次のポリシーでは、ユーザーまたはグループが UpdateDataSource、UpdateMLModel、UpdateBatchPrediction、UpdateEvaluation、DeleteDa および DeleteEvaluation アクションをアカウントのリソースで実行して、AWS アカウントの特定のリソースの更新や削除を実行することを許可します。

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

```
            "machinelearning:Update*",
            "machinelearning:DeleteDataSource",
            "machinelearning:DeleteMLModel",
            "machinelearning:DeleteBatchPrediction",
            "machinelearning:DeleteEvaluation"
        ],
        "Resource": [
            "arn:aws:machinelearning:<region>:<your-account-id>:datasource/S3-DS-ID1",
            "arn:aws:machinelearning:<region>:<your-account-id>:datasource/REDSHIFT-DS-
ID1",
            "arn:aws:machinelearning:<region>:<your-account-id>:mlmodel/ML-MODEL-ID1",
            "arn:aws:machinelearning:<region>:<your-account-id>:batchprediction/BP-
ID1",
            "arn:aws:machinelearning:<region>:<your-account-id>:evaluation/EV-ID1"
        ]
    }]
}
```

例 5: すべての Amazon ML アクションを許可

次のポリシーでは、任意の Amazon ML アクションの使用をユーザーまたはグループに許可します。このポリシーはすべての機械学習リソースへのフルアクセスを許可するため、管理者にのみ適用します。

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

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、アクションを実行する許可を持たないエンティティが、より高い特権を持つエンティティにそのアクションの実行を強制できるというセキュリティ問題です。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ことがあります。サービス間でのなりすま

しは、1 つのサービス (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は、本来ならアクセスすることが許可されるべきではない方法でその許可を使用して、別の顧客のリソースに対する処理を実行するように操作される場合があります。これを防ぐために AWS では、顧客のすべてのサービスのデータを保護するのに役立つツールを提供しています。これには、アカウントのリソースへのアクセス許可が付与されたサービスプリンシパルを使用します。

リソースポリシー内では aws:SourceArn および aws:SourceAccount グローバル条件コンテキストキーを使用して、Amazon Machine Learning が別のサービスに付与する、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。aws:SourceArn の値に Amazon S3 バケット ARN などのアカウント ID が含まれていない場合は、両方のグローバル条件コンテキストキーを使用して、アクセス許可を制限する必要があります。同じポリシーステートメントでこれらのグローバル条件コンテキストキーの両方を使用し、アカウント ID にaws:SourceArn の値が含まれていない場合、aws:SourceAccount 値と aws:SourceArn 値の中のアカウントには、同じアカウント ID を使用する必要があります。クロスサービスのアクセスにリソースを 1 つだけ関連付けたい場合は、aws:SourceArn を使用します。クロスサービスが使用できるように、アカウント内の任意のリソースを関連づけたい場合は、aws:SourceAccount を使用します。

混乱した代理問題から保護するための最も効果的な方法は、リソースの完全な ARN を指定しながら、aws:SourceArn グローバル条件コンテキストキーを使用することです。リソースの完全な ARN が不明な場合や、複数のリソースを指定する場合には、グローバルコンテキスト条件キー aws:SourceArn で、ARN の未知部分を示すためにワイルドカード (*) を使用します。例えば、arn:aws:*servicename*:*:*123456789012*:* です。

次の例は、Amazon ML で aws:SourceArn および aws:SourceAccount グローバル条件コンテキストキーを使用して、Amazon S3 バケットからデータを読み取る際に混乱した代理問題を防ぐ方法を示しています。

```
{
    "Version": "2008-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": { "Service": "machinelearning.amazonaws.com" },
        "Action": "s3:GetObject",
        "Resource": "arn:aws:s3:::examplebucket/exampleprefix/*"
        "Condition": {
            "StringEquals": { "aws:SourceAccount": "123456789012" }
            "ArnLike": { "aws:SourceArn": "arn:aws:machinelearning:us-
east-1:123456789012:*" }
```

```
        }
    },
    {
        "Effect": "Allow",
        "Principal": {"Service": "machinelearning.amazonaws.com"},
        "Action": "s3:ListBucket",
        "Resource": "arn:aws:s3:::examplebucket",
        "Condition": {
            "StringLike": { "s3:prefix": "exampleprefix/*" }
            "StringEquals": { "aws:SourceAccount": "123456789012" }
            "ArnLike": { "aws:SourceArn": "arn:aws:machinelearning:us-
east-1:123456789012:*" }
        }
    }]
}
```

非同期オペレーションの依存関係管理

Amazon ML のバッチオペレーションは、正常に完了するために他の処理に依存しています。これらの依存関係を管理するために、Amazon ML は依存性のあるリクエストを識別し、操作が完了したことを確認します。操作が完了していない場合、Amazon ML は、依存する操作が完了するまで、最初のリクエストを保存します。

バッチオペレーション間にはいくつかの依存関係があります。たとえば、ML モデルを作成する前に、ML モデルをトレーニングできるデータソースを作成しておく必要があります。Amazon ML は、利用可能なデータソースがない場合、ML モデルをトレーニングすることはできません。

ただし、Amazon ML は非同期操作の依存関係管理をサポートしています。たとえば、データモデル上で ML モデルをトレーニングするリクエストを送信する前に、データ統計が計算されるまで待つ必要はありません。代わりに、データソースが作成されるとすぐに、データソースを使用して ML モデルをトレーニングするリクエストを送信できます。Amazon ML は、データソース統計が計算されるまで、実際にトレーニング操作を開始しません。createMLModel リクエストは、統計が計算されるまでキューに入れられます。それが完了すると、Amazon ML は直ちに createMLModel 操作の実行を試みます。同様に、トレーニングを終了していない ML モデルのバッチ予測と評価リクエストを送信できます。

次の表は、異なる AmazonML のアクションを進めるための要件を示しています。

以下をするために	必要なもの
ML モデルを作成する (createMLModel)	計算されたデータ統計を持つデータソース
バッチ予測を作成する (createBatchPrediction)	データソース ML モデル
バッチ評価を作成する (createBatchEvaluation)	データソース ML モデル

リクエストステータスの確認

リクエストを送信する場合、Amazon Machine Learning (Amazon ML) API を使用してステータスを確認できます。例えば、`createMLModel` リクエストを送信する場合、`describeMLModel` 呼び出しを使用してステータスを確認できます。Amazon ML は、次のいずれかのステータスで応答します。

ステータス	定義
保留中	Amazon ML は、リクエストを検証しています。 または Amazon ML は、リクエストを実行する前に演算リソースが利用可能になるのを待っています。これは、アカウントで同時に実行するバッチ操作リクエストの最大数を超えた場合に発生する可能性があります。この場合、他の実行中のリクエストが完了するかキャンセルされたときに、ステータスが InProgress に移行します。 または Amazon ML はリクエストを完了するために必要なバッチ操作を待機しています。
進行中	リクエストがまだ実行中です。

ステータス	定義
COMPLETED	リクエストは完了し、オブジェクトの使用準備 (ML モデルとデータソース) または表示準備 (バッチ予測と評価) が整いました。
FAILED	提供されたデータに何か問題があるか、または、オペレーションがキャンセルされました。たとえば、完了に失敗したデータソースのデータ統計を計算しようとすると、無効または失敗というステータスメッセージが表示されることがあります。エラーメッセージは、操作が正常に完了しなかった理由を説明します。
DELETED	オブジェクトは削除済みです。

Amazon ML は、Amazon ML がオブジェクトの作成を完了したときなど、オブジェクトに関する情報も提供します。詳細については、「オブジェクトのリスト作成」を参照してください。

システムの制限

堅牢で信頼性の高いサービスを提供するために、Amazon ML はシステムに対するリクエストに一定の制限を課しています。ほとんどの ML の問題は、これらの制約内に簡単に収まります。ただし、Amazon ML の使用がこれらの制限によって制約を受けることが判明した場合は、AWS カスタマーサービスに連絡して、制限を上げるようリクエストできます。たとえば、同時に実行できるジョブの数が 5 つに制限されているとします。この制限のために、ジョブがリソース待ちで頻繁にキューにあること分かった場合は、アカウントでこの上限を引き上げるのが適切かもしれません。

次の表は、Amazon ML におけるデフォルトのアカウントごとの制限を示しています。これらの制限のすべてが AWS カスタマーサービスによって引き上げられるわけではありません。

[制限のタイプ]	[システム制限]
各観測値のサイズ	100 KB
トレーニングデータのサイズ *	100 GB
バッチ予測入力のサイズ	1 TB
バッチ予測入力のサイズ (レコード数)	1 億件

[制限のタイプ]	[システム制限]
データファイル (スキーマ) 内の変数数	1,000
レシピの複雑さ (処理された出力変数数)	10,000
各リアルタイムの予測エンドポイントの TPS	200
すべてのリアルタイムの予測エンドポイントについての合計 TPS	10,000
すべてのリアルタイムの予測エンドポイントについての合計 RAM	10 GB
同時に実行されるジョブの数	25
任意のジョブの最長の実行時間	7 日間
複数クラスの ML モデルのクラス数	100
ML モデルサイズ	最小 1 MB、最大 2 GB
オブジェクトあたりのタグの数	50

- データファイルのサイズは、ジョブが迅速に終了するように制限されています。7 日間以上実行中であったジョブが自動的に終了すると、ステータスは "失敗" になります。

すべてのオブジェクトの名前と ID

Amazon ML のすべてのオブジェクトは識別子または ID を持っている必要があります。Amazon ML コンソールの場合は ID 値が生成されますが、API を使用する場合は独自の ID 値を生成する必要があります。各 ID は、AWS アカウントの同じタイプのすべての Amazon ML オブジェクトで一意である必要があります。つまり、同じ ID で 2 つの評価を持つことはできません。同じ ID を使用して評価とデータソースを持つことは可能ですが、推奨されません。

オブジェクトには、ランダムに生成された識別子とタイプを識別する短い文字列のプレフィックスを使用することをお勧めします。たとえば、Amazon ML コンソールがデータソースを生成する場合、「ds-zScWIuWiOxF」のようなランダムで一意の ID をデータソースに割り当てます。この ID は 1 人のユーザーに関して衝突回避のために十分にランダムで、かつコンパクトで読みやすいものです。利便性とわかりやすさのために「ds-」というプレフィックスを付けていますが、必須ではありませ

ん。ID 文字列に何を使用したらよいか明確でない場合は、16 進数 UUID 値 (28b1e915-57e5-4e6c-a7bd-6fb4e729cb23 など) を使用することをお勧めします。これはどのような最新のプログラミング環境でも使用可能です。

ID 文字列には ASCII 文字、数字、ハイフン、アンダースコアを含めることができ、最大 64 文字までです。メタデータを ID 文字列にエンコードすることが可能で、これはおそらく便利な方法に見えます。しかし、オブジェクトが作成された後はその ID が変更できないため、この方法は推奨されません。

オブジェクト名は、ユーザーフレンドリなメタデータを各オブジェクトに関連付ける簡単な方法です。オブジェクトを作成した後、その名前を更新できます。これにより、オブジェクト名に ML ワークフローの一部を反映させることができます。たとえば、最初は ML モデル名を「実験 #3」とし、後でモデル名を「最終本番モデル」と変更できます。名前は、最大 1,024 文字までの任意の文字列です。

オブジェクトの存続期間

Amazon ML で作成するデータソース、ML モデル、評価、またはバッチ予測オブジェクトは、作成後少なくとも 2 年間は使用可能になります。Amazon ML は、2 年以上アクセスまたは使用されていないオブジェクトを自動的に削除することがあります。

リソース

このサービスを利用する際に役立つ関連リソースは次のとおりです。

- Amazon ML 製品情報 – Amazon ML に関するすべての関連製品情報を一元的にキャプチャします。
- Amazon ML のよくある質問 – この製品について、開発者からよく寄せられる質問について説明します。
- Amazon ML サンプルコード – Amazon ML を使用するサンプルアプリケーションです。サンプルコードを開始点として使用して、独自の ML アプリケーションを作成することができます。
- Amazon ML API リファレンス – Amazon ML のすべての API 操作を詳しく説明します。また、サポートされるウェブサービスプロトコルのサンプルのリクエストとレスポンスも提供します。
- AWS デベロッパーリソースセンター – 関連ドキュメント、コードサンプル、リリースノートなどの情報に一か所でアクセスできるため、AWS での革新的なアプリケーションの構築に役立ちます。
- AWS トレーニングおよびコース – AWS に関するスキルを磨き、実践的経験を積むために役立つ、職務別の特別コースとセルフペースラボへのリンクです。
- AWS デベロッパー用ツール – デベロッパー用ツール、および資料、コード例、リリースノート、AWS を利用した革新的なアプリケーションの構築に役立つその他の情報を含むリソースへのリンクです。
- AWS Support Center – AWS support ケースを作成および管理するためのハブです。フォーラム、技術上のよくある質問、サービス状態ステータス、AWS Trusted Advisor などの便利なリソースへのリンクも含まれています。
- AWS Support – 1 対 1 での迅速な対応を行うサポートチャネルである AWS Support に関する情報のメインウェブページです。AWS Support は、クラウドでのアプリケーションの構築および実行を支援します。
- お問い合わせ – AWS の請求、アカウント、イベント、不正使用、その他の問題などに関するお問い合わせの受付窓口です。
- AWS サイトの利用規約 – 当社の著作権、商標、お客様のアカウント、ライセンス、サイトへのアクセス、およびその他のトピックに関する詳細情報です。

ドキュメント履歴

以下の表に、Amazon Machine Learning (Amazon ML) の今回のリリースで行われたドキュメントの重要な変更を示します。

- API バージョン: 2015 年 04 月 09 日
- 前回のドキュメントの更新: 2016 年 08 月 02 日

変更	説明	変更日
メトリクスの追加	今回の Amazon ML のリリースでは Amazon ML オブジェクトの新しいメトリクスが追加されました。 詳細については、「オブジェクトのリスト作成」を参照してください。	2016 年 8 月 2 日
複数のオブジェクトの削除	Amazon ML のこのリリースでは、複数の Amazon ML オブジェクトを削除する機能が追加されました。 詳細については、「オブジェクトの削除」を参照してください。	2016 年 7 月 20 日
タグ付けを追加	Amazon ML のこのリリースでは、Amazon ML オブジェクトにタグ付けする機能が追加されました。 詳細については、「Amazon ML オブジェクトのタグ付け」を参照してください。	2016 年 6 月 23 日
Amazon Redshift データソースのコピー	Amazon ML のこのリリースでは、Amazon Redshift データソースの設定を新しい Amazon Redshift データソースにコピーする機能が追加されました。 Amazon Redshift データソースの設定のコピーの詳細については、「データソースのコピー (コンソール)」を参照してください。	2016 年 4 月 11 日
シャッフルの追加	Amazon ML のこのリリースでは、入力データをシャッフルする機能が追加されました。	2016 年 4 月 5 日

変更	説明	変更日
	[シャッフルタイプ] パラメータの使用方法の詳細については、「トレーニングデータのシャッフルタイプ」を参照してください。	
改善された Amazon Redshift でのデータソースの作成	Amazon ML のこのリリースでは、コンソールに Amazon ML データソースを作成して接続が動作していることを確認する際に、Amazon Redshift の設定をテストする機能が追加されました。詳細については、「Amazon Redshift データ (コンソール) でデータソースを作成する」を参照してください。	2016 年 3 月 21 日
改善された Amazon Redshift データスキーマ変換	Amazon ML のこのリリースでは、Amazon Redshift (Amazon Redshift) データスキーマから Amazon ML データスキーマへの変換が改善されます。 Amazon ML での Amazon Redshift の使用に関する詳細は、「Amazon Redshift のデータから Amazon ML データソースを作成する」を参照してください。	2016 年 2 月 9 日
CloudTrail ロギングが追加されました	Amazon ML のこのリリースでは、AWS CloudTrail (CloudTrail) を使用してリクエストを記録する機能が追加されました。 CloudTrail ロギングの使用の詳細については、「AWS CloudTrail での Amazon ML API コールのログ記録」を参照してください。	2015年12月10日
DataRearrangement オプションの追加	Amazon ML のこのリリースでは、入力データをランダムに分割し、補完的なデータソースを作成する機能が追加されました。 `DataRearrangement` パラメータの使用の詳細については、「データ再配置」を参照してください。交差検証で新しいオプションを使用する方法の詳細については、「交差検証」を参照してください。	2015 年 12 月 3 日

変更	説明	変更日
リアルタイム予測の試用	Amazon ML のこのリリースでは、サービスコンソールでリアルタイム予測を試す機能が追加されました。 リアルタイム予測の試用の詳細については、Amazon Machine Learning デベロッパーガイドの「リアルタイム予測のリクエスト」を参照してください。	2015 年 11 月 19 日
新しい リージョン	Amazon ML のこのリリースでは、欧州 (アイルランド) リージョンのサポートが追加されました。 欧州 (アイルランド) リージョンの Amazon ML の詳細については、「Amazon Machine Learning デベロッパーガイド」の「のリージョンとエンドポイント」を参照してください。	2015 年 8 月 20 日
初回リリース	これは、「Amazon ML デベロッパーガイド」の最初のリリースです。	2015 年 4 月 9 日