aws

개발자 가이드

Amazon Machine Learning

버전 Latest

Amazon Machine Learning: 개발자 가이드

Table of Contents

더 이상 Amazon Machine Learning 서비스를 업데이트하거나 새 사용자를 받지 않습니다. 이 설명서는 기존 사용자에 제공되지만 더 이상 업데이트되지 않습니다. 자세한 내용은 머신 러닝이란? 단원을 참조하세요.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.

Amazon Machine Learning이란?

더 이상 Amazon Machine Learning(Amazon ML) 서비스를 업데이트하거나 새 사용자를 받지 않습니다. 이 설명서는 기존 사용자에 제공되지만 더 이상 업데이트되지 않습니다.

AWS 는 이제 Amazon SageMaker AI라는 강력한 클라우드 기반 서비스를 제공하므로 모든 기술 수준의 개발자가 기계 학습 기술을 사용할 수 있습니다. SageMaker AI는 강력한 기계 학습 모델을 생성하는 데 도움이 되는 완전 관리형 기계 학습 서비스입니다. SageMaker AI를 사용하면 데이터 과학자와 개발자가 기계 학습 모델을 구축 및 훈련한 다음 프로덕션 지원 호스팅 환경에 직접 배포할 수 있습니다.

자세한 내용은 SageMaker AI 설명서를 참조하세요.

주제

Amazon Machine Learning에 사용되는 주요 개념

이 단원에서는 다음 주요 개념을 요약하고 Amazon ML에서 이들 개념이 어떻게 사용되는 지를 자세히 설명합니다.

- 데이터 소스은 ML에 대한 데이터 입력과 관련된 메타데이터를 포함시킵니다.
- ML 모델은 입력 데이터에서 추출한 패턴을 사용하여 예측을 생성합니다.
- 평가은 ML 모델의 품질을 측정합니다.
- 배치 예측은 여러 입력 데이터 관측치에 대한 예측을 비동기적으로 생성합니다.
- 실시간 예측은 개별 데이터 관측치에 대한 예측을 동기적으로 생성합니다.

데이터 소스

데이터 소스는 입력 데이터에 대한 메타데이터를 포함하고 있는 객체입니다. Amazon ML은 입력 데이터를 읽고, 해당 속성에 대한 설명 통계를 계산하고, 스키마 및 기타 정보와 함께 통계를 데이터 소스 객체의 일부로 저장합니다. 그 다음, Amazon ML은 데이터 소스를 사용하여 ML 모델을 학습 및 평가하고 배치 예측을 생성합니다.

> **Important**
>
> 데이터 소스는 입력 데이터의 사본은 저장하지 않습니다. 대신 입력 데이터가 있는 Amazon S3 위치에 대한 참조를 저장합니다. Amazon S3 파일을 이동하거나 변경한 경우 Amazon ML은 이 파일에 액세스하거나 이를 사용하여 ML 모델을 생성하거나 평가를 생성하거나 예측을 생성할 수 없습니다.

다음 표에는 데이터 소스와 관련된 용어가 정의되어 있습니다.

용어	정의
속성	관측치 내에서 고유하고 이름이 지정된 속성. 스프레드시트 또는 쉼표로 구분된 값(.csv) 파일과 같은 표 형식 데이터에서 열 헤더는 특성을 나타내며 행은 각 특성에 대한 값을 포함하고 있습니다. 동의어: 변수, 변수 이름, 필드, 열
데이터 소스 이름	(선택 사항) 사람이 읽을 수 있는 데이터 소스 이름을 정의할 수 있습니다. 이러한 이름을 사용하면 Amazon ML 콘솔에서 데이터 소스를 찾고 관리할 수 있습니다.
입력 데이터	데이터 소스에서 참조하는 모든 관측치의 총칭.
위치	입력 데이터의 위치. 현재 Amazon ML은 Amazon S3 버킷, Amazon Redshift 데이터베이스 또는 Amazon Relational Database Service(RDS)의 MySQL 데이터베이스에 저장된 데이터를 사용할 수 있습니다.
관측치	단일 입력 데이터 단위. 예를 들어 사기 거래를 탐지하기 위한 ML 모델을 만드는 경우 입력 데이터는 각각 개별 거래를 나타내는 많은 관측치로 구성될 것입니다. 동의어: 레코드, 예제, 인스턴스, 행
행 ID	(선택 사항) 입력 데이터에서 예측 출력에 포함시킬 속성을 식별하는 플래그(지정된 경우). 이 속성을 사용하면 어떤 예측이 어떤 관측치에 대응하는 지를 보다 쉽게 연결할 수 있습니다. 동의어: 행 식별자

용어	정의
스키마	속성 이름 및 할당된 데이터 형식, 특수 속성의 이름 등을 포함하여 입력 데이터를 해석하는 데 필요한 정보.
Statistics	입력 데이터의 각 속성에 대한 요약 통계 이 통계는 다음 두 가지 목적을 위한 것입니다. Amazon ML 콘솔은 데이터를 한 눈에 파악하고 불규칙성이나 오류를 식별할 수 있도록 그래프로 표시합니다. Amazon ML은 학습 프로세스 중에 이를 사용하여 결과로 생성된 ML 모델의 품질을 개선합니다.
상태 표시기	데이터 소스의 현재 상태(예: 진행 중, 완료됨, 실패)를 나타냅니다.
대상 속성	ML 모델 학습과 관련하여 대상 속성은 입력 데이터에서 "정답"이 포함된 속성의 이름을 식별합니다. Amazon ML은 이를 사용하여 입력 데이터에서 패턴을 발견하고 ML 모델을 생성합니다. 예측 평가 및 생성의 맥락에서 대상 속성이란 학습된 ML 모델을 통해 예측된 값을 가진 속성을 말합니다. 동의어: 대상

ML 모델

ML 모델은 데이터에서 패턴을 찾아 예측을 생성하는 수학적 모델입니다. Amazon ML은 바이너리 분류, 멀티클래스 분류 및 회귀라는 세 가지 유형의 ML 모델을 지원합니다.

다음 표에는 데이터 품질과 관련된 용어가 정의되어 있습니다.

용어	정의
회귀	회귀 ML 모델 학습의 목표는 숫자 값을 예측하는 것입니다.
멀티클래스	멀티클래스 ML 모델 학습의 목표는 미리 정의된 제한적인 허용 값 집합에 속하는 값을 예측하는 것입니다.
바이너리	이진 ML 모델 학습의 목표는 true 또는 false와 같이 두 가지 상태 중 하나만 가질 수 있는 값을 예측하는 것입니다.

용어	정의
모델 크기	ML 모델은 패턴을 캡처하고 저장합니다. ML 모델이 저장하는 패턴이 많을수록 모델 크기는 더 커집니다. ML 모델 크기는 MB 단위로 설명됩니다.
전달 횟수	ML 모델을 학습할 때는 데이터 소스의 데이터를 사용합니다. 학습 프로세스에서 각 데이터 레코드를 두 번 이상 사용하는 것이 유용한 경우가 있습니다. Amazon ML에서 동일한 데이터 레코드를 사용하도록 허용한 횟수를 전달 횟수라고 합니다.
정규화	정규화란 고품질 모델을 얻는 데 사용할 수 있는 기계 학습 기법입니다. Amazon ML은 대부분의 경우에 잘 작동하는 기본 설정을 제공합니다.

평가

평가는 ML 모델의 품질을 측정하고 성능이 좋은지 판단합니다.

다음 표에는 평가와 관련된 용어가 정의되어 있습니다.

용어	정의
모델 인사이트 정보	Amazon ML은 모델의 예측 성능을 평가하는 데 사용할 수 있는 지표와 다양한 인사이트 정보를 제공합니다.
AUC	ROC 곡선하면적(AUC)에서는 부정 예제보다 긍정 예제에 대해 더 높은 점수를 예측하는 모델의 기능을 측정합니다.
매크로 평균 F1 점수	매크로 평균 F1 점수는 멀티클래스 ML 모델의 예측 성능을 평가하는 데 사용됩니다.
RMSE	평균 제곱근 오차(RMSE)는 회귀 ML 모델의 예측 성능을 평가하는 데 사용되는 지표입니다.
커트라인	ML 모델은 숫자 예측 점수를 생성하는 방식으로 작동합니다. 시스템은 커트라인 값을 적용하여 이러한 점수를 0과 1 레이블로 변환합니다.
정확도	정확도는 올바른 예측의 백분율을 측정합니다.

용어	정의
정밀도	정밀도는 검색된 인스턴스(양수로 예측되는 인스턴스) 중 실제 양성이 확인된 인스턴스(거짓 긍정과 반대)의 비율을 나타냅니다. 즉, 선택한 항목 중 긍정에 해당하는 항목 수를 말합니다.
재현율	재현율은 관련 인스턴스의 총 수 중 실제 긍정의 비율(실제 긍정)을 나타냅니다. 즉, 선택된 긍정 항목의 수를 나타냅니다.

배치 예측

배치 예측은 한 번에 모두 실행할 수 있는 일련의 관측치에 대한 예측입니다. 이는 실시간 요구 사항이 없는 예측 분석에 적합합니다.

다음 표에는 배치 예측과 관련된 용어가 정의되어 있습니다.

용어	정의
출력 위치	배치 예측의 결과는 S3 버킷 출력 위치에 저장됩니다.
매니페스트 파일	매니페스트 파일은 각 입력 데이터 파일을 관련 배치 예측 결과와 관련시킵니다. 이 파일은 S3 버킷 출력 위치에 저장됩니다.

실시간 예측

실시간 예측은 대화형 웹, 모바일 또는 데스크톱 애플리케이션과 같이 지연 시간이 짧아야 하는 애플리케이션을 위한 것입니다. ML 모델에 지연 시간이 짧은 실시간 예측 API를 사용하여 실시간으로 예측을 쿼리할 수 있습니다.

다음 표에는 실시간 예측과 관련된 용어가 정의되어 있습니다.

용어	정의
실시간 예측 API	실시간 예측 API는 요청 페이로드에서 단일 입력 관측치를 수용하고 응답에서 예측을 반환합니다.

용어	정의
실시간 예측 엔드포인트	ML 모델을 실시간 예측 API와 함께 사용하려면 실시간 예측 엔드포인트를 생성해야 합니다. 생성된 엔드포인트에는 실시간 예측을 요청하는 데 사용할 수 있는 URL이 포함됩니다.

머신 러닝에 액세스

ML은 다음 방법 중 하나를 사용하여 액세스할 수 있습니다.

ML 콘솔

ML 콘솔에 액세스하려면 https://console.aws.amazon.com/rds/ 에서 관리 콘솔에 로그인한 후 ML 콘솔을 엽니다.

CLI

CLI를 설치 및 구성하는 방법에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 명령줄 인터페이스로 설정 단원을 참조하세요.

Amazon MQ API

MQ API에 대한 자세한 내용은 ML API 참조 단원을 참조하세요.

SDK

SDK에 대한 자세한 내용은 웹 서비스용 도구 단원을 참조하세요.

지역 및 엔드포인트

Amazon Machine Learning(Amazon ML)은 다음 두 지역에서 실시간 예측 엔드포인트를 지원합니다.

지역명	지역	엔드포인트	프로토콜
미국 동부(버지니아 북부)	us-east-1	machinelearning.us-east-1.amazonaws.com	HTTPS

지역명	지역	엔드포인트	프로토콜
유럽(아일랜드)	eu-west-1	machinelearning.eu-west-1.amazonaws.com	HTTPS

모든 지역에서 데이터 세트를 호스팅하고, 모델을 학습 및 평가하고, 예측을 트리거할 수 있습니다.

모든 리소스를 동일한 지역에 보관하는 것이 좋습니다. 입력 데이터가 Amazon ML 리소스와 다른 지역에 있는 경우 지역 간 데이터 전송 요금이 발생합니다. 어느 지역에서든 실시간 예측 엔드포인트를 호출할 수 있지만 호출하는 엔드포인트가 없는 지역에서 엔드포인트를 호출하면 실시간 예측 지연 시간에 영향을 미칠 수 있습니다.

Amazon EKS 요금

AWS 서비스를 사용하면 사용한 만큼만 비용을 지불하면 됩니다. 최소 요금 및 선수금은 없습니다.

Amazon Machine Learning(Amazon ML)은 데이터 통계를 계산하고 모델을 학습 및 평가하는 데 사용된 컴퓨팅 시간에 대해 시간당 요금을 부과하며, 그 이후에는 애플리케이션에 대해 생성된 예측 수에 따라 비용을 지불하게 됩니다. 실시간 예측의 경우 모델 크기에 따라 시간당 예약 용량 요금도 지불합니다.

ML은 ML 콘솔에서만 예측 비용을 추정합니다.

ML 요금에 대한 자세한 내용은 머신 러닝 요금 단원을 참조하세요.

주제

- 배치 예측 비용 추정
- 실시간 예측 비용 추정

배치 예측 비용 추정

배치 예측 생성 마법사를 사용하여 Amazon ML 모델에서 배치 예측을 요청하면 Amazon ML이 이들 예측의 비용을 추정합니다. 추정치를 계산하는 방법은 사용 가능한 데이터의 유형에 따라 달라집니다.

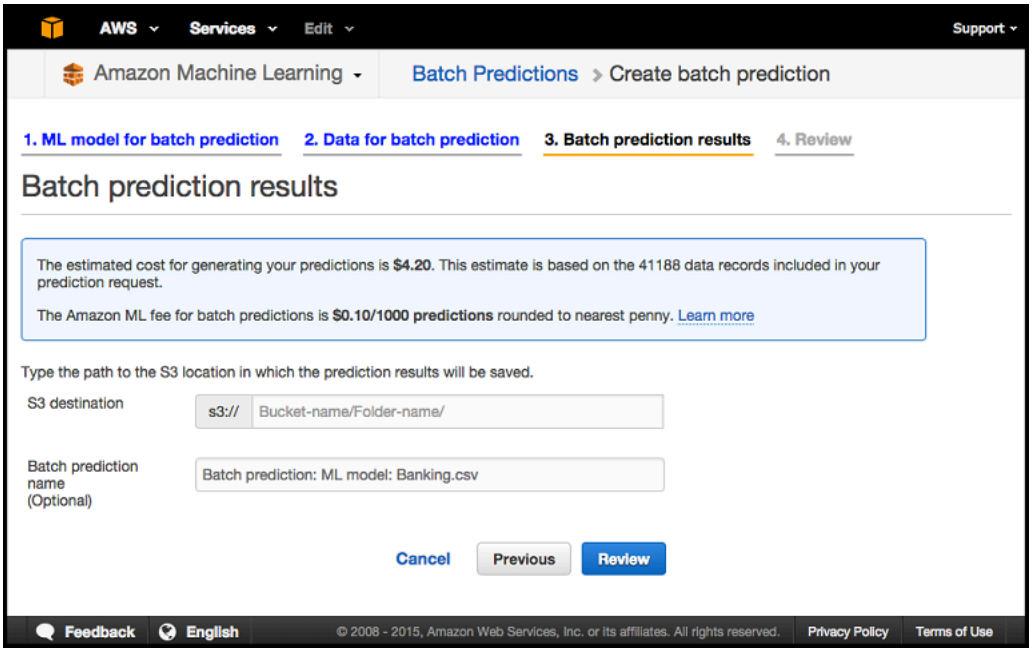
데이터 통계를 사용할 수 있는 경우의 배치 예측 비용 추정

Amazon ML이 예측 요청에 사용된 데이터 소스에 대한 요약 통계를 이미 계산했을 때 가장 정확한 예상 비용이 산출됩니다. 이러한 통계는 Amazon ML 콘솔을 사용하여 생성한 데이터 소스에 대해 항상 계산됩니다. API 사용자는 CreateDataSourceFromS3, CreateDataSourceFromRedshift 또는 CreateDataSourceFromRDS API를 사용하여 프로그래밍 방식으로 데이터 소스를 생성할 때 `ComputeStatistics` 플래그를 `True`로 설정해야 합니다. 통계를 사용할 수 있으려면 데이터 소스가 `READY` 상태에 있어야 합니다.

Amazon ML이 계산하는 통계 중 하나는 데이터 레코드 수입니다. 데이터 레코드 수를 사용할 수 있는 경우 ML 배치 예측 생성 마법사는 데이터 레코드 수에 배치 예측 요금을 곱하여 예측 수를 추정합니다.

실제 비용은 다음과 같은 이유로 이 추정치와 다를 수 있습니다.

- 일부 데이터 레코드는 처리에 실패할 수 있습니다. 실패한 데이터 레코드의 예측에 대해서는 요금이 청구되지 않습니다.
- 추정치에는 기존에 존재하던 크레딧이나 AWS에서 적용한 기타 조정 사항이 고려되지 않습니다.



데이터 크기만 사용할 수 있는 경우의 배치 예측 비용 추정

배치 예측을 요청했는데 요청 데이터 소스에 대한 데이터 통계를 사용할 수 없는 경우 Amazon ML은 다음을 기준으로 비용을 추정합니다.

- 데이터 소스 검증 중에 계산되고 지속되는 총 데이터 크기

- 평균 데이터 레코드 크기. Amazon ML이 데이터 파일의 처음 100MB를 읽고 파싱하여 추정한 데이터 레코드 크기

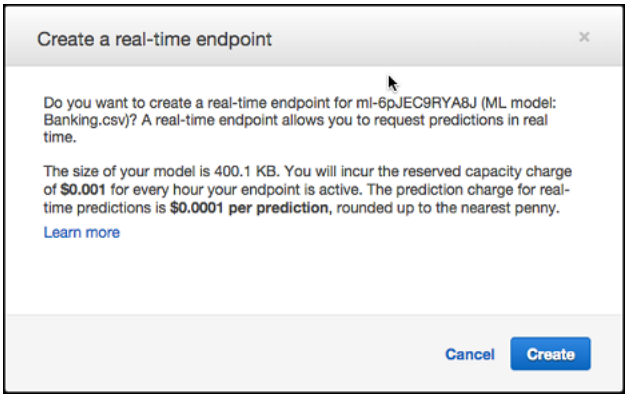
배치 예측 비용을 추정하기 위해 Amazon ML은 총 데이터 크기를 평균 데이터 레코드 크기로 나눕니다. 데이터 파일의 첫 번째 레코드가 평균 레코드 크기를 정확하게 나타내지 못할 수 있으므로 이 비용 예측 방법은 사용 가능한 데이터 레코드 수가 많을 때 사용되는 방법보다 정확하지 않습니다.

데이터 통계와 데이터 크기를 모두 사용할 수 없는 경우의 배치 예측 비용 추정

데이터 통계나 데이터 크기를 모두 사용할 수 없는 경우 Amazon ML이 배치 예측 비용을 추정하지 못합니다. 일반적으로 배치 예측을 요청하는 데 사용하는 데이터 소스가 Amazon ML에서 아직 검증되지 않은 경우가 여기에 해당됩니다. 이는 Amazon Redshift(Amazon Redshift) 또는 Amazon Relational Database Service(Amazon RDS) 쿼리를 기반으로 하는 데이터 소스를 생성했는데 데이터 전송이 아직 완료되지 않았거나 데이터 소스 생성이 계정에서 다른 작업 뒤에 대기하는 경우에 발생할 수 있습니다. 이 경우 Amazon ML 콘솔에서 배치 예측 요금을 알려줍니다. 추정치 없이 배치 예측 요청을 진행하거나, 예측에 사용되는 데이터 소스가 진행 중 또는 준비 상태가 된 후에 마법사를 취소하고 반환하도록 선택할 수 있습니다.

실시간 예측 비용 추정

Amazon ML 콘솔을 사용하여 실시간 예측 엔드포인트를 생성하면 예상 예약 용량 요금이 표시됩니다. 이 요금은 예측 처리를 위해 엔드포인트를 예약하는 데 드는 지속적인 요금입니다. 이 요금은 서비스 요금 페이지에 설명된 대로 모델 크기에 따라 달라집니다. 또한 표준 Amazon ML 실시간 예측 요금에 대한 안내도 받게 됩니다.



기계 학습 개념

기계 학습(ML)은 과거 데이터를 사용하여 더 나은 비즈니스 결정을 내리는 데 도움이 될 수 있습니다. ML 알고리즘은 데이터에서 패턴을 찾아내서 이를 통해 수학적 모델을 구성합니다. 그런 다음 모델을 사용하여 미래 데이터에 대한 예측을 수행할 수 있습니다. 예를 들어, 기계 학습 모델을 적용할 수 있는 방법 중 하나는 고객의 과거 행동을 기반으로 특정 제품을 구매할 가능성을 예측하는 것입니다.

주제

Amazon Machine Learning을 통한 비즈니스 문제 해결

Amazon Machine Learning을 사용하면 실제 답변의 기존 예제가 있는 문제에 기계 학습을 적용할 수 있습니다. 예를 들어 Amazon Machine Learning을 사용하여 이메일이 스팸인지 예측하려면 스팸 또는 스팸 아님으로 올바르게 레이블이 지정된 이메일 예제를 수집해야 합니다. 그런 다음 기계 학습을 사용하여 이러한 이메일 예제를 일반화하여 새 이메일이 스팸일 가능성을 예측할 수 있습니다. 실제 답변으로 레이블이 지정된 데이터를 통해 학습하는 이러한 접근 방식을 감독형 기계 학습이라고 합니다.

감독형 ML 접근 방식은 바이너리 분류(두 가지 가능한 결과 중 하나 예측), 멀티클래스 분류(두 개 이상의 결과 중 하나 예측), 회귀(숫자 값 예측)와 같은 특정 기계 학습 작업에 사용할 수 있습니다.

바이너리 분류 문제의 예제:

- 고객이 이 제품을 구매할까요, 아니면 구매하지 않을까요?
- 이 이메일은 스팸인가요, 아니면 스팸이 아닌가요?
- 이 제품은 책인가요 아니면 가축인가요?
- 이 리뷰는 고객이 작성했나요, 아니면 로봇이 작성했나요?

멀티클래스 분류 문제의 예:

- 이 제품은 책인가요, 영화인가요 아니면 의류인가요?
- 이 영화는 로맨틱 코미디인가요, 다큐멘터리인가요, 스릴러인가요?

- 이 고객이 가장 관심을 갖는 제품 범주는 무엇인가요?

회귀 분류 문제의 예:

- 내일 시애틀의 기온은 어떻습니까?
- 이 제품의 판매량은 몇 대입니까?
- 이 고객이 애플리케이션 사용을 중단하기까지 며칠이 남았습니까?
- 이 주택은 어떤 가격에 팔릴까요?

기계 학습을 사용해야 하는 경우

ML이 모든 유형의 문제에 대한 해결책이 아니라는 점에 유의하세요. ML 기법을 사용하지 않고도 강력한 솔루션을 개발할 수 있는 경우도 있습니다. 예를 들어, 데이터 기반 학습 없이 프로그래밍할 수 있는 간단한 규칙, 계산 또는 미리 정해진 단계를 사용하여 대상 값을 결정할 수 있다면 ML은 필요하지 않습니다.

기계 학습은 다음과 같은 상황에 사용합니다.

- 규칙을 코딩할 수 없는 경우: 이메일이 스팸인지 스팸이 아닌지 인식하는 것과 같은 많은 인간 작업은 간단한 (결정론적) 규칙 기반 솔루션으로는 적절하게 해결할 수 없습니다. 답변에 여러 가지 요인이 영향을 미칠 수 있습니다. 규칙이 너무 많은 요소에 의존하고 이러한 규칙 중 많은 부분이 겹치거나 매우 세밀하게 조정해야 하는 경우 사람이 규칙을 정확하게 코딩하기가 곧 어려워집니다. ML을 사용하면 이 문제를 효과적으로 해결할 수 있습니다.
- 확장할 수 없는 경우: 수백 개의 이메일을 수동으로 인식하여 스팸 여부를 판단할 수 있을 수도 있습니다. 그러나 수백만 개의 이메일에서 이 작업은 지루한 작업이 됩니다. ML 솔루션은 대규모 문제를 처리하는 데 효과적입니다.

Machine Learning 애플리케이션 빌드

ML 애플리케이션 빌드는 일련의 단계가 관련된 반복적인 프로세스입니다. ML 애플리케이션을 빌드하려면 다음과 같은 일반적인 단계를 수행합니다.

1. 관측되는 항목 및 모델이 예측하기를 원하는 대답과 관련하여 핵심 ML 문제를 구성합니다.
2. ML 모델 학습 알고리즘이 사용하기에 적합하도록 데이터를 수집, 정리 및 준비합니다. 데이터를 시각화하고 분석해서 안전성 검사를 실행하여 데이터의 품질을 확인하고 데이터를 이해합니다.

3. 원시 데이터(입력 변수) 및 대답(대상)은 고도로 예측 가능한 모델을 학습시키는 데 사용할 수 있는 방식으로 표현되지 않는 경우가 많습니다. 따라서 일반적으로 원시 변수에서 더 많은 예측 입력 표현 또는 특성을 구성하려고 시도해야 합니다.
4. 그 결과로 나타난 특성을 학습 알고리즘에 제공하여 모델을 빌드하고 모델 빌드에서 제외된 데이터에 대한 모델의 품질을 평가합니다.
5. 모델을 사용하여 새 데이터 인스턴스에 대한 대상 대답의 예측을 생성합니다.

문제 수립

기계 학습의 첫 번째 단계는 레이블 또는 대상 대답이라고 알려진 예측 대상을 결정하는 것입니다. 제품을 제조하려는 시나리오를 상상해 볼 때 각 제품을 제조하는 것에 대한 사용자의 결정은 잠재적인 판매량에 따라 달라집니다. 이 시나리오에서는 각 제품이 얼마나 많이 구매될 것인지 예측할 수 있습니다(판매 횟수 예측). 기계 학습을 사용하여 이 문제를 정의하는 방법에는 여러 가지가 있습니다. 문제를 정의하는 방법에 대한 선택은 사용 사례 또는 비즈니스 요구 사항에 따라 달라집니다.

각 제품에 대한 고객의 구매 건수를 예측하려고 합니까(이 경우 대상은 숫자이며 사용자는 회귀 문제를 해결하고 있음)? 또는 어떤 제품이 10개 이상 판매되는지 예측하려고 합니까(이 경우 대상은 이진수이고 사용자는 바이너리 분류 문제를 해결하고 있음)?

문제가 지나치게 복잡해지지 않게 하고 요구 사항에 맞는 가장 단순한 솔루션을 만드는 것이 중요합니다. 하지만 정보의 손실, 특히 과거 대답에 대한 정보의 손실을 방지하는 것도 중요합니다. 여기에서 이전의 실제 판매 수를 "10 이상" 또는 "10보다 적은" 이진 변수로 변환하면 중요한 정보가 손실됩니다. 예측하기에 가장 적절한 대상을 결정하는 데 시간을 투자하면 질문에 대답하지 못하는 모델을 빌드하지 않게 해줍니다.

레이블이 지정된 데이터 수집

ML 문제는 데이터, 특히 대상 대답을 이미 알고 있는 많은 데이터로 시작하는 것이 좋습니다(예제 또는 관측치). 이미 알고 있는 데이터를 레이블이 지정된 데이터라고 합니다. 감독되는 ML에서 알고리즘은 사용자가 제공하는 레이블이 있는 예제를 학습하도록 지시합니다.

데이터의 각 예제/관측치는 다음 두 가지 요소를 포함해야 합니다.

- 대상 - 예측하려는 대답. 대상(정답)으로 레이블이 지정된 데이터를 학습할 ML 알고리즘에 제공합니다. 그런 다음 학습된 ML 모델을 사용하여 대상 대답을 모르는 데이터에 대해 이 대답을 예측합니다.
- 변수/특성 - 대상 대답을 예측하는 패턴을 식별하는 데 사용할 수 있는 예제의 속성입니다.

예를 들어 이메일 분류 문제의 경우 대상은 이메일이 스팸인지 여부를 나타내는 레이블입니다. 변수의 예로는 이메일을 보낸 사람, 이메일 본문 텍스트, 제목 줄의 텍스트, 이메일을 보낸 시간 및 보낸 사람과 받는 사람 간의 이전 서신이 있습니다.

데이터는 종종 레이블이 지정된 형식으로 사용될 수 없습니다. 변수 및 대상을 수집하고 준비하는 것은 종종 ML 문제를 해결하기 위한 가장 중요한 단계입니다. 예제 데이터는 모델을 사용하여 예측을 수행할 때 보유할 데이터를 나타내야 합니다. 예를 들어 이메일이 스팸인지 스팸이 아닌 지를 예측하려면 기계 학습 알고리즘에 대한 긍정(스팸 이메일) 값과 부정(스팸이 아닌 이메일) 값을 모두 수집하여 두 이메일 유형을 구별할 패턴을 찾을 수 있어야 합니다.

레이블이 지정된 데이터가 있으면 알고리즘 또는 소프트웨어에서 허용하는 형식으로 변환해야 할 수 있습니다. 예를 들어 Amazon ML을 사용하려면, CSV 파일의 한 행을 구성하는 각 예제와 함께 쉼표로 구분된(CSV) 형식으로 데이터를 변환해야 합니다. 이때 각 열은 하나의 입력 변수를 포함하고 한 열은 대상 대답을 포함합니다.

데이터 분석

레이블이 지정된 데이터를 ML 알고리즘에 제공하기 전에 데이터를 검사하여 문제를 식별하고 사용 중인 데이터에 대한 인사이트 정보를 얻는 것이 좋습니다. 모델의 예측 능력은 사용자가 제공하는 데이터만큼의 양호한 수준을 갖습니다.

데이터를 분석할 때 다음 사항을 항상 고려해야 합니다.

- 변수 및 대상 데이터 요약 - 변수에서 취하는 값과 데이터에서 가장 큰 부분을 차지하는 값을 이해하는 것이 유용합니다. 해결하려는 문제에 대한 주제 관련 전문가가 요약을 실행할 수 있습니다. 스스로 또는 주제 관련 전문가에게 다음과 같이 질문합니다. 데이터가 예상과 일치합니까? 데이터 수집 관련 문제가 있는 것처럼 보입니까? 대상의 한 클래스가 다른 클래스보다 빈번하게 나타납니까? 누락된 값이나 잘못된 데이터가 예상보다 많습니까?
- 변수-대상 상관 관계 - 상관 관계가 높다는 것은 변수와 대상 클래스 간에 관계가 있다는 의미이므로 변수와 대상 클래스 간의 상관 관계를 아는 것이 도움이 됩니다. 일반적으로 상관 관계가 높은 변수는 예측 가능성이 높은 변수이므로(신호) 포함시키고, 상관 관계가 낮은 변수는 관련성이 낮으므로 제외시킵니다.

Amazon ML에서는 데이터 소스를 생성하고 결과 데이터 보고서를 검토하여 데이터를 분석할 수 있습니다.

특성 처리

데이터 요약 및 시각화를 통해 데이터를 파악한 후에는 변수를 더욱 의미 있는 것으로 변환하고 싶어 할 수 있습니다. 이를 특성 처리라고 합니다. 예를 들어 이벤트가 발생한 날짜와 시간을 캡처하는 변수가 있다고 가정해 보겠습니다. 이 날짜와 시간은 다시 발생하지 않으므로 대상을 예측하는 데 유용하지 않습니다. 하지만 이 변수를 해당 날짜의 시간, 요일 및 월을 나타내는 특성으로 변환하는 경우, 이러한 변수는 이벤트가 특정 시간, 요일 또는 월에 발생하는 경향이 있는지 학습하기에 유용할 수 있습니다. 학습하기 위해 좀 더 일반화가 가능한 데이터 요소를 형성하는 이러한 특성 처리는 예측 모델을 크게 개선할 수 있습니다.

일반적인 특성 처리의 다른 예:

- 누락되었거나 잘못된 데이터를 더 의미 있는 값으로 대체합니다(예: 제품 유형 변수에 누락된 값이 실제로 책이라는 것을 사용자가 알고 있는 경우 제품 유형의 누락된 값을 모두 책 값으로 대체할 수 있음). 누락된 값을 대체하기 위해 사용되는 일반적인 전략은 누락된 값을 평균 값 또는 중간 값으로 대체하는 것입니다. 누락된 값을 대체하기 위한 전략을 선택하기 전에 데이터를 파악하는 것이 중요합니다.
- 한 변수와 다른 변수의 데카르트 곱을 형성합니다. 예를 들어 인구 밀도(도시, 교외, 농촌)와 주(워싱턴, 오레곤, 캘리포니아)와 같은 두 가지 변수가 있는 경우, 이 두 변수의 데카르트 곱에 대한 결과로 형성된 특성에 유용한 정보가 있을 수 있습니다(urban_Washington, suburban_Washington, rural_Washington, urban_Oregon, suburban_Oregon, rural_Oregon, urban_California, suburban_California, rural_California).
- 카테고리에 숫자 변수를 비닝(binning)하는 것과 같은 비선형 변환. 많은 경우에 숫자 특성과 대상 간의 관계는 선형적이지 않습니다(특성 값은 대상에 따라 점차 증가하거나 감소하지 않음). 이 경우 다양한 범위의 숫자 특성을 나타내는 카테고리 특성으로 숫자 특성을 비닝하는 것이 유용할 수 있습니다. 각 카테고리 특성(빈)은 대상과의 선형 관계를 갖도록 모델링될 수 있습니다. 예를 들어 연속적인 숫자 특징 age가 책을 구입할 가능성과 선형적으로 관련이 없다는 것을 알고 있다고 가정해 보겠습니다. 사용자는 대상과의 관계를 더욱 정확하게 캡처할 수 있는 범주형 특성으로 age를 비닝할 수 있습니다. 숫자 변수에 대한 최적의 빈 수는 변수의 특징과 대상과의 관계에 따라 달라지며, 이는 실험을 통해 가장 효과적으로 결정됩니다. Amazon ML은 제안된 레시피의 데이터 통계를 기반으로 숫자 특성에 대한 최적의 빈 수를 제안합니다. 추천 레시피에 대한 세부 정보는 개발자 안내서를 참조하세요.
- 도메인별 특성(예: 별도의 변수로 길이, 너비 및 높이를 가지며, 이 세 가지 변수의 곱으로 새 볼륨 특성을 만들 수 있음).
- 변수별 특성. 텍스트 특성, 웹 페이지의 구조를 캡처하는 특성 또는 문장 구조와 같은 일부 변수 유형은 구조와 컨텍스트를 추출하는 데 도움이 되는 일반적인 처리 방법을 가집니다. 예를 들어 "the

fox jumped over the fence"라는 텍스트의 n-gram을 형성하는 것은 unigram(the, fox, jumped, over, fence) 또는 bigram(the fox, fox jumped, jumped over, over the, the fence)으로 표현될 수 있습니다.

보다 관련성이 높은 특성을 포함시키면 예측 능력을 개선하는 데 도움이 됩니다. 분명하게도 "신호" 또는 예측 영향이 있는 특성을 항상 미리 알 수는 없습니다. 따라서 대상 레이블과 잠재적으로 관련될 수 있는 모든 특성을 포함시키고 모델 학습 알고리즘이 가장 강한 상관 관계가 있는 특성을 선택하게 하는 것이 좋습니다. Amazon ML에서 특성 처리는 모델을 생성할 때 레시피에서 지정할 수 있습니다. 사용 가능한 특성 처리 목록은 개발자 안내서를 참조하세요.

데이터를 학습 및 평가 데이터로 분리

ML의 근본적인 목표는 모델을 학습시키는 데 사용된 데이터 인스턴스를 그 이상으로 일반화하는 것입니다. 우리는 모델 평가를 통해 해당 모델이 학습되지 않은 데이터에 대한 패턴 일반화의 품질을 평가하려고 합니다. 하지만 미래 인스턴스는 알 수 없는 대상 값을 가지며 미래 인스턴스에 대한 예측의 정확성을 지금 확인할 수 없으므로, 이미 알고 있는 데이터 중 일부를 향후 데이터의 프록시로 사용해야 합니다. 학습에 사용된 것과 동일한 데이터를 사용하여 모델을 평가하는 것은 유용하지 않은데, 이는 일반화하는 것과 대조적으로 학습 데이터를 "기억"할 수 있는 모델에 대한 보상을 하기 때문입니다.

일반적인 전략은 사용 가능한 분류된 데이터를 모두 가져와서 학습 및 평가 하위 집합으로 분리하는 것입니다. 보통 학습에 70~80%, 평가에 20~30%의 비율이 사용됩니다. ML 시스템은 학습 데이터를 사용하여 모델을 학습시켜서 패턴을 확인하고, 평가 데이터를 사용하여 학습된 모델의 예측 품질을 평가합니다. ML 시스템은 다양한 지표를 사용하여 평가 데이터 세트의 예측을 참 값(실측 정보라고 함)과 비교하는 방식으로 예측 성능을 평가합니다. 보통 평가 하위 집합에 대해 "최적"의 모델을 사용하여 대상 대답을 모르는 미래 인스턴스에 대한 예측을 수행합니다.

Amazon ML은 Amazon ML 콘솔을 통해 학습용으로 전송된 데이터를 학습에 70%, 평가에 30%로 분리합니다. 기본적으로 Amazon ML은 입력 데이터의 처음 70%를 소스 데이터에 표시된 순서대로 학습 데이터 소스에 사용하고 나머지 30%는 평가 데이터 소스에 사용합니다. 또한 Amazon ML에서는 처음 70%를 사용하고 이 무작위 하위 집합의 보완을 평가에 사용하는 대신 소스 데이터의 70%를 무작위로 선택하여 학습할 수 있습니다. Amazon ML API를 사용하여 사용자 지정 분할 비율을 지정하고 Amazon ML 외부로 분할된 학습 및 평가 데이터를 제공할 수 있습니다. 또한 Amazon ML은 데이터를 분할하기 위한 전략을 제공합니다. 분할 전략에 대한 자세한 내용은 데이터 분할 단원을 참조하세요.

모델 학습

이제 ML 알고리즘(학습 알고리즘)에 학습 데이터를 제공할 준비가 되었습니다. 해당 알고리즘은 변수를 대상에 매핑하는 학습 데이터 패턴을 통해 학습하고 이러한 관계를 캡처하는 모델을 출력합니다. 그러면 ML 모델을 사용하여 대상 대답을 모르는 새로운 데이터에 대한 예측을 얻을 수 있습니다.

선형 모델

다양한 ML 모델을 사용할 수 있습니다. Amazon ML은 한 가지 유형의 ML 모델, 즉 선형 모델을 학습합니다. 선형 모델이라는 용어는 모델이 특성의 선형 조합으로 지정된다는 것을 의미합니다. 학습 데이터를 기반으로 학습 프로세스는 각 특성에 대해 하나의 가중치를 계산하여 대상 값을 예측하거나 추정할 수 있는 모델을 생성합니다. 예를 들어 고객이 구입할 보험 금액이 대상이고, 연령과 소득이 변수인 경우 단순한 선형 모델은 다음과 같습니다.

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

학습 알고리즘

학습 알고리즘의 과제는 모델의 가중치를 학습하는 것입니다. 가중치는 모델이 학습하고 있는 패턴이 데이터의 실제 관계를 반영할 가능성을 설명합니다. 학습 알고리즘은 손실 함수 및 최적화 기법으로 구성됩니다. 손실은 ML 모델이 제공하는 대상의 추정치가 대상과 정확하게 일치하지 않을 때 발생하는 페널티입니다. 손실 함수는 이 페널티를 단일 값으로 수량화합니다. 최적화 기법은 손실 최소화를 추구합니다. Amazon Machine Learning에서는 세 가지 유형의 예측 문제 각각에 대해 세 가지 손실 함수를 하나씩 사용합니다. Amazon ML에서 사용되는 최적화 기법은 온라인 SGD(Stochastic Gradient Descent)입니다. SGD는 학습 데이터에 대해 순차적으로 전달하며, 각 전달 중에 손실을 최소화하는 최적의 가중치를 위한 목적으로 한 번에 한 가지 예를 사용하여 특성 가중치를 업데이트합니다.

Amazon ML은 다음과 같은 학습 알고리즘을 사용합니다.

- 바이너리 분류의 경우 Amazon ML은 로지스틱 회귀(로지스틱 손실 함수 + SGD)를 사용합니다.
- 멀티클래스 분류의 경우 Amazon ML은 다항 로지스틱 회귀(다항 로지스틱 손실 + SGD)를 사용합니다.
- 회귀의 경우 Amazon ML은 선형 회귀(제곱 손실 함수 + SGD)를 사용합니다.

학습 파라미터

Amazon ML 학습 알고리즘은 결과 모델의 품질을 조정할 수 있는 하이퍼파라미터 또는 학습 파라미터라고 불리는 파라미터를 허용합니다. 하이퍼파라미터에 따라 Amazon ML은 설정을 자동 선택하거나 하이퍼파라미터의 정적 기본값을 제공합니다. 기본 하이퍼파라미터 설정은 일반적으로 유용한 모델을 생성하지만, 하이퍼파라미터 값을 변경하여 모델의 예측 성능을 향상시킬 수 있습니다. 다음에 이어지는 단원에서는 Amazon ML이 생성한 것과 같이 선형 모델의 학습 알고리즘과 관련된 일반적인 하이퍼파라미터를 설명합니다.

학습률

학습률은 SGD(Stochastic Gradient Descent) 알고리즘에서 사용되는 상수 값입니다. 학습률은 알고리즘이 최적의 가중치에 도달(수렴)하는 속도에 영향을 줍니다. SGD 알고리즘은 확인하는 모든 데이터 예에 대해 선형 모델의 가중치를 업데이트합니다. 이 업데이트의 크기는 학습률에 의해 제어됩니다. 학습률이 너무 높으면 가중치가 최적의 솔루션에 접근하지 못할 수 있습니다. 값이 너무 작으면 최적의 가중치에 접근하기 위해 많은 전달이 필요한 알고리즘으로 이어집니다.

Amazon ML에서 학습률은 데이터를 기반으로 자동 선택됩니다.

모델 크기

입력 특성이 많은 경우, 사용 가능한 데이터의 패턴 수가 많으면 대규모 모델로 이어질 수 있습니다. 대규모 모델은 학습 중 또는 예측을 생성할 때 모델을 유지하는 데 더 많은 RAM이 필요하다는 것과 같은 실질적인 의미를 가집니다. Amazon ML에서는 L1 정규화를 사용하거나 최대 크기를 지정하여 모델 크기를 구체적으로 제한함으로써 모델 크기를 줄일 수 있습니다. 모델 크기를 너무 많이 줄이면 모델의 예측 능력이 감소할 수 있습니다.

기본 모델 크기에 대한 내용은 학습 파라미터: 유형 및 기본 값 단원을 참조하세요. 정규화에 대한 자세한 내용은 정규화 단원을 참조하세요.

전달 횟수

SGD 알고리즘은 학습 데이터를 순차적으로 전달합니다. `Number of passes` 파라미터는 알고리즘이 학습 데이터에 대해 수행하는 전달 횟수를 제어합니다. 전달이 많을수록 데이터에 더 적합한 모델이 되지만(학습률이 너무 높지 않은 경우), 전달 횟수가 증가하면 이점이 줄어들게 됩니다. 더 작은 데이터 세트의 경우 전달 횟수를 크게 늘릴 수 있게 되어 학습 알고리즘이 데이터에 더 적합하도록 효과적으로 조정할 수 있습니다. 매우 큰 데이터 세트의 경우에도 단일 전달로 충분할 수 있습니다.

기본 전달 횟수에 대한 자세한 정보는 학습 파라미터: 유형 및 기본 값 단원을 참조하세요.

데이터 셔플링

Amazon ML에서 SGD 알고리즘은 학습 데이터의 행 순서에 따른 영향을 받기 때문에 사용자는 데이터를 셔플링해야 합니다. 학습 데이터를 셔플링하면 SGD 알고리즘이 전체 데이터 범위가 아닌 첫 번째로 확인하는 데이터 유형에 최적화된 솔루션을 피하는 데 도움이 되므로 더 나은 수준의 ML 모델이 됩니다. 셔플링은 데이터 순서를 혼합하기 때문에 SGD 알고리즘이 연속적으로 너무 많은 관측치에서 한 가지 유형의 데이터만 접하지 않습니다. 연속적으로 많은 가중치 업데이트에서 한 가지 유형의 데이터만 확인하는 경우, 업데이트 규모가 너무 클 수 있기 때문에 알고리즘이 새 데이터 유형의 모델 가중치를 수정하지 못할 수 있습니다. 또한 데이터가 임의로 제공되지 않으면 알고리즘이 모든 데이터 유형에 대해 최적화된 솔루션을 빠르게 찾을 수 없으며, 경우에 따라 알고리즘이 최적화된 솔루션을 아예 찾지

못할 수도 있습니다. 학습 데이터를 셔플링하면 알고리즘이 최적의 솔루션으로 빠르게 수렴하는 데 도움이 됩니다.

예를 들어 ML 모델을 학습시켜서 제품 유형을 예측하려고 하며, 학습 데이터에는 영화, 장난감 및 비디오 게임 제품 유형이 포함되어 있다고 가정해 보겠습니다. 데이터를 Amazon S3에 업로드하기 전에 제품 유형 열 기준으로 데이터를 정렬하는 경우, 알고리즘은 제품 유형 기준에 따라 알파벳순으로 데이터를 확인합니다. 알고리즘이 먼저 영화에 대한 모든 데이터를 확인하면 ML 모델이 영화에 대한 패턴을 학습하기 시작합니다. 그런 다음 모델이 장난감에 대한 데이터를 접하면 해당 알고리즘이 수행하는 모든 업데이트는 장난감 제품 유형에 대한 모델에 적합하게 됩니다. 이러한 업데이트로 인해 영화에 적합한 패턴이 저하되는 경우에도 마찬가지입니다. 영화에서 장난감 유형으로 갑자기 전환하면 제품 유형을 정확하게 예측하는 방법을 학습하지 못하는 모델이 생성될 수 있습니다.

기본 셔플링 유형에 대한 자세한 정보는 학습 파라미터: 유형 및 기본 값 단원을 참조하세요.

정규화

정규화를 사용하면 극단적인 가중치 값에 페널티를 부과하여 선형 모델이 학습 데이터 예에 과적합(즉, 패턴을 일반화하는 대신 암기하는 것)되는 것을 방지할 수 있습니다. L1 정규화는 적은 가중치를 갖게 될 특성의 가중치를 0으로 만들어 모델에서 사용되는 특성의 수를 줄이는 효과를 가집니다. 결과적으로 L1 정규화는 희소 모델을 생성하고 모델의 노이즈를 줄입니다. L2 정규화는 전체적으로 더 적은 가중치로 이어지며, 입력 특성 간의 상관 관계가 높을 때 가중치를 안정화합니다. `Regularization type` 및 `Regularization amount` 파라미터를 사용하여, 적용되는 L1 또는 L2 정규화의 정도를 제어합니다. 매우 큰 정규화 값을 사용하면 모든 특성의 가중치가 0이 되어 모델이 패턴을 학습하지 못하게 될 수 있습니다.

기본 정규화 값에 대한 자세한 정보는 학습 파라미터: 유형 및 기본 값 단원을 참조하세요.

모델 정확성 평가

ML 모델의 목표는 학습 중에 표시되는 데이터를 기억하는 대신 보이지 않는 데이터를 효과적으로 일반화하는 패턴을 학습하는 것입니다. 모델이 있는 경우 모델 학습에 사용하지 않은 미확인 예제에 대해서도 모델 성능을 확인해야 합니다. 이를 위해서는 모델을 사용하여 평가 데이터 세트(데이터 유지)에 대한 대답을 예측한 다음 예측된 대상을 실제 대답(실측 정보)과 비교합니다.

모델의 예측 정확성을 측정하기 위해 ML에서 여러 지표가 사용됩니다. 정확성 지표의 선택은 ML 작업에 따라 다릅니다. 이러한 지표를 검토하여 모델 성능이 좋은지 판단해야 합니다.

바이너리 분류

많은 바이너리 분류 알고리즘의 실제 출력은 예측 점수입니다. 점수는 주어진 관측치가 긍정 클래스에 속한다는 시스템의 확실성을 나타냅니다. 이 점수의 소비자는 관측치를 긍정으로 분류할지 또는 부정

으로 분류할 지를 결정하기 위해 분류 임계값(커트라인)을 선택하여 점수를 해석하고 점수를 이 값과 비교합니다. 임계값보다 점수가 높은 관측치는 긍정 클래스로 예측되고, 임계값보다 점수가 낮으면 부정 클래스로 예측됩니다.

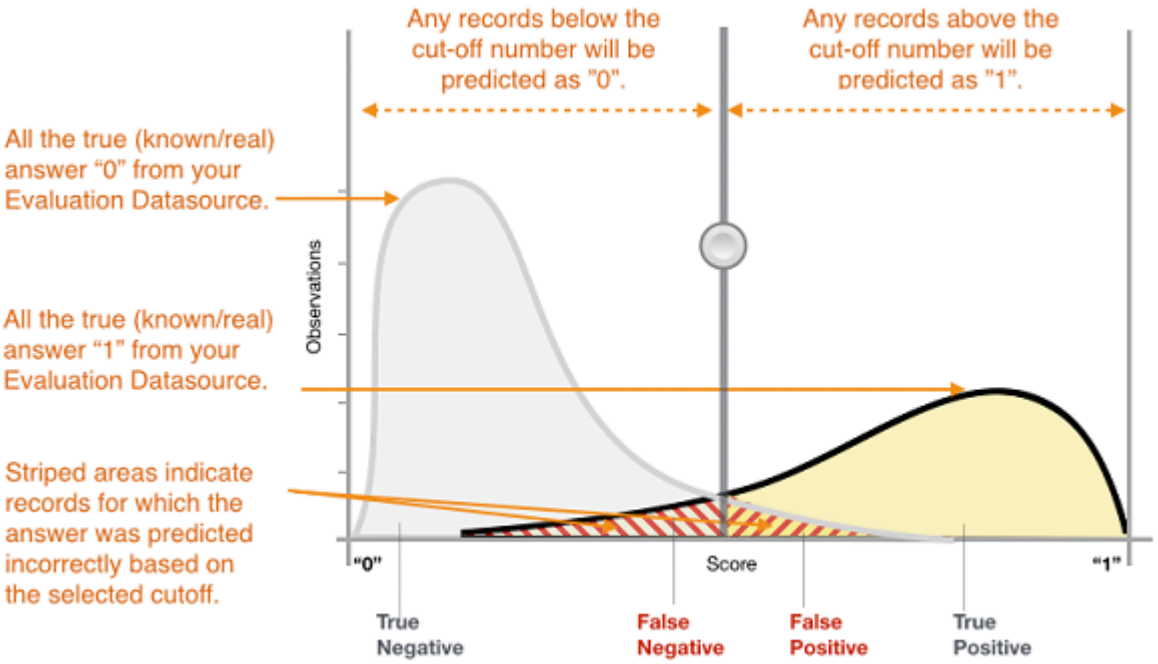


그림 1: 바이너리 분류 모델의 점수 분포

이제 예측은 실제로 알려진 대답 및 예측된 대답을 바탕으로 올바른 긍정 예측(참 긍정), 올바른 부정 예측(참 부정), 잘못된 긍정 예측(거짓 긍정) 및 잘못된 부정 예측(거짓 부정)의 4가지 그룹으로 분류됩니다.

바이너리 분류 정확성 지표는 두 가지 유형의 올바른 예측과 두 가지 유형의 오류를 정량화합니다. 일반적인 지표는 정확도(ACC), 정밀도, 재현율, 거짓 긍정 비율, F1 측정치입니다. 각 지표는 예측 모델의 다른 측면을 측정합니다. 정확도(ACC)는 올바른 예측의 비율을 측정합니다. 정밀도는 긍정으로 예측되는 사례 중 실제 긍정의 비율을 측정합니다. 재현율은 긍정으로 예측된 실제 긍정의 수를 측정합니다. F1 측정치는 정밀도와 재현율의 조화 평균입니다.

AUC는 다른 유형의 지표입니다. AUC에서는 부정적인 예보다 긍정적인 예에 대해 더 높은 점수를 예측하는 모델의 기능을 측정합니다. AUC는 선택한 임계값과는 별개이므로, 임계값을 선택하지 않고도 AUC 지표에서 모델의 예측 성능을 파악할 수 있습니다.

비즈니스 문제에 따라 이러한 지표의 특정 하위 집합에 대해 효과적으로 수행되는 모델에 더 관심을 가질 수 있습니다. 예를 들어 다음과 같이 두 비즈니스 애플리케이션은 ML 모델에 대해 매우 다른 요구 사항을 가질 수 있습니다.

- 한 애플리케이션은 실제로 긍정(높은 정밀도)인 긍정 예측에 대해 매우 높은 수준의 확신을 가져야 하며, 일부 긍정 사례를 부정(보통 수준의 재현율)으로 잘못 분류할 수 있어야 합니다.
- 다른 애플리케이션은 가능한 많은 수의 긍정 사례(높은 재현율)를 정확하게 예측해야 하며, 긍정(보통 수준의 정밀도)으로 잘못 분류된 일부 부정 사례를 수용해야 합니다.

Amazon ML에서 관측치는 [0,1]의 범위로 예측된 점수를 얻습니다. 예제를 0 또는 1로 분류할지 결정하기 위한 점수 임계값은 기본적으로 0.5로 설정됩니다. Amazon ML을 사용하면 다양한 점수 임계값 선택이 미치는 영향을 검토하고 비즈니스 요구 사항에 맞는 적절한 임계값을 선택할 수 있습니다.

멀티클래스 분류

바이너리 분류 문제 프로세스와는 달리 예측을 하기 위해 점수 임계값을 선택할 필요가 없습니다. 예측된 대답은 가장 높은 예측 점수를 가진 클래스(예: 레이블)입니다. 일부 경우에는 대답이 높은 점수로 예측되는 경우에만 예측된 대답을 사용해야 합니다. 이 경우 예상 대답을 수용할지 여부에 따라 예상 점수에 대한 임계값을 선택할 수 있습니다.

멀티클래스에 사용되는 일반적인 지표는 바이너리 분류 사례에서 사용되는 지표와 동일합니다. 지표는 모든 다른 클래스를 두 번째 클래스에 속한 것으로 그룹화한 후 바이너리 분류 문제로 간주하여 각 클래스에 대해 계산됩니다. 그런 다음 이진수 지표를 모든 클래스에 대해 평균화하여 매크로 평균(각 클래스를 동등하게 간주) 또는 가중 평균(클래스 빈도로 가중치 처리) 지표를 얻습니다. Amazon ML에서 매크로 평균 F1 측정치는 멀티클래스 분류기의 예측 성공을 평가하기 위해 사용됩니다.

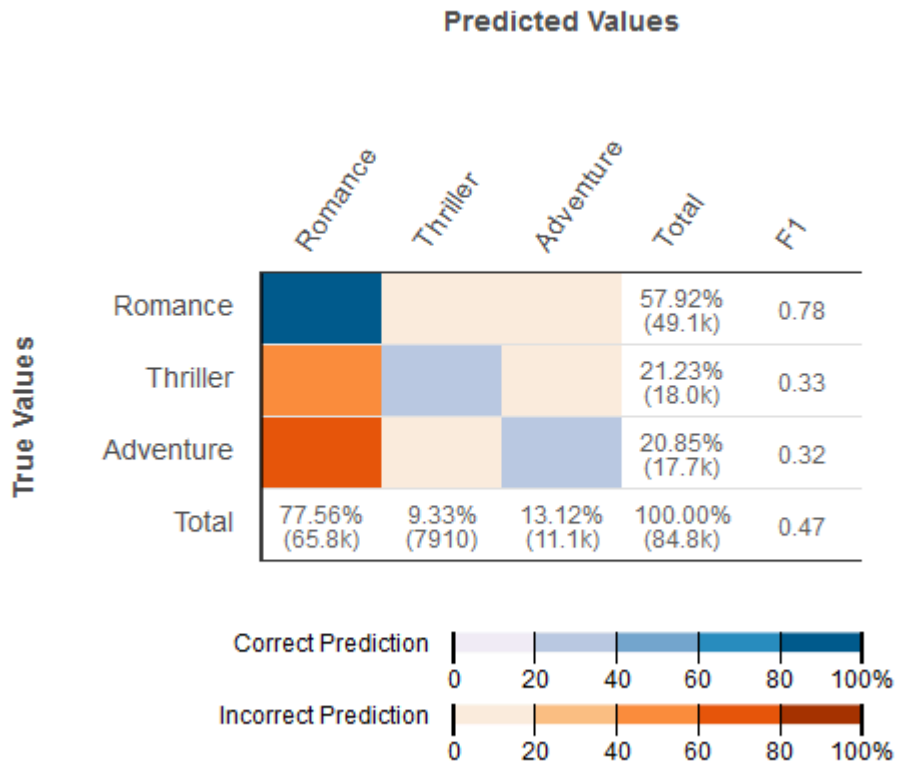


그림 2: 멀티클래스 분류 모델의 혼동 행렬

멀티클래스 문제에 대해 혼동 행렬을 검토하는 것이 유용합니다. 혼동 행렬은 평가 데이터의 각 클래스와 올바른 예측 및 잘못된 예측의 수 또는 백분율을 표시하는 표입니다.

회귀

회귀 작업의 경우 일반적인 정확도 지표는 RMSE(평균 제곱근 오차) 및 MAPE(평균 절대 백분율 오차)입니다. 이러한 지표는 예상 수치 대상과 실제 수치 대답(실측 정보) 간의 거리를 측정합니다. Amazon ML에서 RMSE 지표는 회귀 모델의 예측 정확성을 평가하기 위해 사용됩니다.

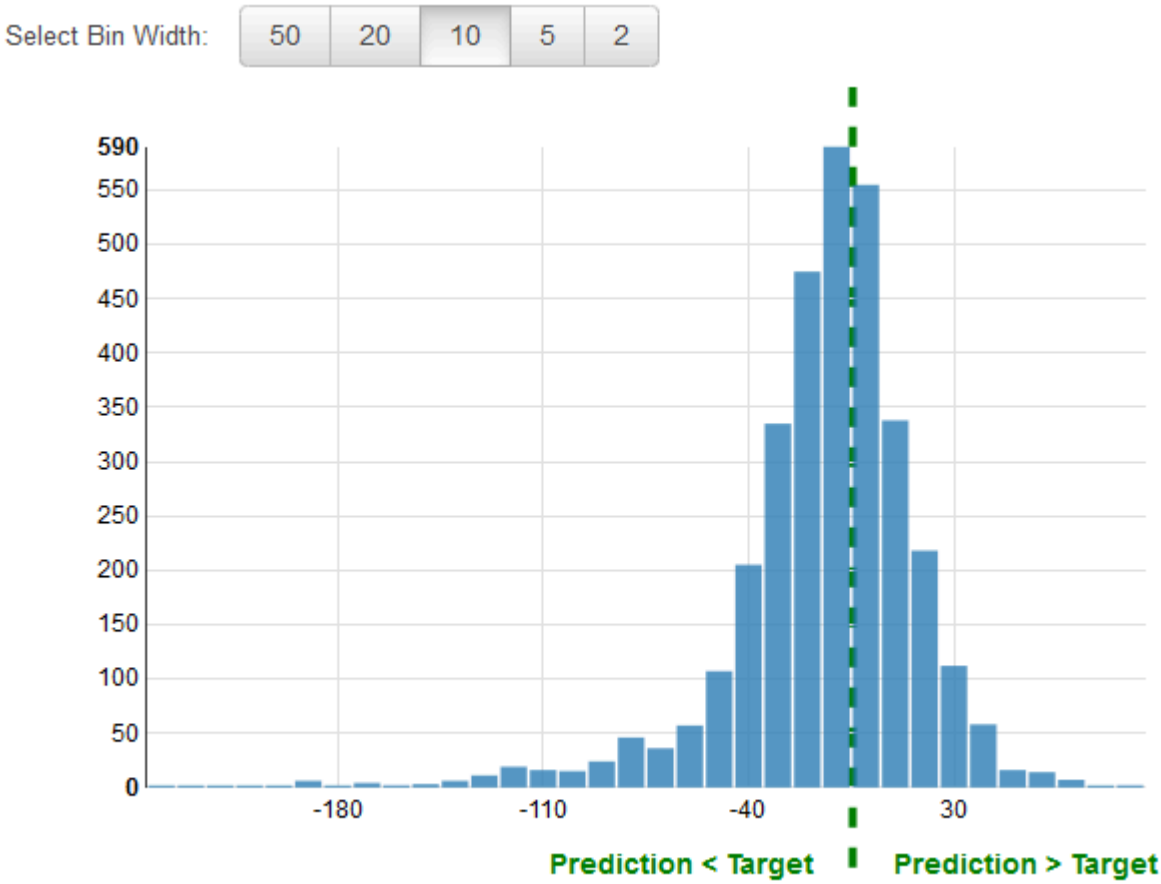


그림 3: 회귀 모델의 잔차 분포

회귀 문제에 대해 잔차를 검토하는 것이 일반적입니다. 평가 데이터에서 관측치에 대한 잔차는 실제 대상과 예측된 대상 간의 차이입니다. 잔차는 대상 중 모델이 예측할 수 없는 부분을 나타냅니다. 긍정 잔차는 모델이 대상을 과소평가하고 있다는 것을 나타냅니다(실제 대상이 예측된 대상보다 큼). 부정 잔차는 모델이 과대평가하고 있다는 것을 나타냅니다(실제 대상이 예측된 대상보다 작음). 종 모양으로 분포되고 0에 중심을 둔, 평가 데이터에 대한 잔차 히스토그램은 모델이 임의의 방식으로 오류를 만들고 대상 값의 특정 범위를 체계적으로 예측할 수 없다는 것을 나타냅니다. 잔차가 0에 중심을 둔 종 모양을 형성하지 않으면 모델의 예측 오차에 약간의 구조를 가집니다. 모델에 변수를 더 추가하면 모델이 현재 모델이 캡처하지 않은 패턴을 캡처하는 데 도움이 될 수 있습니다.

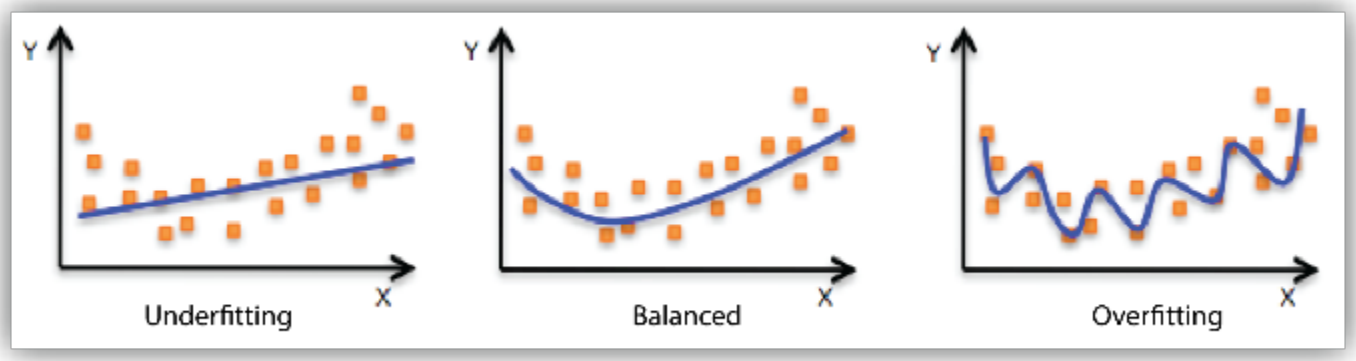
모델 정확성 개선

요구 사항에 맞는 ML 모델을 얻는 것에는 보통 이 ML 프로세스를 반복하고 몇 가지 변형을 시도하는 것이 포함됩니다. 첫 번째 반복에서 예측 능력이 매우 높은 모델을 얻지 못할 수도 있고, 더 나은 예측을 위해 모델을 개선해야 할 수도 있습니다. 성능을 개선하기 위해 다음 단계를 반복할 수 있습니다.

1. 데이터 수집: 학습 예제의 수 증가
2. 특성 처리: 더 많은 변수 추가 및 특성 처리 향상
3. 모델 파라미터 조정: 학습 알고리즘에서 사용하는 학습 파라미터의 대체 값 고려

모델 적합성: 과소적합과 과적합 비교

잘못된 모델 정확성에 대한 근본 원인을 이해하려면 모델 적합성을 이해하는 것이 중요합니다. 이러한 이해를 통해 올바른 수정 단계를 수행할 수 있습니다. 학습 데이터 및 평가 데이터의 예측 오차를 확인하여 예측 모델에 대한 학습 데이터의 과소적합 또는 과적합 여부를 결정할 수 있습니다.



모델이 학습 데이터에 대해 좋은 성능을 나타내지 않을 때, 모델은 학습 데이터에 과소적합한 것입니다. 이는 모델이 입력 예제(종종 X라고 함)와 대상 값(종종 Y라고 함) 간의 관계를 캡처할 수 없기 때문입니다. 모델이 학습 데이터에 대해 좋은 성능을 나타내지만 평가 데이터에 대해서는 좋은 성능을 나타내지 않을 때, 모델은 학습 데이터에 과적합한 것입니다. 모델이 확인한 데이터를 암기하고 있으며, 미확인 예제는 일반화할 수 없기 때문입니다.

학습 데이터에 대한 성능이 좋지 않은 이유는 모델이 너무 단순해서 대상을 잘 설명할 수 없기 때문입니다(입력 특성의 표현이 충분하지 않음). 모델 유연성을 개선하여 성능을 향상시킬 수 있습니다. 모델 유연성을 개선하려면 다음을 시도합니다.

- 새로운 도메인별 특성 및 더 많은 특성 데카르트 곱을 추가하고 사용되는 특성 처리 유형을 변경합니다(예: n-gram 크기 증가).
- 사용된 정규화 정도를 줄입니다.

모델이 학습 데이터에 과적합한 경우 모델 유연성을 줄이는 조치를 취하는 것이 좋습니다. 모델 유연성을 줄이려면 다음을 시도합니다.

- 특성 선택: 특성 조합을 더 적게 사용하고 n-gram 크기를 줄이며 숫자 속성 빈 수를 줄이는 것을 고려합니다.
- 사용된 정규화 정도를 높입니다.

학습 알고리즘에 학습 데이터가 충분하지 않기 때문에 학습 및 테스트 데이터의 정확성이 떨어질 수 있습니다. 다음 작업을 수행하여 성능을 향상시킬 수 있습니다.

- 학습 데이터 예제 수를 높입니다.
- 기존 학습 데이터에 대한 전달 횟수를 높입니다.

모델을 사용하여 예측 수행

이제 제대로 수행하는 ML 모델이 있으므로 이 모델을 사용하여 예측을 수행합니다. Amazon Machine Learning에는 다음 두 가지 방식으로 모델을 사용하여 예측을 수행합니다.

배치 예측

배치 예측은 한 번에 한 집합의 관측치에 대한 예측을 생성한 다음 일정 비율 또는 수의 관측치에 대해 조치를 취하려고 할 때 유용합니다. 일반적으로 이러한 애플리케이션에 대해 짧은 지연 시간이 요구됩니다. 예를 들어 제품에 대한 광고 캠페인의 일부로 대상 고객을 결정하려는 경우, 모든 고객에 대한 예측 점수를 얻고 모델의 예측을 정렬하여 구입 가능성이 가장 높은 고객을 식별하면 대상은 구입 가능성이 가장 높은 상위 5% 고객이 될 수 있습니다.

온라인 예측

온라인 예측 시나리오는 지연 시간이 짧은 환경에서 다른 예제와 독립적으로 각 예제에 대해 하나씩 예측을 생성하려는 경우에 사용됩니다. 예를 들어 예측을 사용하여 특정 거래가 사기 거래인지 여부를 즉시 결정할 수 있습니다.

새 데이터에 대한 모델 재학습

모델이 정확하게 예측하기 위해서는 예측을 수행하는 데이터가 모델을 학습시킨 데이터와 유사한 분포를 가져야 합니다. 데이터 분포는 시간이 지나면서 이동할 수 있으므로, 모델 배포는 일회성 작업이 아니라 연속적인 프로세스입니다. 수신 데이터를 지속적으로 모니터링하고 데이터 분포가 기존의 학습 데이터 분포에서 크게 벗어난 경우 새로운 데이터로 모델을 재학습시키는 것이 좋습니다. 데이터 분포의 변화를 감지하기 위한 모니터링 데이터가 높은 오버헤드를 갖는 경우, 더 단순한 전략은 모델을 주기적으로(매일, 매주 또는 매월) 학습시키는 것입니다. Amazon ML에서 모델을 다시 학습시키려면 새 학습 데이터를 기반으로 새로운 모델을 생성해야 합니다.

Amazon Machine Learning 프로세스

다음 표에서는 Amazon ML 콘솔을 사용하여 이 문서에 설명되어 있는 ML 프로세스를 수행하는 방법을 설명합니다.

ML 프로세스	Amazon MI 작업
데이터 분석	Amazon ML에서 데이터를 분석하려면 데이터 소스를 만들고 데이터 인사이트 정보 페이지를 검토합니다.
데이터를 학습 및 평가 데이터로 분리	Amazon ML은 데이터 소스를 분리하여 데이터의 70%는 모델 학습에 그리고 30%는 모델의 예측 성능 평가에 사용할 수 있습니다. ML 모델 생성 마법사를 기본 설정 상태로 사용하면 Amazon ML이 자동으로 데이터를 분리합니다. ML 모델 생성 마법사를 사용자 지정 설정 상태로 사용하고 ML 모델을 평가하도록 선택하면 Amazon ML에서 데이터를 분리하고 데이터의 30%에 대해 평가를 실행할 수 있는 옵션이 표시됩니다.
학습 데이터 셔플링	ML 모델 생성 마법사를 기본 설정 상태로 사용하면 Amazon ML이 자동으로 데이터를 셔플링합니다. Amazon ML로 데이터를 가져오기 전에 데이터를 셔플링할 수도 있습니다.
프로세서 기능	학습 및 일반화를 위한 최적의 형식으로 학습 데이터를 모으는 프로세스를 특성 변환이라고 합니다. ML 모델 생성 마법사를 기본 설정 상태로 사용하면 Amazon ML이 데이터에 맞는 특성 처리 설정을 추천합니다. 특성 처리 설정을 지정하려면 ML 모델 생성 마법사의 사용자 지정 옵션을 사용하고 특성 처리 레시피를 제공합니다.
모델 학습	ML 모델 생성 마법사를 사용하여 Amazon ML에서 모델을 생성하면 Amazon ML이 모델을 학습시킵니다.
모델 파라미터 선택	Amazon ML에서는 모델의 예측 성능에 영향을 미치는 네 가지 파라미터, 즉 모델 크기, 전달 횟수, 셔플링 유형, 정규화를 조정할 수 있습니다. ML 모델 생성 마법사를 사용하여 ML 모델을 생성하고 사용자 지정 옵션을 선택하면 이들 파라미터를 설정할 수 있습니다.

ML 프로세스	Amazon MI 작업
모델 성능 평가	평가 생성 마법사를 사용하면 모델의 예측 성능을 평가할 수 있습니다.
특성 선택	Amazon ML 학습 알고리즘은 학습 프로세스에 크게 기여하지 않는 특성을 삭제할 수 있습니다. 이러한 특성을 삭제하고 싶다는 것을 나타내려면 ML 모델을 생성할 때 `L1 regularization` 파라미터를 선택합니다.
예측 정확도에 대한 점수 임계값 설정	평가 보고서에서 다양한 점수 임계값에서 모델의 예측 성능을 검토한 다음 비즈니스 애플리케이션에 따라 점수 임계값을 설정합니다. 점수 임계값은 모델이 예측 일치를 정의하는 방법을 결정합니다. 숫자를 조정하여 거짓 긍정 및 거짓 부정을 제어할 수 있습니다.
모델 사용	모델을 사용하면 배치 예측 생성 마법사를 통해 한 묶음의 관측치에 대한 예측을 얻을 수 있습니다. 또는 ML 모델이 `Predict` API를 사용하여 실시간 예측을 처리하도록 지원하여 필요에 따라 개별 관측치에 대한 예측을 얻을 수 있습니다.

Amazon Machine Learning 설정

Amazon Machine Learning을 처음 사용하려면 먼저 AWS 계정이 필요합니다. 계정이 없으면 AWS에 등록 단원을 참조하세요.

AWS에 등록

Amazon Web Services(AWS)에 등록하면 Amazon ML을 포함하여 AWS의 모든 서비스에 AWS 계정이 자동으로 등록됩니다. 사용자에게는 사용한 서비스에 대해서만 요금이 청구됩니다. 이미 AWS 계정이 있다면 이 단계를 건너뜁니다. AWS 계정이 없는 경우에는 다음 절차에 따라 계정을 만듭니다.

AWS 계정에 등록하려면

1. https://aws.amazon.com/ 으로 이동하여 등록을 선택합니다.
2. 화면에 표시되는 지시 사항을 따릅니다.

등록 절차 중 전화를 받고 전화 키패드를 사용하여 PIN을 입력하는 과정이 있습니다.

자습서: Amazon ML을 사용하여 마케팅 제안에 대한 응답 예측

Amazon Machine Learning(Amazon ML)을 사용하면 예측 모델을 빌드 및 학습시키고 확장 가능한 클라우드 솔루션에서 애플리케이션을 호스팅할 수 있습니다. 이 자습서에서는 Amazon ML 콘솔을 사용하여 데이터 소스를 생성하고, 기계 학습(ML) 모델을 빌드하고, 모델을 사용하여 애플리케이션에서 사용할 수 있는 예측을 생성하는 방법을 보여줍니다.

샘플 연습에서는 타겟 마케팅 캠페인의 잠재 고객을 식별하는 방법을 보여주지만 동일한 원칙을 적용하여 다양한 ML 모델을 만들고 사용할 수 있습니다. 샘플 실습을 완료하려면 캘리포니아 대학교 어바인 캠퍼스(UCI)의 기계 학습 리포지토리에서 공개적으로 사용할 수 있는 은행 및 마케팅 데이터 세트를 사용해야 합니다. 이러한 데이터 세트에는 고객에 대한 일반 정보와 고객이 이전 마케팅 담당자에 대해 어떻게 반응했는지에 대한 정보가 포함되어 있습니다. 이 데이터를 사용하여 새 상품인 예금 증서(CD)라고도 하는 은행 정기 예금에 가입할 가능성이 가장 높은 고객을 식별할 수 있습니다.

> **Warning**
>
> 이 자습서는 AWS 프리 티어에 포함되어 있지 않습니다. ML 요금에 대한 자세한 내용은 머신 러닝 요금 단원을 참조하세요.

사전 조건

개별 교습을 수행하려면 AWS 계정이 있어야 합니다. AWS 계정이 없는 경우 머신 러닝 설정 단원을 참조하세요.

단계

- 1단계: 데이터 준비
- 2단계: 학습 데이터 세트 생성
- 3단계: ML 모델 생성
- 4단계: ML 모델의 예측 성능 검토 및 점수 임계값 설정
- 5단계: ML 모델을 사용하여 예측 생성
- 6단계: 정리

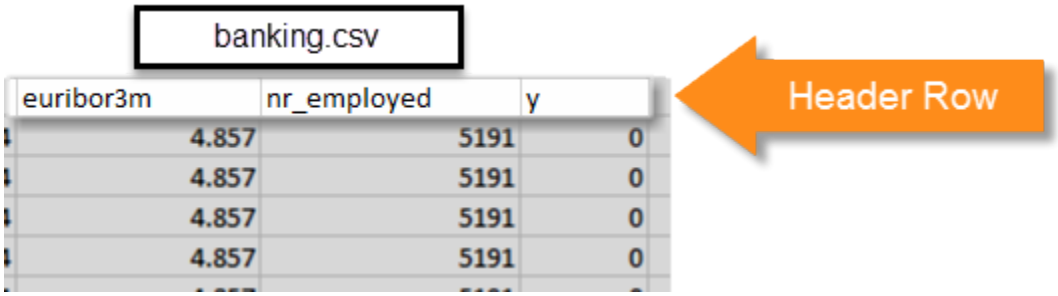
1단계: 데이터 준비

기계 학습에서는 일반적으로 학습 프로세스를 시작하기 전에 데이터를 확보하고 형식이 올바른지 확인합니다. 본 자습서의 목적에 맞게 UCI 기계 학습 리포지토리에서 샘플 데이터 세트를 확보했고 ML 지침에 맞게 형식을 지정했고 다운로드할 수 있게 만들었습니다. 이 주제의 절차에 따라 Amazon Simple Storage Service(Amazon S3) 저장 위치에서 데이터 세트를 다운로드하고 자체 S3 버킷으로 업로드합니다.

ML 형식 요구 사항은 Amazon ML의 데이터 형식에 대한 이해 단원을 참조하세요.

데이터 세트를 다운로드하려면

1. banking.zip을 클릭하여 은행 정기 예금과 유사한 상품을 구입한 고객의 과거 데이터가 포함된 파일을 다운로드합니다. 폴더의 압축을 풀고 banking.csv 파일을 컴퓨터에 저장합니다.
2. banking-batch.zip을 클릭하여 잠재 고객이 귀하의 제안에 응답하는지 여부를 예측하는 데 사용할 파일을 다운로드합니다. 폴더의 압축을 풀고 banking-batch.csv 파일을 컴퓨터에 저장합니다.
3. `banking.csv`를 엽니다. 데이터의 행 및 열을 확인할 수 있습니다. 헤더 열에는 각 열의 속성 이름이 들어 있습니다. 속성은 각 고객의 특정한 특성을 설명하는 이름이 지정된 고유한 속성입니다. 예를 들어 nr_employed는 고객의 취업 상태를 나타냅니다. 각 행은 단일 고객에 대한 관측치 모음을 나타냅니다.

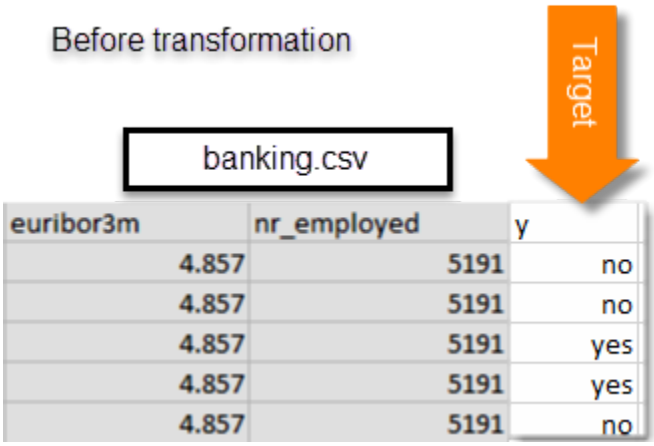


ML 모델에게서 “이 고객이 새 상품에 가입할 것입니까?”라는 질문의 대답을 얻고 싶습니다. `banking.csv` 데이터 세트에서 이 질문에 대한 대답은 1(yes인 경우) 또는 0(no인 경우)의 값을 포함하는 속성 y입니다. ML이 예측 방법을 학습하길 원하는 속성을 대상 속성이라고 합니다.

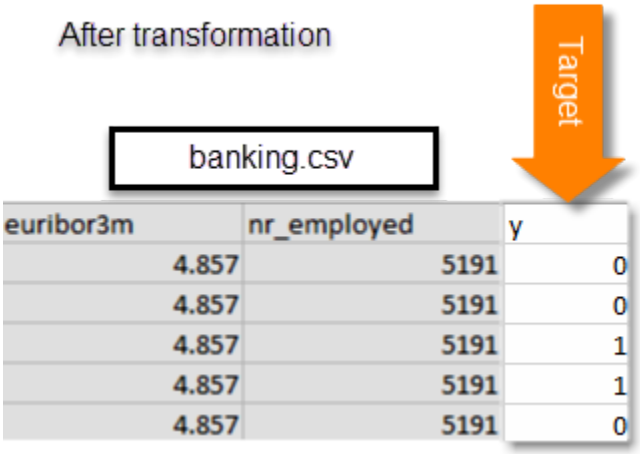
> **Note**
>
> 속성 y는 이진 속성입니다. 이 속성은 두 개의 값 중 하나만 포함할 수 있으며 이 경우 0 또는 1입니다. 원본 UCI 데이터 세트에서 y 속성은 Yes 또는 No입니다. 원본 데이터 세트를 편집해 드렸습니다. 이제 yes를 의미하는 속성 y의 모든 값은 1이며, no를 의미하는 모든

값은 0입니다. 자체 데이터를 사용하는 경우 이진 속성에 다른 값을 사용할 수 있습니다. 유효한 값에 대한 자세한 내용은 AttributeType 필드 사용 단원을 참조하세요.

다음 예제에서는 속성 y의 값을 이진 속성 0 및 1로 변경하기 전후의 데이터를 보여줍니다.



euribor3m	nr_employed	y
4.857	5191	no
4.857	5191	no
4.857	5191	yes
4.857	5191	yes
4.857	5191	no



euribor3m	nr_employed	y
4.857	5191	0
4.857	5191	0
4.857	5191	1
4.857	5191	1
4.857	5191	0

`banking-batch.csv` 파일에는 y 속성이 없습니다. ML 모델을 생성한 후에는 이 모델을 사용하여 해당 파일의 각 레코드에 대해 y를 예측합니다.

그 다음 `banking.csv` 및 `banking-batch.csv` 파일을 S3에 업로드합니다.

Amazon S3 위치에 파일을 업로드하려면

1. 에 로그인AWS Management Console하고 https://console.aws.amazon.com/s3/ Amazon S3 콘솔을 엽니다.
2. 모든 버킷 목록에서 버킷을 생성하거나 파일을 업로드할 위치를 선택합니다.
3. 탐색 모음에서 업로드를 선택합니다.
4. 파일 추가를 선택합니다.

5. 대화 상자에서 바탕 화면으로 이동하여 `banking.csv` 및 `banking-batch.csv`를 선택한 다음 열기를 선택합니다.

이제 학습 데이터 소스를 생성할 준비가 되었습니다.

2단계: 학습 데이터 세트 생성

Simple Storage Service(S3) 위치에 `banking.csv` 데이터 세트를 업로드한 후 데이터 세트를 사용하여 학습 데이터 소스를 생성합니다. 데이터 소스는 입력 데이터와 입력 데이터에 대한 중요한 메타데이터의 위치를 포함하고 있는 Amazon Machine Learning(Amazon ML) 객체입니다. Amazon ML은 데이터 소스를 ML 모델 학습 및 평가 등의 작업에 사용합니다.

데이터 소스를 생성하려면 다음을 제공합니다.

- 데이터의 Amazon S3 위치 및 데이터에 대한 액세스 권한
- 데이터에 있는 속성의 이름과 각 속성의 유형(숫자, 텍스트, 범주형 또는 이진)을 포함하고 있는 스키마
- Amazon ML에서 예측 방법을 학습하도록 하려는 답변(대상 속성)이 들어 있는 속성의 이름

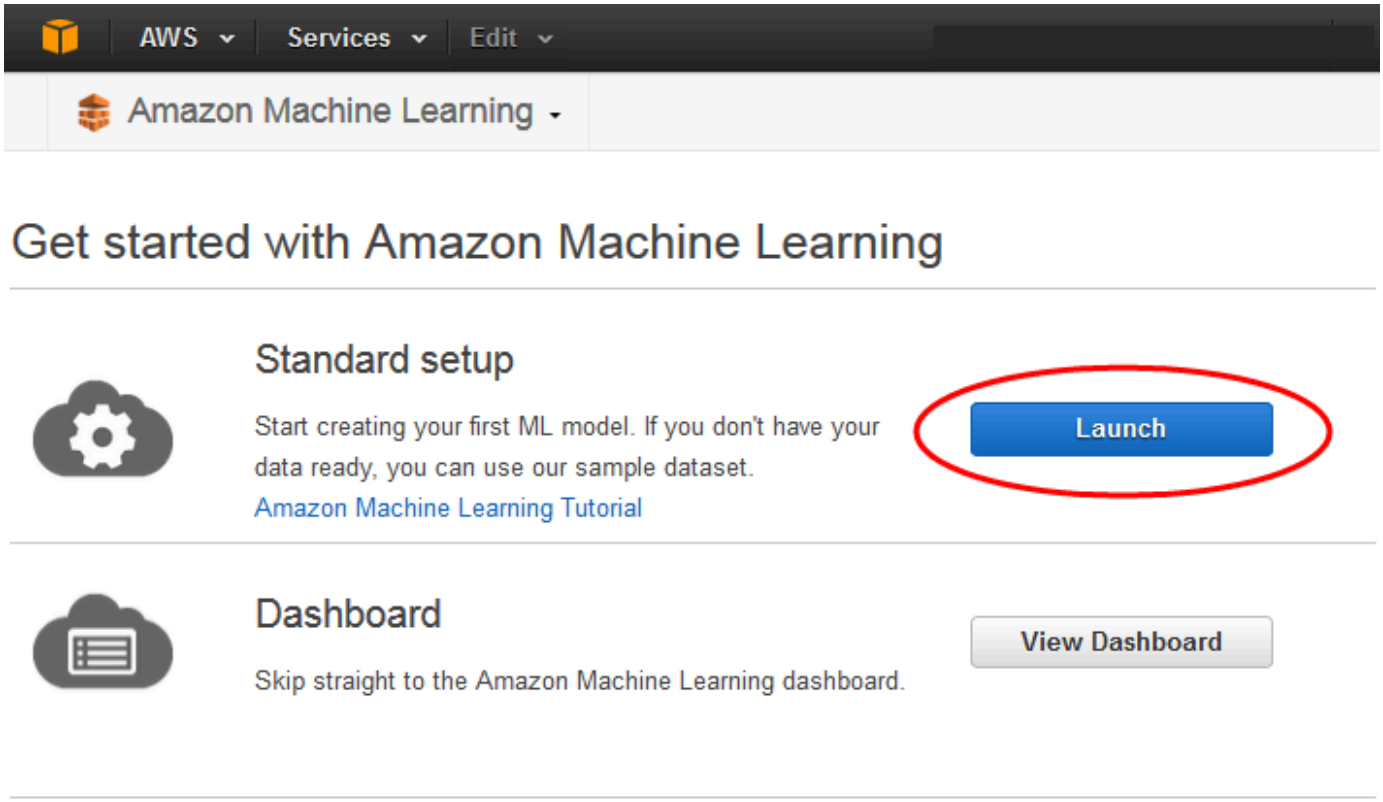
> **Note**
>
> 데이터 소스는 데이터를 실제로 저장하지 않고 참조만 합니다. Amazon S3에 저장된 파일을 이동하거나 변경하지 마십시오. 파일을 이동하거나 변경할 경우 Amazon ML이 해당 파일에 액세스하여 ML 모델을 만들거나 평가를 생성하거나 예측을 생성할 수 없습니다.

학습 데이터 소스를 만들려면

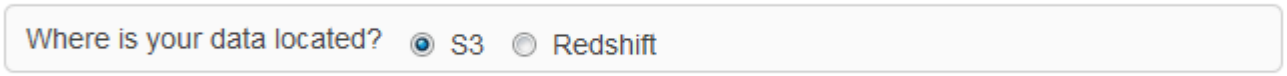
1. https://console.aws.amazon.com/machinelearning/에서 머신 러닝 콘솔을 엽니다.
2. Get started를 선택합니다.

> **Note**
>
> 이 자습서에서는 Amazon ML을 처음 사용하는 것으로 가정합니다. 이전에 ML을 사용해 본 적이 있다면 ML 대시보드의 새로 생성... 드롭다운 목록을 사용하여 새 데이터 소스를 생성해도 됩니다.

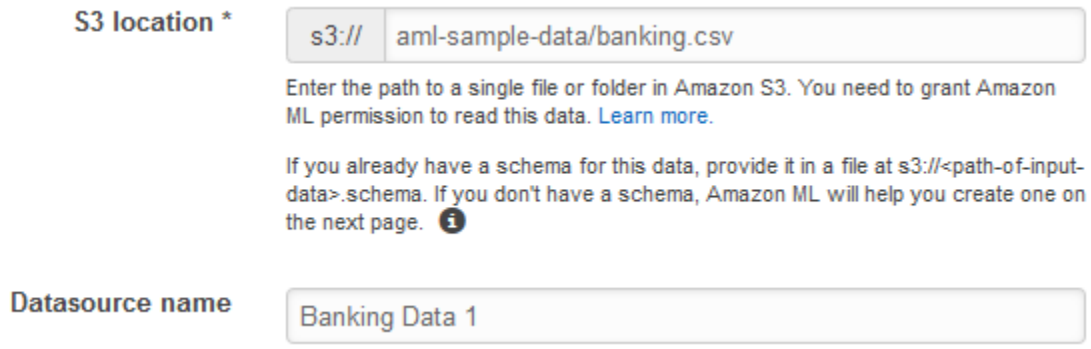
3. 머신 러닝 시작하기 페이지에서 시작을 선택합니다.



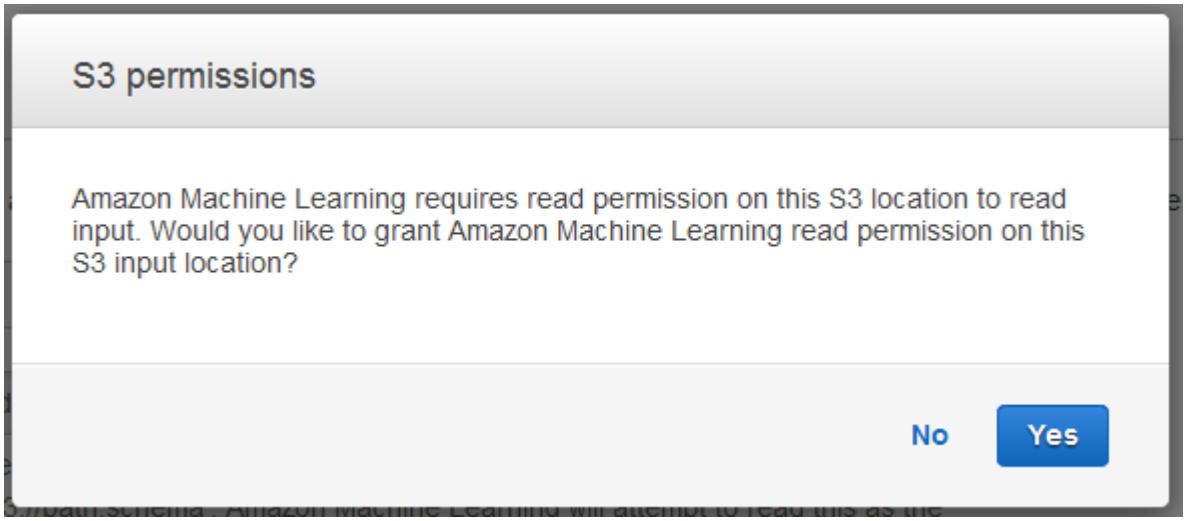
4. 데이터 입력 페이지에서 데이터 위치에 대해 S3가 선택되었는지 확인합니다.



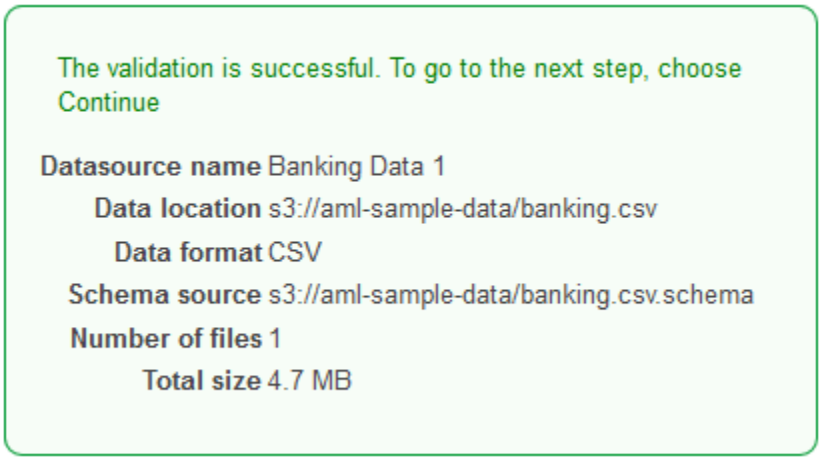
5. S3 위치에 대해 1단계: 데이터 준비의 `banking.csv` 파일 전체 위치를 입력합니다. 예: *`your-bucket`***`/banking.csv`**. Amazon ML이 사용자를 대신하여 버킷 이름 앞에 s3://를 추가합니다.
6. 데이터소스 이름에 대해 **`Banking Data 1`**를 입력합니다.



7. Verify를 선택합니다.
8. S3 권한 대화 상자에서 예를 선택합니다.



9. Amazon ML이 S3 위치의 데이터 파일에 액세스하고 읽을 수 있는 경우 다음과 비슷한 페이지가 표시됩니다. 속성을 검토한 다음 계속을 선택합니다.



그 다음, 스키마를 설정합니다. 스키마는 ML이 속성 이름 및 할당된 데이터 형식, 특수 속성의 이름 등을 포함하여 ML 모델의 입력 데이터를 해석하는 데 필요한 정보입니다. Amazon ML에 스키마를 제공하는 방법은 다음 두 가지가 있습니다.

- Amazon S3 데이터를 업로드할 때 별도의 스키마 파일을 제공합니다.
- Amazon ML이 속성 유형을 유추하고 스키마를 생성하도록 허용합니다.

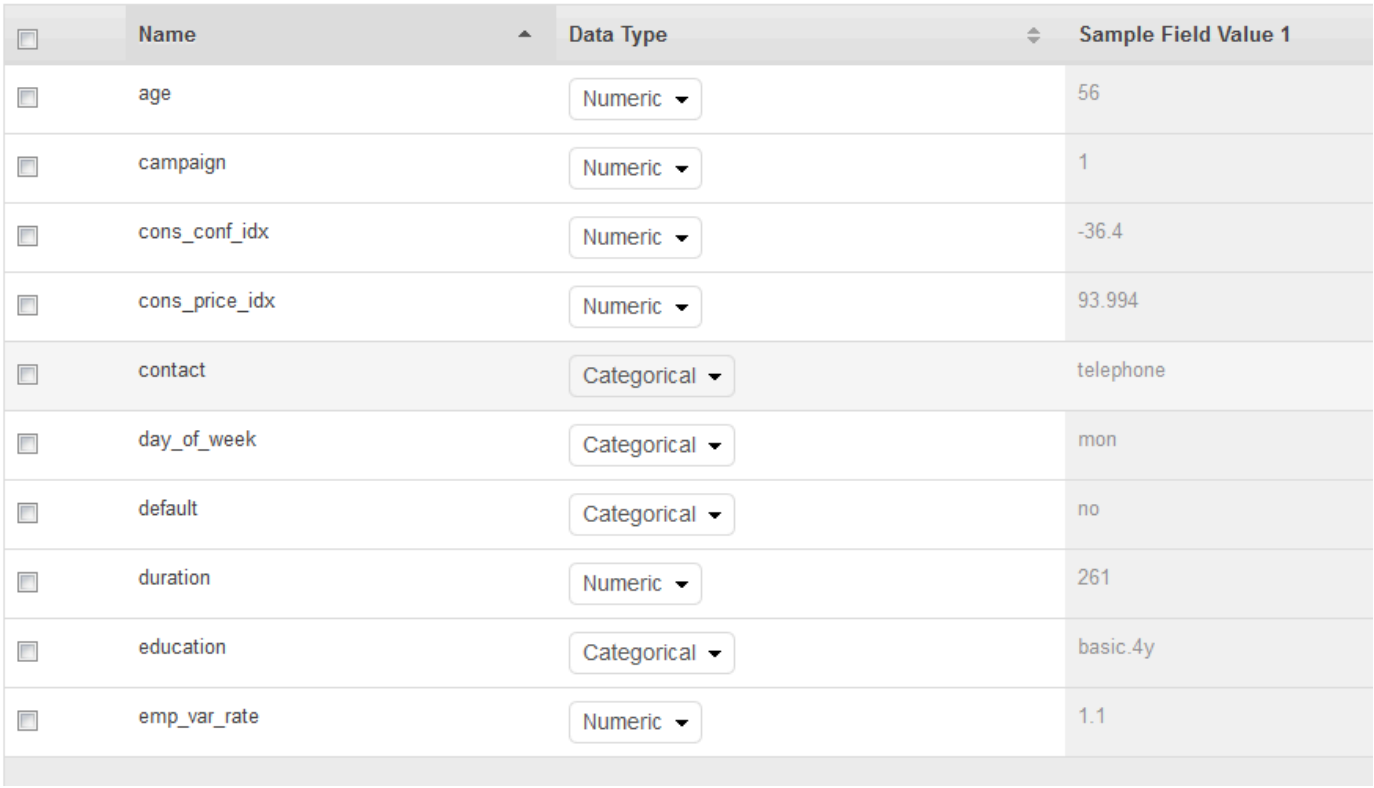
이 자습서에서는 Amazon ML에 스키마를 유추하도록 요청할 것입니다.

별도의 스키마 파일을 생성하는 방법에 대한 자세한 내용은 Amazon ML용 데이터 스키마 생성 단원을 참조하세요.

Amazon ML이 스키마를 유추할 수 있도록 하려면

1. 스키마 페이지에는 ML이 유추한 스키마가 표시됩니다. Amazon ML에서 속성에 대해 유추한 데이터 유형을 검토합니다. Amazon ML이 데이터를 올바르게 수집하고 속성에 대한 올바른 특성 처리를 가능하게 하려면 속성에 올바른 데이터 유형을 할당하는 것이 중요합니다.

   - 예 또는 아니요와 같이 두 가지 상태만 가능한 속성은 이진으로 표시해야 합니다.
   - 범주를 나타내는 데 사용되는 숫자 또는 문자열인 속성은 범주형으로 표시해야 합니다.
   - 순서에 의미가 있는 숫자 수량인 속성은 숫자로 표시해야 합니다.
   - 문자열을 공백으로 구분한 단어로 취급하려는 속성은 텍스트로 표시해야 합니다.



2. 이 자습서에서는 ML이 모든 속성에 대한 데이터 유형을 올바르게 식별했으므로 계속을 선택합니다.

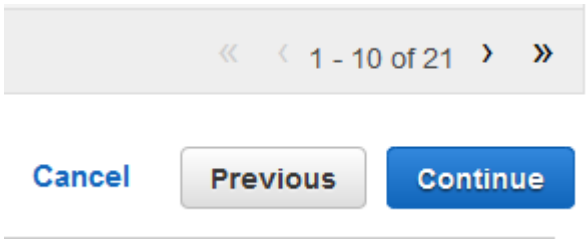
그 다음, 대상 속성을 선택합니다.

대상은 ML 모델이 예측 방법을 학습해야 하는 속성이라는 점을 명심하세요. 속성 y는 개인이 과거에 캠페인을 구독했는지 여부를 나타냅니다(1(예) 또는 0(아니요)).

> **Note**
>
> ML 모델 학습 및 평가에 데이터 소스를 사용할 경우에만 대상 속성을 선택합니다.

y를 대상 속성으로 선택하려면

1. 표 오른쪽 하단에서 단일 화살표를 선택하여 표의 마지막 페이지로 이동합니다. 그러면 y라는 이름이 지정된 속성이 나타납니다.



2. 대상 열에서 y를 선택합니다.

   ML이 y가 대상으로 선택되었음을 확인해줍니다.

3. 계속을 선택합니다.
4. 행 ID 페이지에서 데이터에 식별자가 포함되어 있습니까?에 대해 기본값인 아니요가 선택되어 있는지 확인합니다.
5. 계속, 등록을 차례로 선택합니다.

이제 학습 데이터 소스가 준비되었으므로 모델을 생성할 준비가 되었습니다.

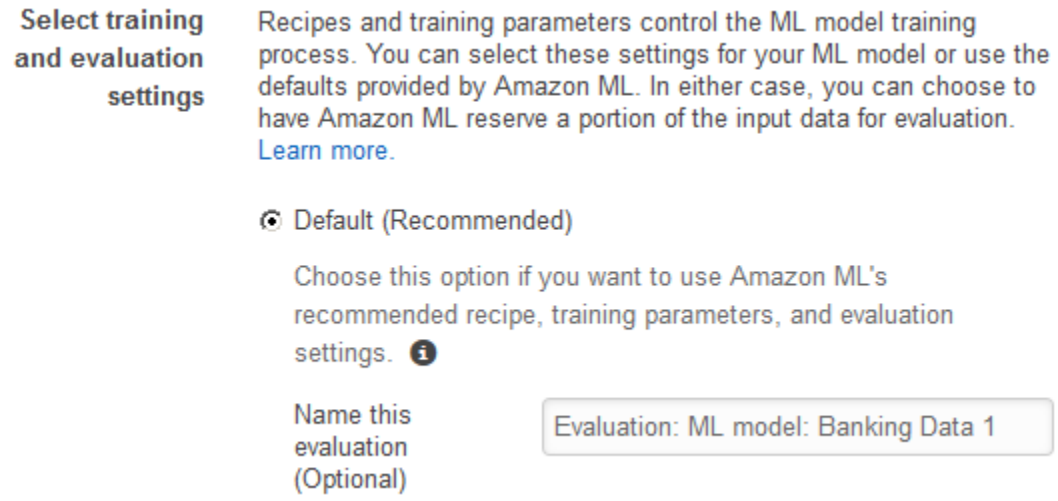
3단계: ML 모델 생성

학습 데이터 소스를 생성한 후 이를 사용하여 ML 모델을 생성하고 모델을 학습시킨 다음 결과를 평가합니다. ML 모델은 Amazon ML이 학습 중에 데이터에서 발견한 패턴 모음입니다. 모델을 사용하여 예측을 생성합니다.

ML 모델을 생성하려면

1. 시작 마법사가 학습 데이터 소스와 모델을 모두 생성해주므로 머신 러닝(ML)은 방금 생성한 학습 데이터 소스를 자동으로 사용하여 ML 모델 설정 페이지로 바로 이동합니다. ML 모델 설정 페이지에서 ML 모델 이름에 대해 기본값인 **`ML model: Banking Data 1`**이 표시되어 있는지 확인합니다.

   기본값과 같이 친숙한 이름을 사용하면 ML 모델을 쉽게 식별하고 관리할 수 있습니다.
2. 학습 및 평가 설정에 대해 기본값이 선택되었는지 확인합니다.



3. 이 평가에 이름 지정에 대해 기본값인 **`Evaluation: ML model: Banking Data 1`**을 그대로 사용합니다.
4. 검토를 선택하고, 설정을 검토한 다음 완료를 선택합니다.

   완료를 선택하면 ML이 모델을 처리 대기열에 추가합니다. Amazon ML은 모델을 생성할 때 기본값을 적용하고 다음 작업을 수행합니다.

   - 학습 데이터 소스를 두 섹션으로 분리합니다. 하나는 데이터의 70%를 포함하고 있고 다른 하나는 나머지 30%를 포함하고 있습니다.
   - 입력 데이터의 70%가 포함된 섹션에서 ML 모델을 학습시킵니다.
   - 입력 데이터의 나머지 30%를 사용하여 모델을 평가합니다.

모델이 대기열에 있는 동안 ML은 상태를 보류 중으로 보고합니다. ML은 모델을 생성하는 동안 상태를 진행 중으로 보고합니다. 모든 작업이 완료되면 상태를 완료됨으로 보고합니다. 평가가 완료될 때까지 기다렸다가 진행합니다.

이제 모델의 성능을 검토하고 커트라인 점수를 설정할 준비가 되었습니다.

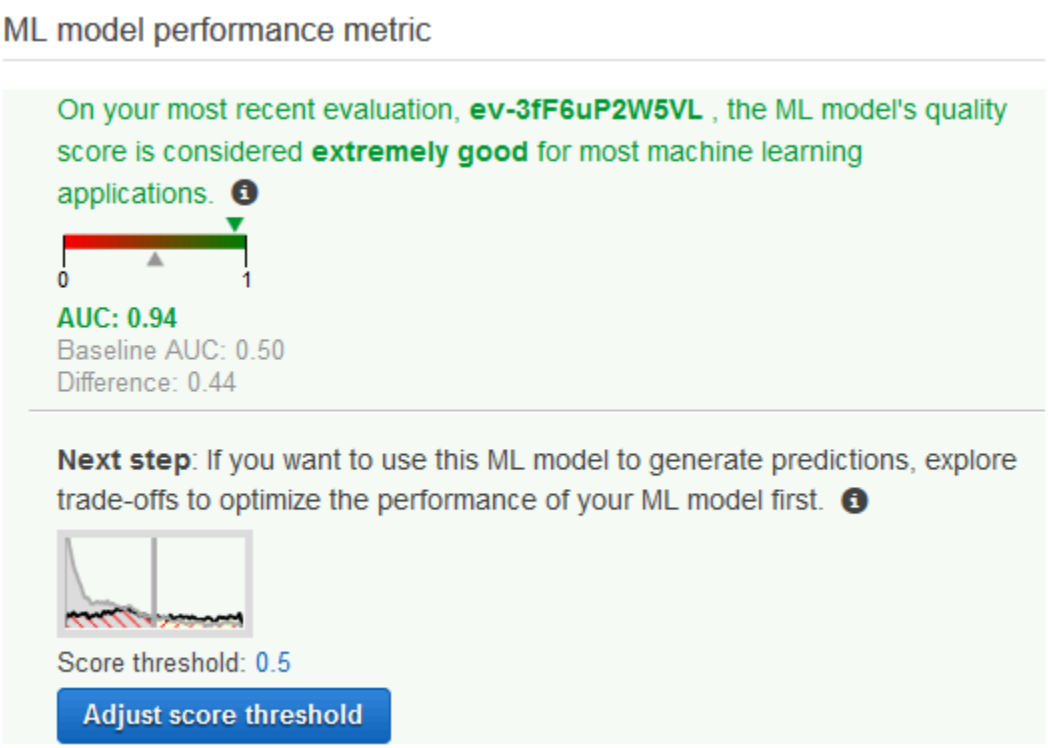
모델 학습 및 평가에 대한 자세한 내용은 모델 학습 및 evaluate an ML model 단원을 참조하세요.

4단계: ML 모델의 예측 성능 검토 및 점수 임계값 설정

ML 모델을 생성하고 Amazon Machine Learning(Amazon ML)에서 이를 평가했으니 이제 사용하기에 충분한지 살펴보겠습니다. 평가 과정에서 Amazon ML은 곡선하면적(AUC) 지표라는 업계 표준 품질 지표를 계산하여 ML 모델의 성능 품질을 나타냅니다. 또한 Amazon ML은 AUC 지표를 해석하여 ML 모델의 품질이 대부분의 기계 학습 애플리케이션에 적합한지 여부도 알려줍니다. (ML 모델 정확도 측정의 AUC에 대해 알아봅니다.) AUC 지표를 검토한 다음 점수 임계값 또는 커트라인을 조정하여 모델의 예측 성능을 최적화해 보겠습니다.

ML 모델의 AUC 지표를 검토하려면

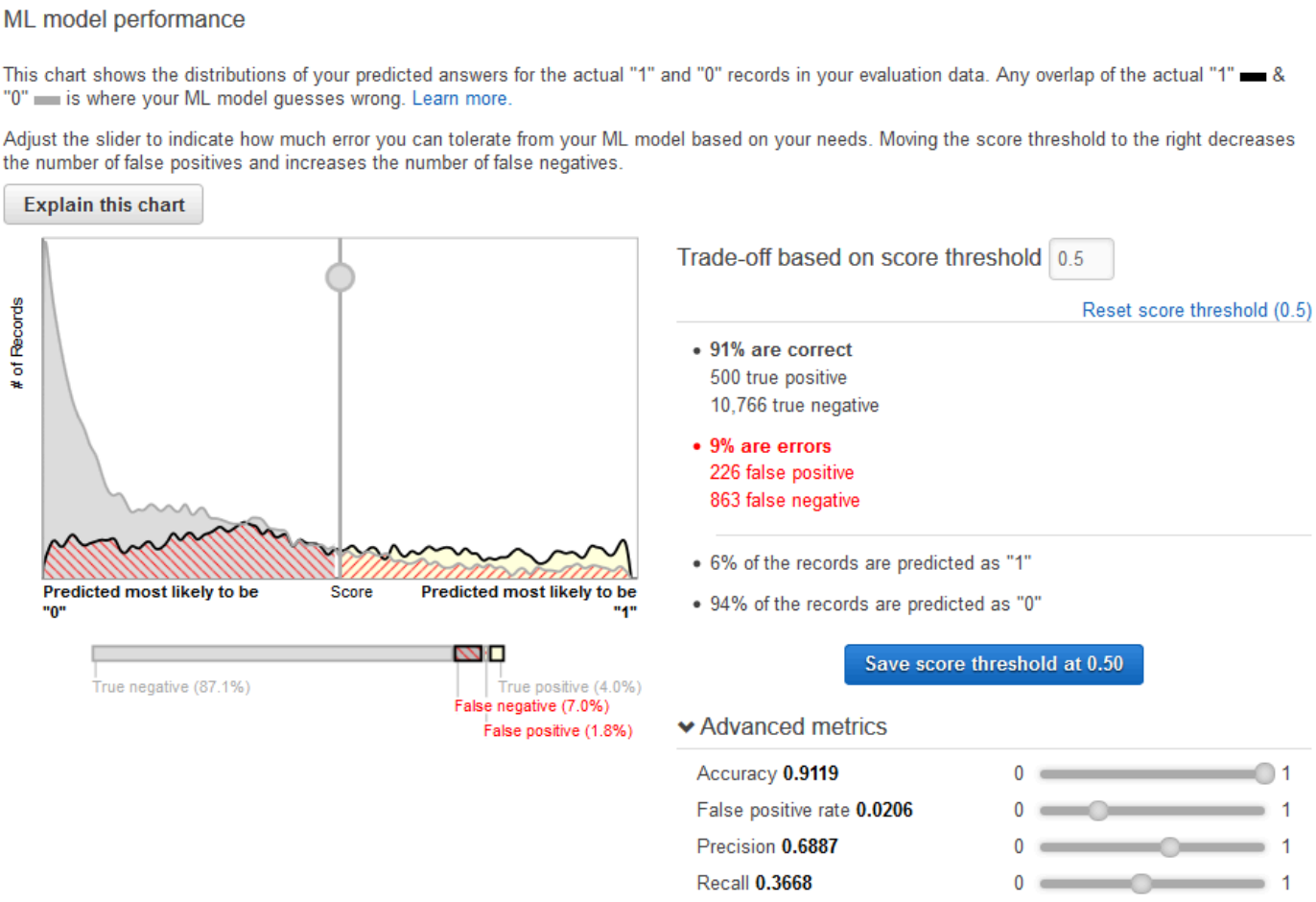
1. ML 모델 요약 페이지의 ML 모델 보고서 탐색 창에서 평가를 선택하고 평가: ML 모델: 은행 모델 1을 선택한 다음 요약을 선택합니다.
2. 평가 요약 페이지에서 모델의 AUC 성능 지표를 포함하여 평가 요약을 검토합니다.



ML 모델은 예측 데이터 소스의 각 레코드에 대한 숫자 예측 점수를 생성한 다음 임계값을 적용하여 이 점수를 0(아니요) 또는 1(예)의 이진 레이블로 변환합니다. 점수 임계값을 변경하여 ML 모델이 이러한 레이블을 할당하는 방식을 조정할 수 있습니다. 이제 점수 임계값을 설정합니다.

ML 모델의 점수 임계값을 설정하려면

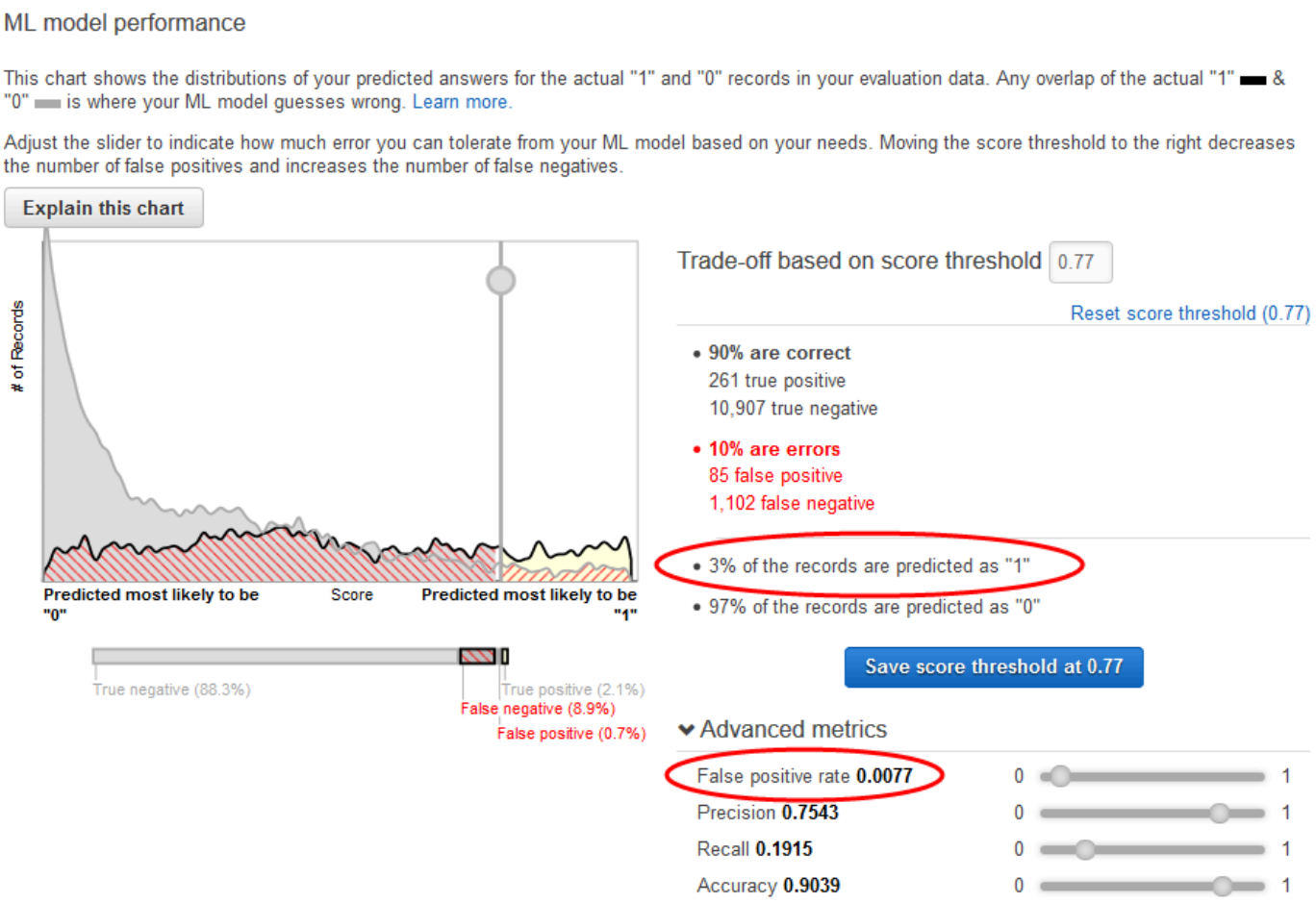
1. 평가 요약 페이지에서 점수 임계값 조정을 선택합니다.



점수 임계값을 조정하면 ML 모델 성능 지표를 세밀하게 조정할 수 있습니다. 이 값을 조정하면 예측이 긍정적으로 간주되기 전에 모델이 예측에 대해 가져야 하는 신뢰 수준이 변경됩니다. 또한 예측에서 허용할 수 있는 거짓 부정과 거짓 긍정의 수도 달라집니다.

참 긍정이 될 가능성이 가장 높은 예측만 긍정으로 간주할 때까지 점수 임계값을 높여 모델이 긍정적 예측으로 간주하는 기준을 제어할 수 있습니다. 더 이상 거짓 부정이 발생하지 않을 때까지 점수 임계값을 낮출 수도 있습니다. 비즈니스 요구 사항을 반영하여 커트라인을 선택하세요. 이 자습서에서는 거짓 긍정이 발생할 때마다 캠페인 비용이 발생하므로 참 긍정 대 거짓 긍정의 비율을 높이고자 합니다.

2. 제품을 구독할 고객 중 상위 3%를 타겟팅하기를 원한다고 가정해 보겠습니다. 세로 선택기를 밀어 점수 임계값을 “1”로 예측되는 레코드의 3%에 해당하는 값으로 설정합니다.



이 점수 임계값이 ML 모델의 성능에 미치는 영향: 거짓 긍정률이 0.007임을 유의하세요. 거짓 긍정 비율이 수용 가능하다고 가정해 보겠습니다.

3. 점수 임계값을 0.77에서 저장을 선택합니다.

이 ML 모델을 사용하여 예측할 때마다 점수가 0.77을 초과하는 레코드는 “1”로, 나머지 레코드는 “0”으로 예측합니다.

점수 임계값에 대한 자세한 내용은 바이너리 분류 단원을 참조하세요.

이제 모델을 사용하여 예측을 만들 준비가 되었습니다.

5단계: ML 모델을 사용하여 예측 생성

Amazon Machine Learning(Amazon ML)은 배치 예측과 실시간 예측이라는 두 가지 유형의 예측을 생성할 수 있습니다.

실시간 예측은 ML이 온디맨드 방식으로 생성하는 단일 관측치에 대한 예측입니다. 실시간 예측은 결과를 대화식으로 사용해야 하는 모바일 앱, 웹 사이트 및 기타 애플리케이션에 적합합니다.

배치 예측은 관측치 그룹에 대한 예측 집합입니다. Amazon ML은 레코드를 배치 예측으로 함께 처리하므로 처리에 다소 시간이 걸릴 수 있습니다. 배치 예측은 관측치 집합에 대한 예측이 필요한 애플리케이션이나 결과를 대화식으로 사용하지 않는 예측이 필요한 애플리케이션에 사용합니다.

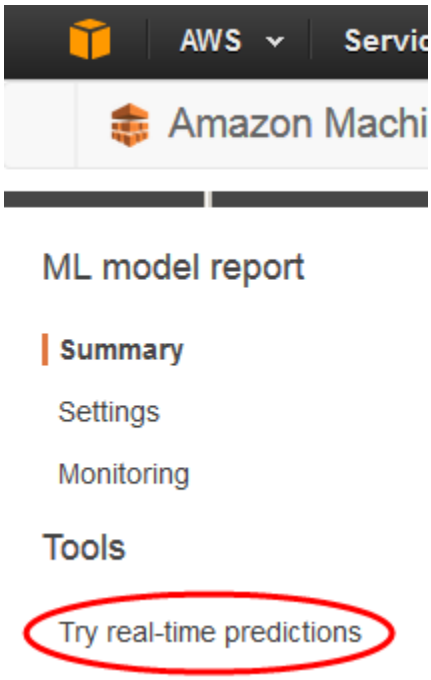
이 자습서에서는 한 명의 잠재 고객이 신제품을 구독할지 여부를 예측하는 실시간 예측을 생성합니다. 또한 대규모 잠재 고객에 대한 예측도 생성할 수 있습니다. 배치 예측의 경우 1단계: 데이터 준비에서 업로드한 `banking-batch.csv` 파일을 사용하게 될 것입니다.

실시간 예측부터 시작해 보겠습니다.

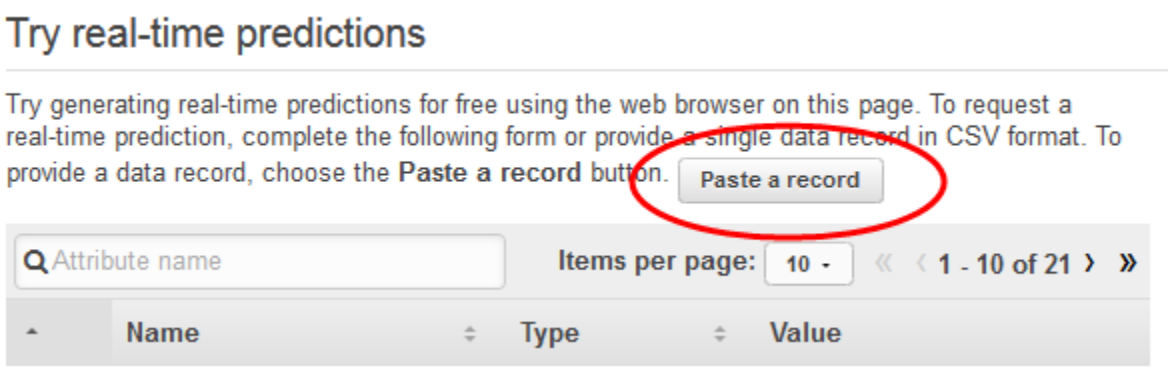
> **Note**
>
> 실시간 예측이 필요한 애플리케이션의 경우 ML 모델을 위한 실시간 엔드포인트를 만들어야 합니다. 실시간 엔드포인트를 사용할 수 있는 동안 요금이 발생합니다. 실시간 예측을 사용하기 전에 실시간 엔드포인트를 만들지 않고도 웹 브라우저의 실시간 예측 기능을 사용해 볼 수 있습니다. 이것이 바로 이 개인 교습에서 우리가 할 일입니다.

실시간 예측을 시도하려면

1. ML 모델 보고서 탐색 창에서 실시간 예측 시도를 선택합니다.



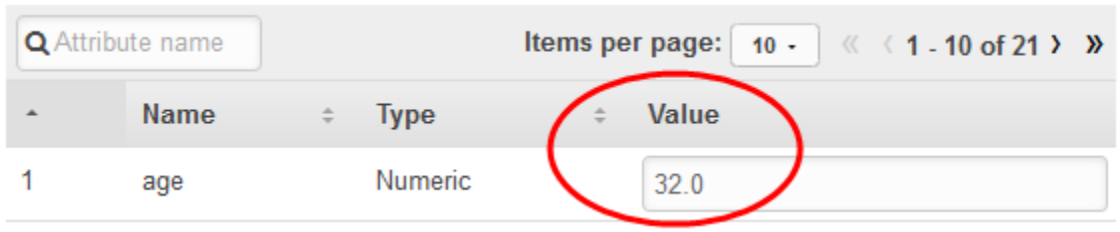
2. 레코드 붙여넣기를 선택합니다.



3. 레코드 붙여넣기 대화 상자에서 다음 관측치를 붙여 넣습니다.

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

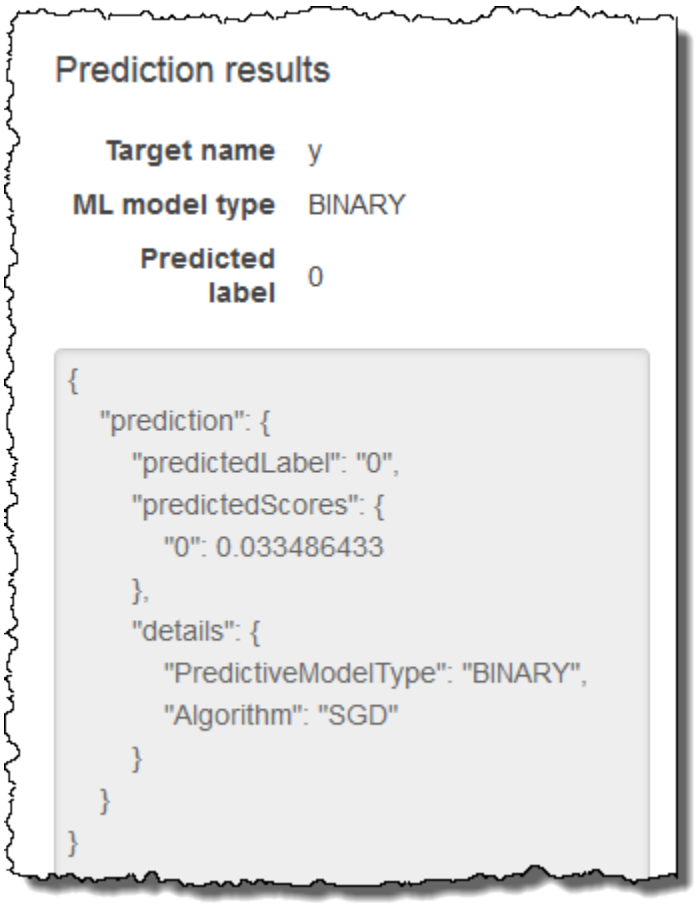
4. 레코드 붙여넣기 대화 상자에서 제출을 선택하여 이 관측치에 대한 예측을 생성할지 확인합니다. Amazon ML이 실시간 예측 양식에 값을 채워줍니다.



> **Note**
>
> 개별 값을 입력하여 값 필드를 채울 수도 있습니다. 어떤 방법을 선택하든 모델 학습에 사용되지 않은 관측치를 제공해야 합니다.

5. 페이지 하단에서 예측 생성을 선택합니다.

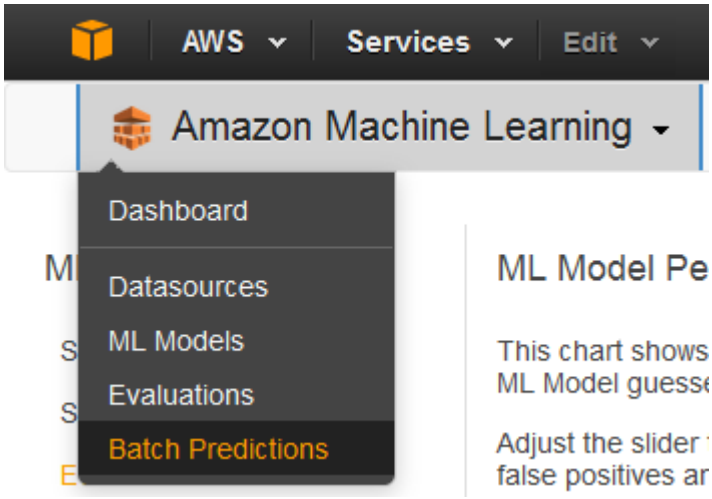
예측이 오른쪽의 예측 결과 창에 표시됩니다. 이 예측에는 `0`의 예측 레이블이 붙어 있는데, 이는 이 잠재 고객이 캠페인에 반응할 가능성이 낮다는 것을 의미합니다. `1`의 예측 레이블은 고객이 캠페인에 반응할 가능성이 높다는 뜻입니다.



이제 배치 예측을 생성합니다. 사용 중인 ML 모델의 이름, 예측을 생성하려는 입력 데이터의 Amazon Simple Storage Service(Amazon S3) 위치(Amazon ML이 이 데이터로부터 배치 예측 데이터 소스를 생성함), 결과를 저장할 Amazon S3 위치를 Amazon ML에 제공하게 될 것입니다.

배치 예측을 생성하려면

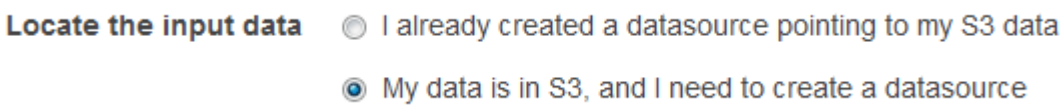
1. 머신 러닝을 선택한 다음 배치 예측을 선택합니다.



2. 새 배치 예측 생성을 선택합니다.
3. 배치 예측용 ML 모델 페이지에서 ML 모델: 은행 데이터 1을 선택합니다.

   Amazon ML이 ML 모델 이름, ID, 생성 시간 및 관련 데이터 소스 ID를 표시합니다.
4. 계속을 선택합니다.
5. 예측을 생성하려면 Amazon ML에 예측이 필요한 데이터를 제공해야 합니다. 이를 입력 데이터라고 합니다. 먼저 Amazon ML에서 액세스할 수 있도록 입력 데이터를 데이터 소스에 넣습니다.

   입력 데이터 찾기에서 내 데이터가 S3에 있고, 데이터 소스를 생성해야 합니다를 선택합니다.

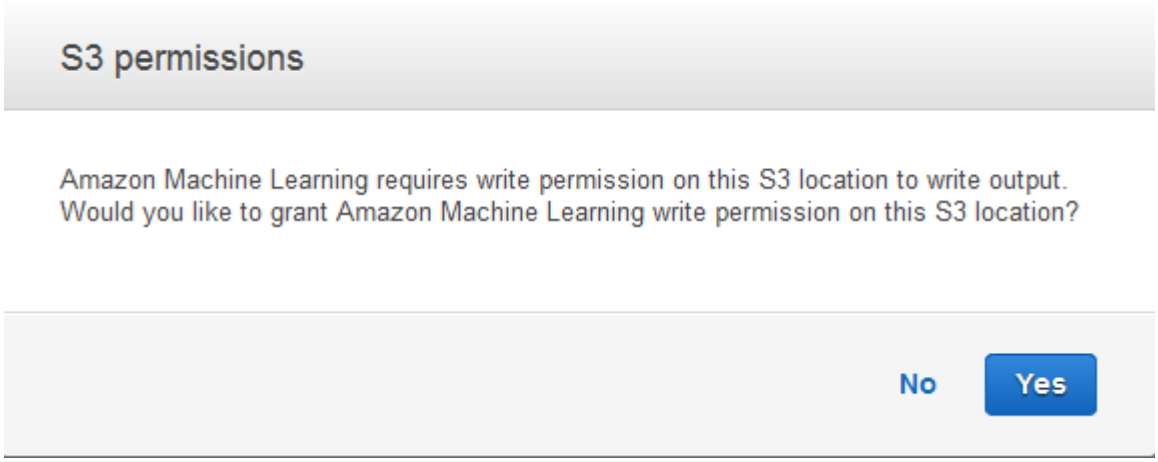
   

6. 데이터 소스 이름에서 **Banking Data 2**를 입력합니다.
7. S3 위치에서 `banking-batch.csv` 파일: *your-bucket*`/banking-batch.csv`의 전체 위치를 입력합니다.
8. CSV의 첫 줄에 열 이름이 들어 있습니까?에 대해, 예를 선택합니다.
9. 확인을 선택합니다.

   Amazon ML이 데이터의 위치를 검증합니다.
10. 계속을 선택합니다.
11. S3 목적지에서 1단계: 데이터 준비에서 파일을 업로드했던 S3 위치의 이름을 입력합니다. Amazon ML이 여기에 예측 결과를 업로드합니다.
12. 배치 예측 이름에 대해 기본값인 **Batch prediction: ML model: Banking Data 1**를 그대로 사용합니다. Amazon ML은 예측 생성에 사용할 모델을 기반으로 기본 이름을 선택합니다. 이

자습서에서는 학습 데이터 소스, `Banking Data 1`의 이름을 따서 모델 및 예측의 이름을 지정합니다.

13. 검토를 선택합니다.
14. S3 권한 대화 상자에서 예를 선택합니다.

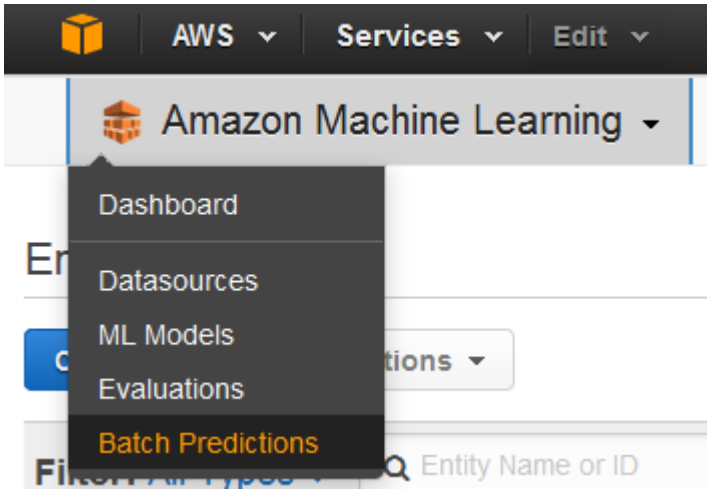


15. 검토 페이지에서 완료를 선택합니다.

배치 예측 요청이 Amazon ML로 전송되고 대기열로 들어갑니다. Amazon ML에서 배치 예측을 처리하는 데 걸리는 시간은 데이터 소스의 크기와 ML 모델의 복잡성에 따라 달라집니다. ML은 요청을 처리하는 동안 상태가 진행 중으로 보고됩니다. 배치 예측이 완료되면 요청 상태가 완료됨으로 변경됩니다. 이제 결과를 확인할 수 있습니다.

예측을 확인하려면

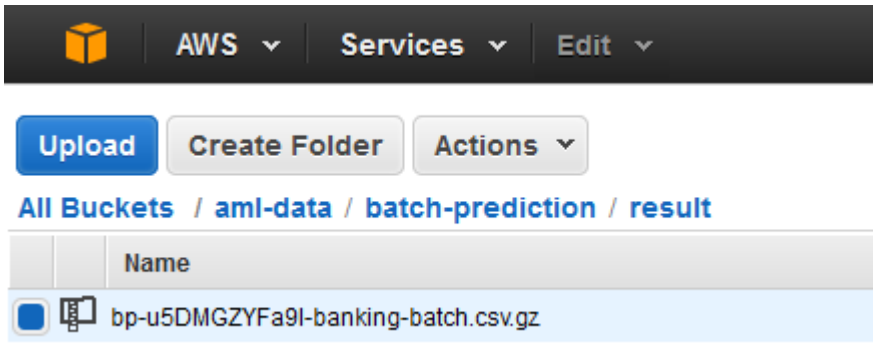
1. 머신 러닝을 선택한 다음 배치 예측을 선택합니다.



2. 예측 목록에서 배치 예측: ML 모델: 은행 데이터 1을 선택합니다. 배치 예측 정보 페이지가 나타납니다.

Name	Subscription propensity Predictions
ID	bp-u5DMGZYFa9l
Creation Time	Mar 5, 2015 3:28:33 PM
Status	Completed
Log	Download Log
Datasource ID	ds-33Rqgz9w3ee
ML Model ID	ml-u7ljoShX2kX
Input S3 URL	s3://aml-data/banking-batch.csv
Output S3 URL	s3://aml-data/

3. 배치 예측 결과를 확인하려면 S3 콘솔(https://console.aws.amazon.com/s3/)로 이동하여 출력 S3 URL 필드에 참조된 S3 위치로 이동합니다. 여기에서 `s3://aml-data/batch-prediction/result`과 비슷한 이름을 가진 결과 폴더로 이동합니다.



예측은 확장자가.gz인 압축된.gzip 파일 형태로 저장됩니다.

4. 예측 파일을 데스크톱으로 다운로드하고 압축을 풀고 엽니다.

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

이 파일에는 BestAnswer와 점수라는 두 개의 열과 데이터 소스의 각 관측치에 대한 행이 있습니다. 최고응답 열의 결과는 4단계: ML 모델의 예측 성능 검토 및 점수 임계값 설정에서 설정했던 점수 임계값인 0.77을 기반으로 합니다. 0.77보다 큰 점수는 긍정 응답 또는 예측에 해당하는 최고응답 1이 되고, 0.77보다 작은 점수는 부정 응답 또는 예측에 해당하는 최고응답 0이 됩니다.

다음 예에서는 점수 임계값인 0.77을 기반으로 한 긍정 예측과 부정 예측을 보여줍니다.

긍정 예측:

bestAnswer	score
1	0.8228876

이 예제에서 최고응답의 값은 1이고 점수의 값은 0.8228876입니다. 점수가 점수 임계값인 0.77보다 크기 때문에 최고응답의 값은 1이 됩니다. 최고응답이 1이면 고객이 제품을 구매할 가능성이 높다는 의미이므로 긍정 예측으로 간주됩니다.

부정 예측:

bestAnswer	score
0	0.7695356

이 예제에서 점수 값이 0.7695356이고 점수 임계값인 0.77보다 작기 때문에 최고응답의 값이 0이 됩니다. 최고응답가 0이면 고객이 제품을 구매할 가능성이 낮다는 의미이므로 부정 예측으로 간주됩니다.

배치 결과의 각 행은 배치 입력(데이터 소스의 관측치)의 행에 해당합니다.

예측을 분석한 후 타겟 마케팅 캠페인을 실행할 수 있습니다. 예를 들어 예측 점수가 1인 전단지를 모든 사람에게 발송할 수 있습니다.

이제 모델을 만들고, 검토하고, 사용했으니, 생성한 데이터와 AWS 리소스를 정리하여 불필요한 비용이 발생하지 않도록 하고 작업 공간을 깔끔하게 유지합니다.

6단계: 정리

Amazon Simple Storage Service(Amazon S3) 요금이 추가로 발생하지 않도록 하려면 Amazon S3에 저장했던 데이터를 삭제합니다. 사용하지 않은 다른 Amazon ML 리소스에 대해서는 요금이 청구되지 않지만, 작업 공간을 깔끔하게 유지하려면 해당 리소스를 삭제하는 것이 좋습니다.

Amazon S3에 저장된 입력 데이터를 삭제하려면

1. https://console.aws.amazon.com/s3/에서 S3 콘솔을 엽니다.

2. banking.csv 및 banking-batch.csv 파일을 저장한 S3 위치로 이동합니다.
3. banking.csv, banking-batch.csv 및 .writePermissionCheck.tmp 파일을 선택합니다.
4. 작업을 선택한 후 삭제를 선택합니다.
5. 확인 메시지가 표시되면 확인을 선택합니다.

Amazon ML에서 실행한 배치 예측이나 자습서 중에 생성한 데이터 소스, 모델 및 평가의 기록을 보관하는 데는 요금이 부과되지 않지만, 작업 공간이 복잡해지지 않도록 삭제해 두는 것이 좋습니다.

배치 예측을 삭제하려면

1. 배치 예측의 출력을 저장한 Amazon S3 위치로 이동합니다.
2. batch-prediction 폴더를 선택합니다.
3. 작업을 선택한 후 삭제를 선택합니다.
4. 확인 메시지가 표시되면 확인을 선택합니다.

Amazon ML 리소스를 삭제하려면

1. Amazon ML 대시보드에서 다음 리소스를 선택합니다.
   - Banking Data 1 데이터 소스
   - Banking Data 1_[percentBegin=0, percentEnd=70, strategy=sequential] 데이터 소스
   - Banking Data 1_[percentBegin=70, percentEnd=100, strategy=sequential] 데이터 소스
   - Banking Data 2 데이터 소스
   - ML model: Banking Data 1 ML 모델
   - Evaluation: ML model: Banking Data 1 평가
2. 작업을 선택한 후 삭제를 선택합니다.
3. 대화 상자에서 삭제를 선택하여 선택한 리소스를 모두 삭제합니다.

이제 자습서를 성공적으로 완료했습니다. 콘솔을 계속 사용하여 데이터 소스, 모델 및 예측을 생성하려면 머신 러닝 개발자 안내서를 참조하세요. API 사용 방법을 알아보려면 머신 러닝 API 참조 단원을 참조하세요.

데이터 소스 생성 및 사용

Amazon ML 데이터 소스를 사용하여 ML 모델을 학습시키고, ML 모델을 평가하고, ML 모델을 사용하여 배치 예측을 생성할 수 있습니다. 데이터 소스 객체에는 입력 데이터에 대한 메타데이터가 포함되어 있습니다. 데이터 소스를 생성하면 Amazon ML은 입력 데이터를 읽고, 해당 속성에 대한 설명 통계를 계산하고, 통계, 스키마 및 기타 정보를 데이터 소스 객체의 일부로 저장합니다. 데이터 소스를 생성한 후 ML 데이터 인사이트를 사용하여 입력 데이터의 통계적 속성을 탐구하고 데이터 소스를 사용하여 ML 모델을 학습시킬 수 있습니다.

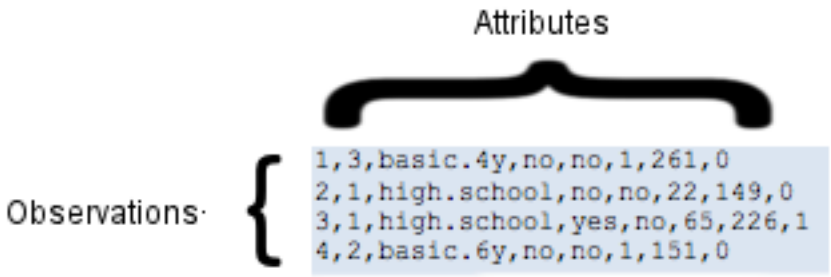
> **Note**
>
> 이 단원에서는 머신 러닝의 개념을 잘 알고 있다고 가정하고 설명합니다.

주제

- Amazon ML의 데이터 형식에 대한 이해
- Amazon ML용 데이터 스키마 생성
- 데이터 분할
- 데이터 인사이트 정보
- Amazon ML에서 Amazon S3 사용
- Amazon Redshift의 데이터에서 Amazon ML 데이터 소스 생성
- Amazon RDS 데이터베이스의 데이터를 사용하여 Amazon ML 데이터 소스 생성

Amazon ML의 데이터 형식에 대한 이해

입력 데이터는 데이터 소스를 생성하는 데 사용하는 데이터입니다. 입력 데이터는 쉼표로 구분된 값(.csv) 형식으로 저장해야 합니다. .csv 파일의 각 행은 단일 데이터 레코드 또는 관측값에 해당합니다. .csv 파일의 각 열에는 관측치의 속성이 들어 있습니다. 예를 들어, 다음 그림은 각각 행에 네 개의 관측치가 있는.csv 파일의 내용을 보여줍니다. 각 관측치에 쉼표로 구분된 8개의 속성이 있습니다. 속성은 관측치(customerId,jobId,education,housing,loan,campaign,duration,willRespondToCampaign)로 표시되는 각 개인에 대한 다음 정보를 나타냅니다.



속성

Amazon ML에는 각 속성에 이름이 필요합니다. 다음과 같이 속성 이름을 지정할 수 있습니다.

- 입력 데이터로 사용하는 .csv 파일의 첫 번째 줄(헤더 라인이라고도 함)에 속성 이름을 포함시켜서
- 입력 데이터와 동일한 S3 버킷에 있는 별도의 스키마 파일에 속성 이름을 포함시켜서

스키마 파일 사용에 대한 자세한 내용은 데이터 스키마 생성 단원을 참조하세요.

.csv 파일의 다음 예제에는 헤더 라인에 속성 이름이 포함되어 있습니다.

```
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
```

입력 파일 형식 요구 사항

입력 데이터가 포함된 .csv 파일은 다음 요구 사항을 충족시켜야 합니다.

- ASCII, 유니코드 또는 EBCDIC과 같은 문자 세트를 사용하여 일반 텍스트로 작성되어야 합니다.
- 한 라인에 관측치 한 개씩, 관측치로 구성되어야 합니다.
- 각 관측치에 대해 속성 값을 쉼표로 구분해야 합니다.
- 속성 값에 쉼표(구분 기호)가 포함된 경우 전체 속성 값을 큰따옴표로 묶어야 합니다.
- 각 관측값은 라인 끝을 나타내는 특수 문자 또는 일련의 문자인 라인 끝 문자로 끝나야 합니다.

- 속성 값을 큰 따옴표로 묶더라도 속성 값에 라인 끝 문자를 포함할 수 없습니다.
- 모든 관측치는 동일한 수의 속성과 속성 순서를 가져야 합니다.
- 각 관측치는 100KB를 넘지 않아야 합니다. Amazon ML은 처리 중에 100KB를 초과하는 관측치를 모두 거부합니다. Amazon ML이 10,000개 이상의 관측치를 거부하게 될 경우 전체 .csv 파일을 거부합니다.

여러 파일을 Amazon ML에 데이터 입력으로 사용

Amazon ML에 입력 내용을 단일 파일 또는 파일 모음으로 제공할 수 있습니다. 파일 모음은 다음 조건을 충족시켜야 합니다.

- 모든 파일에 동일한 데이터 스키마가 있어야 합니다.
- 모든 파일에 동일한 Amazon Simple Storage Service(Amazon S3) 접두사에 있어야 합니다. 파일 모음에 제공하는 경로는 슬래시('/') 문자로 끝나야 합니다.

예를 들어 데이터 파일 이름이 input1.csv, input2.csv, input3.csv이고 S3 버킷 이름이 s3://examplebucket 인 경우 파일 경로는 다음과 같을 수 있습니다.

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

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

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

Amazon ML에 대한 입력으로 다음과 같은 S3 위치를 제공하면 됩니다.

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

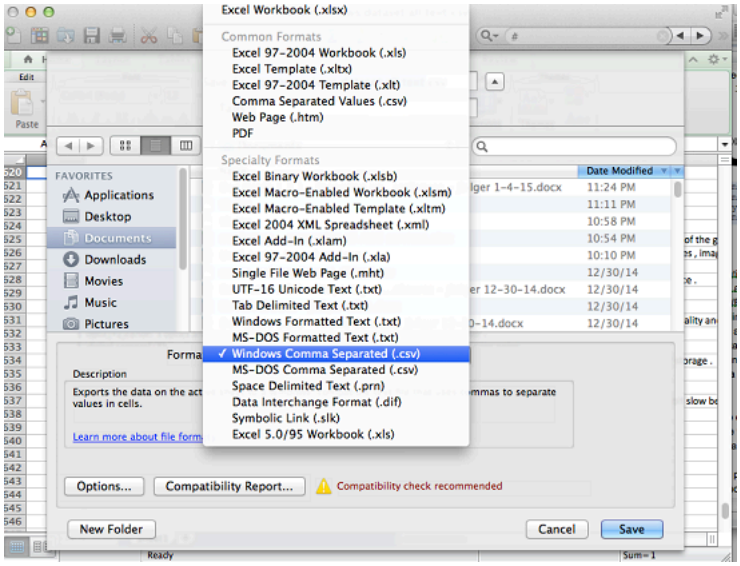
CSV 형식의 라인 끝 문자

.csv 파일을 만들면 각 관측치는 특수 라인 끝 문자로 종료됩니다. 이 문자는 보이지 않지만 Enter 또는 Return 키를 누르면 각 관측치 끝에 자동으로 포함됩니다. 라인 끝을 나타내는 특수 문자는 운영 체제에 따라 다릅니다. Linux 또는 OS X와 같은 Unix 시스템에서는 “\n”(10진수는 ASCII 코드 10 또는 16진수의 경우 0x0a)으로 표시되는 줄 바꿈 문자를 사용합니다. Microsoft Windows에서는 “\r\n”(ASCII 코드 십진수 13과 10, 16진수의 경우 0x0d 및 0x0a)으로 표시되는 캐리지 리턴과 줄 바꿈이라는 두 가지 문자를 사용합니다.

OS X와 Microsoft Excel을 사용하여.csv 파일을 만들려면 다음 절차를 수행합니다. 올바른 형식을 선택했는지 확인합니다.

OS X 및 Excel을 사용하는 경우 .csv 파일을 저장하려면

1. .csv 파일을 저장할 때는 형식을 선택한 다음 Windows 쉼표로 분리(.csv)를 선택합니다.
2. 저장을 선택합니다.



> **Important**
>
> ML에서 읽을 수 없으므로 쉼표로 구분된 값(.csv) 또는 MS-DOS 쉼표로 구분된 값(.csv) 형식을 사용하여.csv 파일을 저장하지 마세요.

Amazon ML용 데이터 스키마 생성

스키마는 입력 데이터의 모든 속성과 해당 데이터 유형으로 구성됩니다. Amazon ML은 이를 통해 데이터 소스의 데이터를 이해할 수 있습니다. Amazon ML은 스키마의 정보를 사용하여 입력 데이터를 읽고 해석하고, 통계를 계산하고, 올바른 속성 변환을 적용하고, 학습 알고리즘을 미세 조정합니다. 스키마를 제공하지 않으면 Amazon ML은 데이터에서 스키마를 추론합니다.

스키마 예제

Amazon ML이 입력 데이터를 올바르게 읽고 정확한 예측을 생성하려면 각 속성에 올바른 데이터 유형을 할당해야 합니다. 예제를 통해 데이터 유형이 속성에 할당되는 방식과 속성과 데이터 유형이 스키마에 포함되는 방식을 살펴보겠습니다. 이메일 캠페인에 응답할 고객을 예측하기 위해 이 예제를 "고객 캠페인"이라고 부르겠습니다. 입력 파일은 9개의 열로 구성된 .csv 파일입니다.

```
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
```

이 데이터의 스키마는 다음과 같습니다.

```
{
    "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"
        }

    ]
}
```

이 예제의 스키마 파일에서 rowId의 값은 customerId입니다.

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

속성 willRespondToCampaign은 대상 속성으로 정의됩니다.

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

customerId 속성 및 CATEGORICAL 데이터 유형은 첫 번째 열과 연결되고, jobId 속성 및 CATEGORICAL 데이터 유형은 두 번째 열과 연결되고, jobDescription 속성 및 TEXT 데이터 유형은 세 번째 열과 연결되고, education 속성 및 CATEGORICAL 데이터 유형은 네 번째 열과 연결되는 식입니다. 아홉 번째 열은 BINARY 데이터 유형을 가진 willRespondToCampaign 속성과 연결되며 이 속성도 대상 속성으로 정의됩니다.

targetAttributeName 필드 사용

targetAttributeName 값은 예측하려는 속성의 이름입니다. 모델을 생성하거나 평가할 때 targetAttributeName를 할당해야 합니다.

ML 모델을 학습하거나 평가할 때 targetAttributeName는 입력 데이터에서 대상 속성에 대한 "정답"을 포함하고 있는 속성의 이름을 식별합니다. Amazon ML은 정답이 포함된 대상을 이용하여 패턴을 찾아내고 ML 모델을 생성합니다.

모델을 평가할 때 Amazon ML은 대상을 이용하여 예측의 정확성을 확인합니다. ML 모델을 생성하고 평가한 후에는 할당되지 않은 `targetAttributeName`를 이용하여 ML 모델로 예측을 생성할 수 있습니다.

대상 속성은 데이터 소스를 생성할 때 Amazon ML 콘솔이나 스키마 파일에서 정의합니다. 자체 스키마 파일을 생성하는 경우 다음 구문을 사용하여 대상 속성을 정의합니다.

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

이 예제에서 `exampleAttributeTarget`는 입력 파일에 있는 대상 속성의 이름입니다.

rowID 필드 사용

`row ID`는 입력 데이터의 속성과 관련된 선택적 플래그입니다. 지정된 경우 `row ID`로 표시된 속성이 예측 출력에 포함됩니다. 이 속성을 이용하면 어떤 예측이 어떤 관측치에 대응하는지 쉽게 연결할 수 있습니다. 양호한 `row ID`의 예로는 고객 ID 또는 이와 유사한 고유 속성이 있습니다.

> **Note**
>
> 행 ID는 참조용으로만 사용됩니다. Amazon ML은 ML 모델을 학습할 때 이를 사용하지 않습니다. 속성을 행 ID로 선택하면 ML 모델 학습에 사용되지 않습니다.

데이터 소스를 생성할 때 ML 콘솔이나 스키마 파일에서 `row ID`을 정의합니다. 자체 스키마 파일을 생성하는 경우 다음 구문을 사용하여 `row ID`을 정의합니다.

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

위 예제에서 `exampleRow`는 행 ID로 정의된 입력 파일의 속성 이름입니다.

배치 예측을 생성할 때 출력은 다음과 같습니다.

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

이 예제에서 RowID는 속성 `customerId`를 나타냅니다. 예를 들어 customerId 55는 이메일 캠페인에 대해 낮은 신뢰도(0.46317)로 응답할 것으로 예측되는 반면, `customerId` 102은 이메일 캠페인에 대해 높은 신뢰도(0.89625)로 응답할 것으로 예측됩니다.

AttributeType 필드 사용

Amazon ML에는 속성에 대한 다음 4가지 데이터 유형이 있습니다.

이진

가능한 상태가 두 개(예: yes 또는 no)만 있는 속성의 경우 BINARY을 선택합니다.

예를 들어, 속성 isNew는 개인이 신규 고객인지 여부를 추적하기 위한 것으로, 개인이 신규 고객임을 나타내는 true 값과, 신규 고객이 아님을 나타내는 false 값이 있습니다.

유효한 음수 값은 0, n, no, f 및 false입니다.

유효한 양수 값은 1, y, yes, t 및 true입니다.

Amazon ML은 이진 입력의 대소문자를 무시하고 앞뒤 공백을 제거합니다. 예를 들어 " FaLSe "는 유효한 이진수 값입니다. 같은 데이터 소스에서 사용하는 이진수 값을 혼합하여 사용할 수 있습니다(예: true, no 및 1 사용). ML은 이진 속성에 대해 0 및 1만 출력합니다.

범주형

제한된 수의 고유 문자열 값을 가진 속성의 경우 CATEGORICAL을 선택합니다. 예를 들어 사용자 ID, 월, 우편번호가 범주형 값에 해당합니다. 범주형 속성은 단일 문자열로 처리되며 더 이상 토큰화되지 않습니다.

숫자

수량을 값으로 취하는 속성의 경우 NUMERIC을 선택합니다.

예를 들어 온도, 무게, 클릭률이 숫자 값에 해당합니다.

숫자를 포함하는 모든 속성이 숫자인 것은 아닙니다. 달의 어떤 날짜 및 ID와 같은 범주형 속성도 종종 숫자로 표시됩니다. 숫자로 간주되려면 숫자가 다른 숫자와 비슷해야 합니다. 예를 들어 고객 ID 664727은 고객 ID124552 에 대해 아무 것도 알려주지 않지만 가중치가 10인 경우 해당 속성이 가중치가 5인 속성보다 무겁다는 것을 나타냅니다. 달의 어떤 날짜는 어떤 달의 첫 날이 다른 달의 두 번째 날보다 앞이나 뒤에 올 수 있기 때문에 숫자에 해당되지 않습니다.

> **Note**
>
> ML을 사용하여 스키마를 생성하면 숫자를 사용하는 모든 속성에 Numeric 데이터 유형이 할당됩니다. ML에서 스키마를 생성하는 경우 잘못된 할당이 있는지 확인하고 해당 속성을 CATEGORICAL로 설정합니다.

텍스트

단어 문자열에 해당하는 속성의 경우 TEXT을 선택합니다. 텍스트 속성을 읽을 때 Amazon ML은 이들 속성을 공백으로 구분된 토큰으로 변환합니다.

예를 들어, `email subject`은 `email` 및 `subject`이 되고, `email-subject here`는 `email-subject` 및 `here`이 됩니다.

학습 스키마의 변수에 대한 데이터 유형이 평가 스키마의 해당 변수에 대한 데이터 유형과 일치하지 않는 경우 Amazon ML은 교육 데이터 유형과 일치하도록 평가 데이터 유형을 변경합니다. 예를 들어 학습 데이터 스키마는 변수 `age`에 대해 `TEXT`의 데이터 유형을 할당하지만 평가 스키마는 `age`에 대해 `NUMERIC`의 데이터 유형을 할당하는 경우 ML은 평가 데이터의 연령을 `TEXT` 변수 대신 `NUMERIC` 변수로 취급합니다.

각 데이터 유형과 관련된 통계에 대한 자세한 내용은 설명 통계 단원을 참조하세요.

Amazon ML에 스키마 제공

모든 데이터 소스에는 스키마가 필요합니다. 다음 두 가지 방법 중에서 선택하여 Amazon ML에 스키마를 제공할 수 있습니다.

- Amazon ML은 입력 데이터 파일에 있는 각 속성의 데이터 유형을 추론하고 자동으로 스키마를 생성하도록 허용합니다.
- Amazon Simple Storage Service(Amazon S3) 데이터를 업로드할 때 스키마 파일을 제공합니다.

Amazon ML이 스키마를 생성하도록 허용

Amazon ML 콘솔을 사용하여 데이터 소스를 생성할 때 Amazon ML은 변수 값을 기반으로 하는 간단한 규칙을 사용하여 스키마를 생성합니다. Amazon ML에서 생성한 스키마를 검토하고 정확하지 않은 경우 데이터 유형을 수정하는 것이 좋습니다.

스키마 제공

스키마 파일을 생성한 후에는 Amazon ML에서 사용할 수 있도록 설정해야 합니다. 여기에는 두 가지 옵션이 있습니다.

1. Amazon ML 콘솔을 사용하여 스키마를 제공합니다.

   콘솔을 사용하여 데이터 소스를 생성하고, 입력 데이터 파일의 파일 이름에 .schema 확장자를 추가하여 스키마 파일을 포함시킵니다. 예를 들어 입력 데이터에 대한 Amazon Simple Storage

Service(Amazon S3) URI가 s3://my-bucket-name/data/input.csv 인 경우 스키마의 URI는 s3://my-bucket-name/data/input.csv.schema 가 됩니다. Amazon ML은 데이터에서 스키마를 유추하려고 하지 않고 사용자가 제공한 스키마 파일을 자동으로 찾습니다.

파일 디렉터리를 Amazon ML에 대한 데이터 입력으로 사용하려면 디렉터리 경로에 .schema 확장자를 추가합니다. 예를 들어 데이터 파일이 s3://examplebucket/path/to/data/ 위치에 있는 경우 스키마의 URI는 s3://examplebucket/path/to/data/.schema 가 됩니다.

2. Amazon ML API를 사용하여 스키마를 제공합니다.

   ML API를 호출하여 데이터 소스를 생성하려는 경우, S3에 스키마 파일을 업로드한 다음 `CreateDataSourceFromS3` API의 `DataSchemaLocationS3` 속성에서 해당 파일에 URI를 제공할 수 있습니다. 자세한 내용은 CreateDataSourceFromS3를 참조하세요.

   스키마를 S3에 먼저 저장하는 대신 `CreateDataSource*` APIs 페이로드에 직접 제공해도 됩니다. 이렇게 하려면 전체 스키마 문자열을 `CreateDataSourceFromS3`, `CreateDataSourceFromRDS` 또는 `CreateDataSourceFromRedshift` API의 `DataSchema` 속성에 배치합니다. 자세한 내용은 머신 러닝 API 참조 단원을 참조하세요.

데이터 분할

ML 모델의 기본 목표는 모델 학습에 사용되는 데이터 외에도 미래의 데이터 인스턴스에 대해 정확한 예측을 하는 것입니다. ML 모델을 사용하여 예측하기 전에 먼저 모델의 예측 성능을 평가해야 합니다. 아직 확인하지 못한 데이터로 ML 모델 예측의 품질을 추정하기 위해, 이미 답을 알고 있는 데이터의 일부를 미래 데이터에 대한 프록시로 예약하거나 분할하여 ML 모델이 해당 데이터에 대한 정답을 얼마나 잘 예측하는지 평가할 수 있습니다. 데이터 소스를 학습 데이터 소스의 일부와 평가 데이터 소스의 일부로 분할합니다.

Amazon ML에서는 다음 세 가지 데이터 분할 옵션을 제공합니다.

- 데이터 사전 분할 - 데이터를 두 개의 데이터 입력 위치로 분할한 다음, 데이터를 Simple Storage Service(S3)에 업로드하고 이를 통해 두 개의 개별 데이터 소스를 생성할 수 있습니다.
- ML 순차 분할 - 학습 및 평가 데이터 소스를 생성할 때 ML에 데이터를 순차적으로 분할하도록 지시할 수 있습니다.
- ML 임의 분할 - 학습 및 평가 데이터 소스를 생성할 때 ML에 초기 설정된 무작위 방법을 사용하여 데이터를 분할하도록 지시할 수 있습니다.

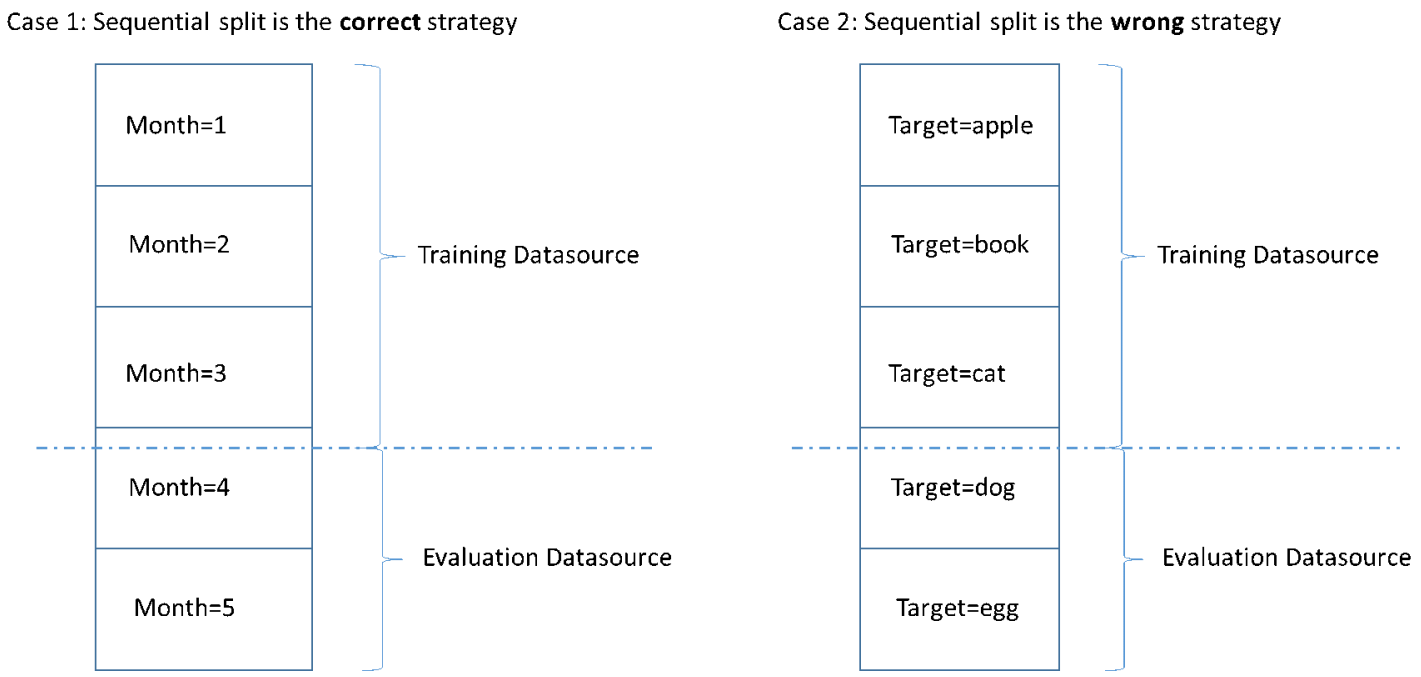
데이터 사전 분할

학습 및 평가 데이터 소스의 데이터를 명시적으로 제어하려면 데이터를 별도의 데이터 위치로 분할하고 입력 위치와 평가 위치에 대해 별도의 데이터 소스를 만듭니다.

데이터 순차적 분할

학습 및 평가를 위해 입력 데이터를 분할하는 간단한 방법은 데이터 레코드의 순서를 유지하면서 중복되지 않는 데이터 하위 집합을 선택하는 것입니다. 이 접근 방식은 특정 날짜 또는 특정 시간 범위 내의 데이터를 기반으로 ML 모델을 평가하려는 경우에 유용합니다. 예를 들어 지난 5개월 동안의 고객 참여 데이터가 있는데 이 과거 데이터를 사용하여 다음 달의 고객 참여를 예측한다고 가정해 보겠습니다. 전체 데이터 범위에서 추출한 레코드 데이터를 사용하는 것보다 범위의 시작 부분은 학습에 사용하고, 범위의 끝 부분에서 나온 데이터는 평가에 사용한다면 모델 품질을 더 정확하게 추정할 수 있습니다.

다음 그림은 순차 분할 전략을 사용해야 하는 경우와 무작위 전략을 사용해야 하는 경우의 예를 보여줍니다.

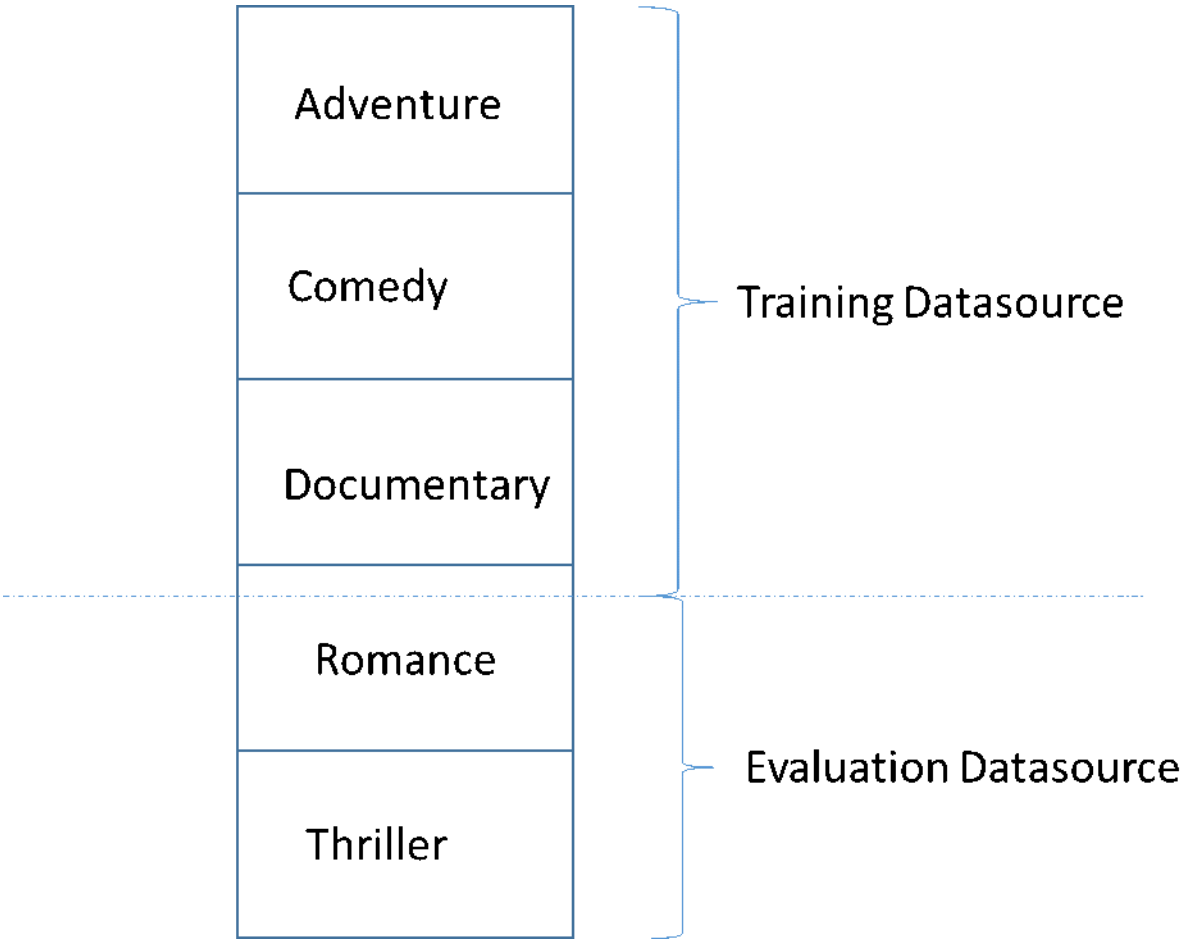


데이터 소스를 생성할 때 데이터 소스를 순차적으로 분할하도록 선택할 수 있으며, Amazon ML은 데이터의 처음 70%는 학습에 사용하고 나머지 30%는 평가에 사용합니다. 이것이 Amazon ML 콘솔을 사용하여 데이터를 분할할 때의 기본 접근 방식입니다.

데이터 무작위 분할

입력 데이터를 학습 및 평가 데이터 소스로 무작위로 분할하면 데이터가 학습 데이터 소스와 평가 데이터 소스에서 비슷하게 분배될 수 있습니다. 입력 데이터의 순서를 유지할 필요가 없는 경우 이 옵션을 선택하세요.

Amazon ML은 시드된 유사 난수 생성 방법을 사용하여 데이터를 분할합니다. 시드는 일부는 입력 문자열 값을, 일부는 데이터 자체의 내용을 각각 기반으로 합니다. Amazon ML 콘솔은 기본적으로 입력 데이터의 S3 위치를 문자열로 사용합니다. API 사용자는 사용자 지정 문자열을 제공할 수 있습니다. 즉, 동일한 S3 버킷과 데이터가 주어지면 Amazon ML은 매번 같은 방식으로 데이터를 분할합니다. ML이 데이터를 분할하는 방식을 변경하려면 `CreateDatasourceFromS3`, `CreateDatasourceFromRedshift` 또는 `CreateDatasourceFromRDS` API를 사용하고 시드 문자열에 값을 제공하면 됩니다. 이러한 API를 사용하여 학습 및 평가를 위한 별도의 데이터 소스를 생성할 때는 학습 데이터와 평가 데이터 간에 중복이 없도록 두 데이터 소스에 동일한 시드 문자열 값을 사용하고 한 데이터 소스에는 보완 플래그를 사용하는 것이 중요합니다.



고품질 ML 모델을 개발할 때 흔히 저지르는 실수 중 하나가 학습에 사용되는 데이터와 유사하지 않은 데이터를 기반으로 ML 모델을 평가하는 것입니다. 예를 들어 ML을 사용하여 영화 장르를 예측하고 있

는데 학습 데이터에 어드벤처, 코미디, 다큐멘터리 장르의 영화가 포함되어 있다고 가정해 보겠습니다. 하지만 평가 데이터에는 로맨스와 스릴러 장르의 데이터만 포함되어 있습니다. 이 경우 ML 모델이 로맨스 장르와 스릴러 장르에 대한 정보를 전혀 학습하지 않았으며, 어드벤처, 코미디, 다큐멘터리 장르의 패턴을 모델이 얼마나 잘 학습했는지 평가하지 않았습니다. 따라서 장르 정보는 쓸모가 없으며 모든 장르에 대한 ML 모델 예측의 품질이 저하됩니다. 모델과 평가가 너무 달라지게 되어(설명 통계가 매우 다름) 유용하지 않게 됩니다. 입력 데이터를 데이터 세트의 열 중 하나를 기준으로 정렬한 다음 순차적으로 분할하면 이렇게 될 수 있습니다.

학습 데이터 소스와 평가 데이터 소스의 데이터 분포가 서로 다른 경우 모델 평가 시 평가 경보가 표시됩니다. 평가 경보에 대한 자세한 내용은 평가 경보 단원을 참조하세요.

S3에서 입력 데이터를 무작위로 셔플링하거나 데이터 소스를 생성할 때 Redshift SQL 쿼리의 `random()` 함수 또는 MySQL 쿼리의 `rand()` 함수를 사용하는 등 이미 입력 데이터를 무작위로 구성한 경우 ML에서 임의 분할을 사용할 필요가 없습니다. 이러한 경우 순차 분할 옵션을 사용하여 유사한 분포의 학습 및 평가 데이터 소스를 생성할 수 있습니다.

데이터 인사이트 정보

Amazon ML은 입력 데이터에 대한 설명 통계를 계산하여 데이터를 이해하는 데 사용할 수 있습니다.

설명 통계

Amazon ML은 다양한 속성 유형에 대해 다음과 같은 설명 통계를 계산합니다.

숫자

- 분포 히스토그램
- 유효하지 않은 값 수
- 최소값, 중앙값, 평균값, 최대값

이진 및 범주형:

- 카운트(범주별 고유 값)
- 가치 분포 히스토그램
- 가장 빈번한 값
- 고유 값 카운트
- 실제 값의 백분율(이진만 해당)
- 가장 눈에 띄는 단어

- 가장 자주 사용하는 단어

텍스트

- 속성의 이름
- 대상과의 상관 관계(대상이 설정된 경우)
- 총 단어 수
- 고유 단어
- 한 줄의 단어 수 범위
- 단어 길이 범위
- 가장 눈에 띄는 단어

Amazon ML 콘솔에서 데이터 인사이트 정보에 액세스

ML 콘솔에서는 데이터 소스의 이름 또는 ID를 선택하면 데이터 인사이트 정보 페이지를 볼 수 있습니다. 이 페이지는 다음 정보를 포함하여 데이터 소스와 관련된 입력 데이터에 대해 알아볼 수 있는 지표와 시각화를 제공합니다.

- 데이터 요약
- 대상 분포
- 누락 값
- 유효하지 않은 값
- 데이터 유형별 변수 요약 통계
- 데이터 유형별 변수 분포

다음에 이어지는 단원에서는 지표와 시각화에 대해 보다 자세히 설명합니다.

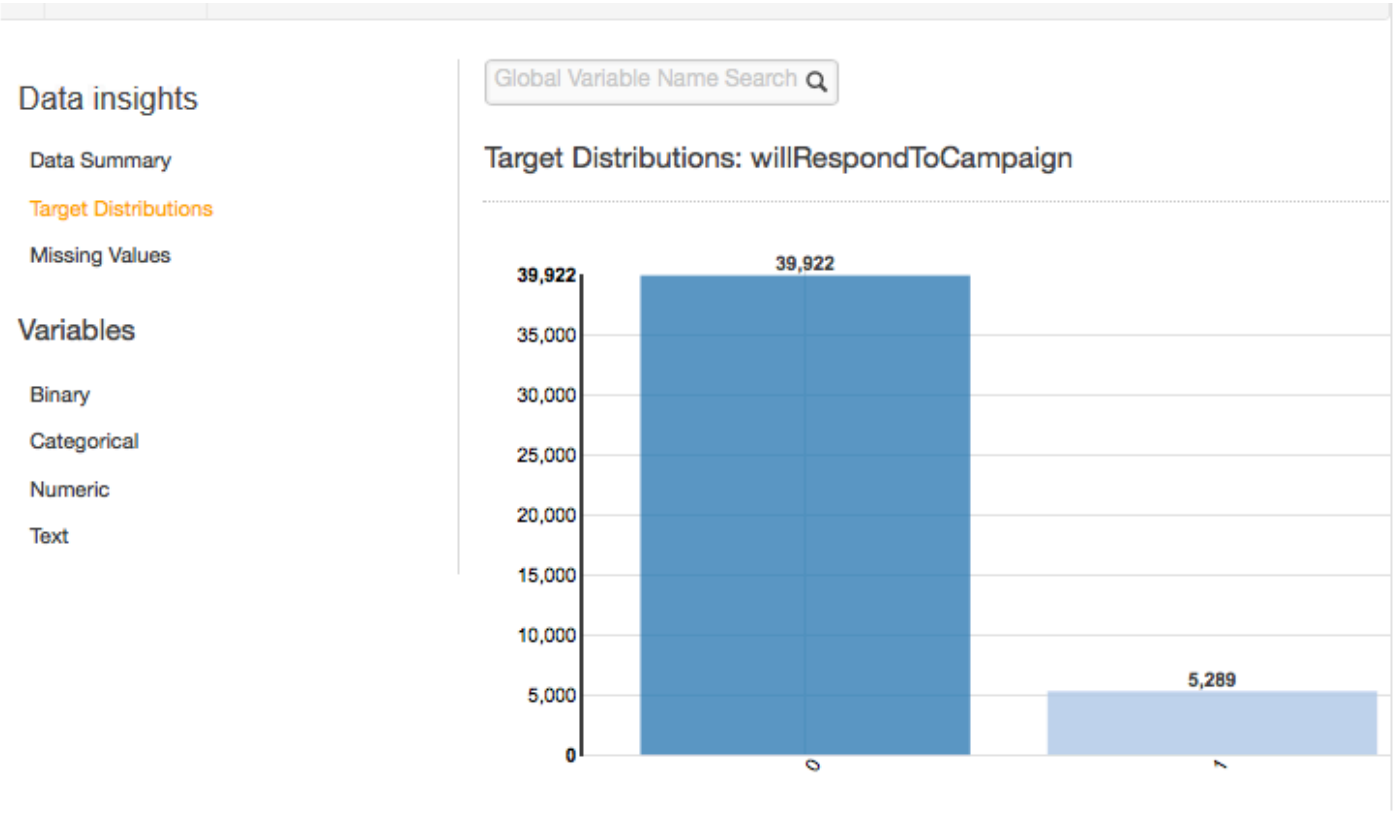
데이터 요약

데이터 소스의 데이터 요약 보고서에는 데이터 소스 ID, 이름, 완료 위치, 현재 상태, 대상 속성, 입력 데이터 정보(S3 버킷 위치, 데이터 형식, 처리된 레코드 수, 처리 중 발생한 잘못된 레코드 수), 데이터 유형별 변수 수 등의 요약 정보가 표시됩니다.

대상 분포

대상 분포 보고서는 데이터 소스의 대상 속성 분포를 보여줍니다. 다음 예시에서는 willRespondToCampaign 대상 속성이 0인 39,922개의 관측치가 있습니다. 이것은 이메일 캠페인에 응

답하지 않은 고객의 수입니다. 5,289개의 관측 결과가 있으며, willRespondToCampaign은 1입니다. 이것은 이메일 캠페인에 응답한 고객 수입니다.



누락 값

누락 값 보고서에는 입력 데이터에서 누락된 값이 있는 속성이 나열됩니다. 숫자 데이터 유형의 속성에만 누락된 값이 있을 수 있습니다. 누락 값은 ML 모델 학습 품질에 영향을 미칠 수 있으므로 가능하면 누락 값을 제공하는 것이 좋습니다.

ML 모델 학습 중에 대상 속성이 누락된 경우 Amazon ML은 해당 레코드를 거부합니다. 대상 속성이 레코드에 있지만 다른 숫자 속성 값이 누락된 경우 Amazon ML은 누락된 값을 간과합니다. 이 경우 Amazon ML은 대체 속성을 생성하고 이 속성을 1로 설정하여 이 속성이 누락되었음을 나타냅니다. 이를 통해 Amazon ML은 누락된 값의 발생으로부터 패턴을 학습할 수 있습니다.

유효하지 않은 값

유효하지 않은 값은 숫자 및 이진 데이터 유형에서만 발생할 수 있습니다. 데이터 유형 보고서에서 변수의 요약 통계를 보면 유효하지 않은 값을 찾을 수 있습니다. 다음 예에서는 Duration Number 속성에 유효하지 않은 값이 하나 있고 이진 데이터 유형에는 유효하지 않은 값이 두 개 있습니다(주택 속성과 대출 속성에 하나).

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%)	

변수-대상 상관 관계

데이터 소스를 생성한 후 Amazon ML은 데이터 소스를 평가하고 변수와 대상 간의 상관 관계 또는 영향을 확인할 수 있습니다. 예를 들어 제품 가격은 베스트셀러 여부에 큰 영향을 미칠 수 있지만 제품의 크기는 예측력이 거의 없을 수 있습니다.

일반적으로 학습 데이터에 최대한 많은 변수를 포함시키는 것이 가장 좋습니다. 그러나 예측력이 거의 없는 변수를 많이 포함시킴으로써 발생하는 노이즈는 ML 모델의 품질과 정확성에 부정적인 영향을 미칠 수 있습니다.

모델을 학습할 때 영향이 거의 없는 변수를 제거하면 모델의 예측 성능을 개선할 수 있습니다. ML의 변환 메커니즘인 레시피로 기계 학습 프로세스에 사용할 수 있는 변수를 정의할 수 있습니다. 레시피에 대해 자세히 알아보려면 기계 학습을 위한 데이터 변환 단원을 참조하세요.

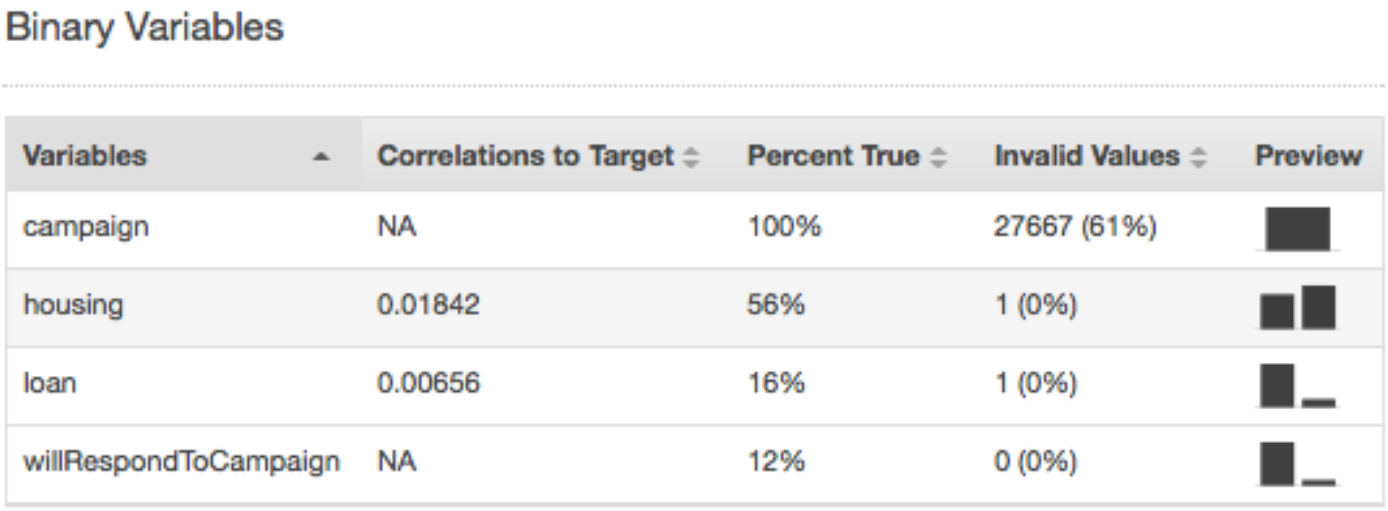
데이터 유형별 속성의 요약 통계

데이터 인사이트 보고서에서 다음 데이터 유형별로 속성 요약 통계를 볼 수 있습니다.

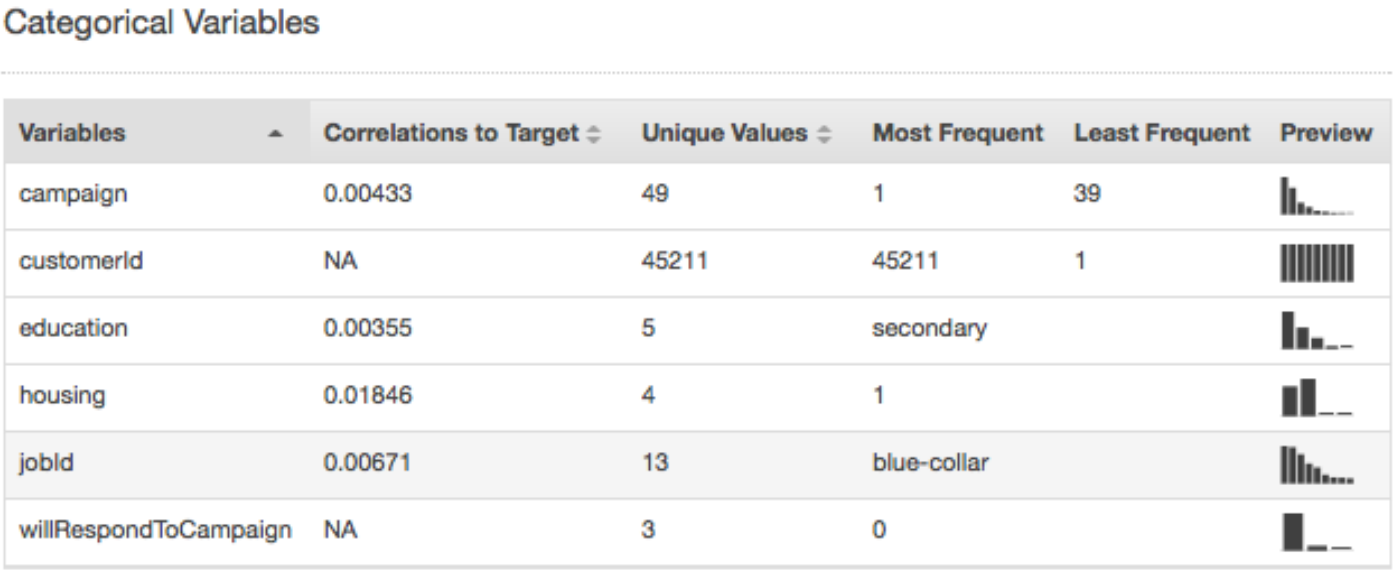
- 이진
- 범주형

- Numeric
- 텍스트

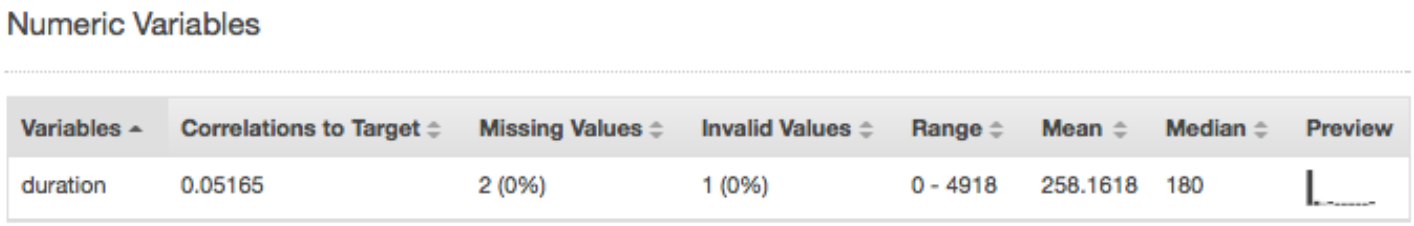
이진 데이터 유형에 대한 요약 통계는 모든 이진 속성을 보여줍니다. 대상과의 상관 관계 열에는 대상 열과 속성 열 간에 공유되는 정보가 표시됩니다. 실제 백분율 열에는 값이 1인 관측치의 백분율이 표시됩니다. 유효하지 않은 값 열에는 유효하지 않은 값의 수와 각 속성에 대한 유효하지 않은 값의 백분율이 표시됩니다. 미리 보기 열에는 각 속성의 그래픽 분포에 대한 링크가 제공됩니다.



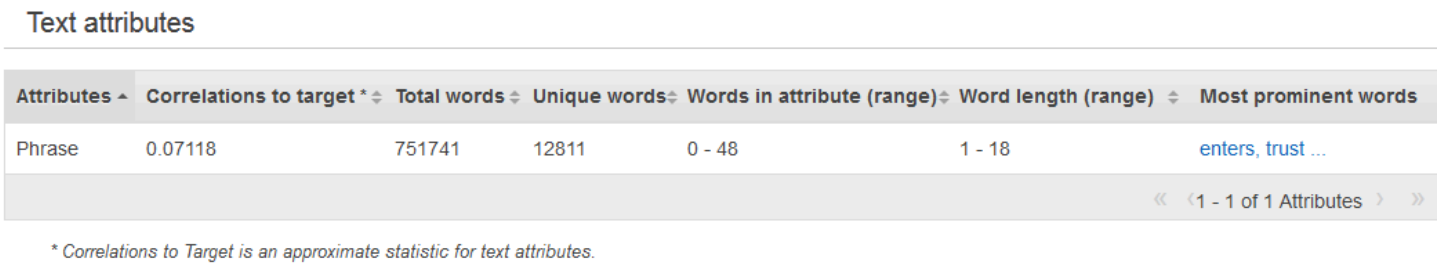
범주형 데이터 유형에 대한 요약 통계는 모든 범주형 속성을 고유 값 수, 가장 빈번한 값 및 최소 빈도 값과 함께 표시합니다. 미리 보기 열에는 각 속성의 그래픽 분포에 대한 링크가 제공됩니다.



숫자 데이터 유형의 요약 통계에는 누락된 값 수, 유효하지 않은 값, 값 범위, 평균 및 중앙값이 포함된 모든 숫자 속성이 표시됩니다. 미리 보기 열에는 각 속성의 그래픽 분포에 대한 링크가 제공됩니다.



텍스트 데이터 유형에 대한 요약 통계에는 모든 텍스트 속성, 해당 속성의 총 단어 수, 해당 속성의 고유 단어 수, 속성의 단어 범위, 단어 길이 범위, 가장 눈에 띄는 단어 등이 표시됩니다. 미리 보기 열에는 각 속성의 그래픽 분포에 대한 링크가 제공됩니다.



다음 예제에서는 4개의 레코드가 포함된 review라는 텍스트 변수에 대한 텍스트 데이터 유형 통계를 보여줍니다.

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

이 예제의 열에는 다음 정보가 표시됩니다.

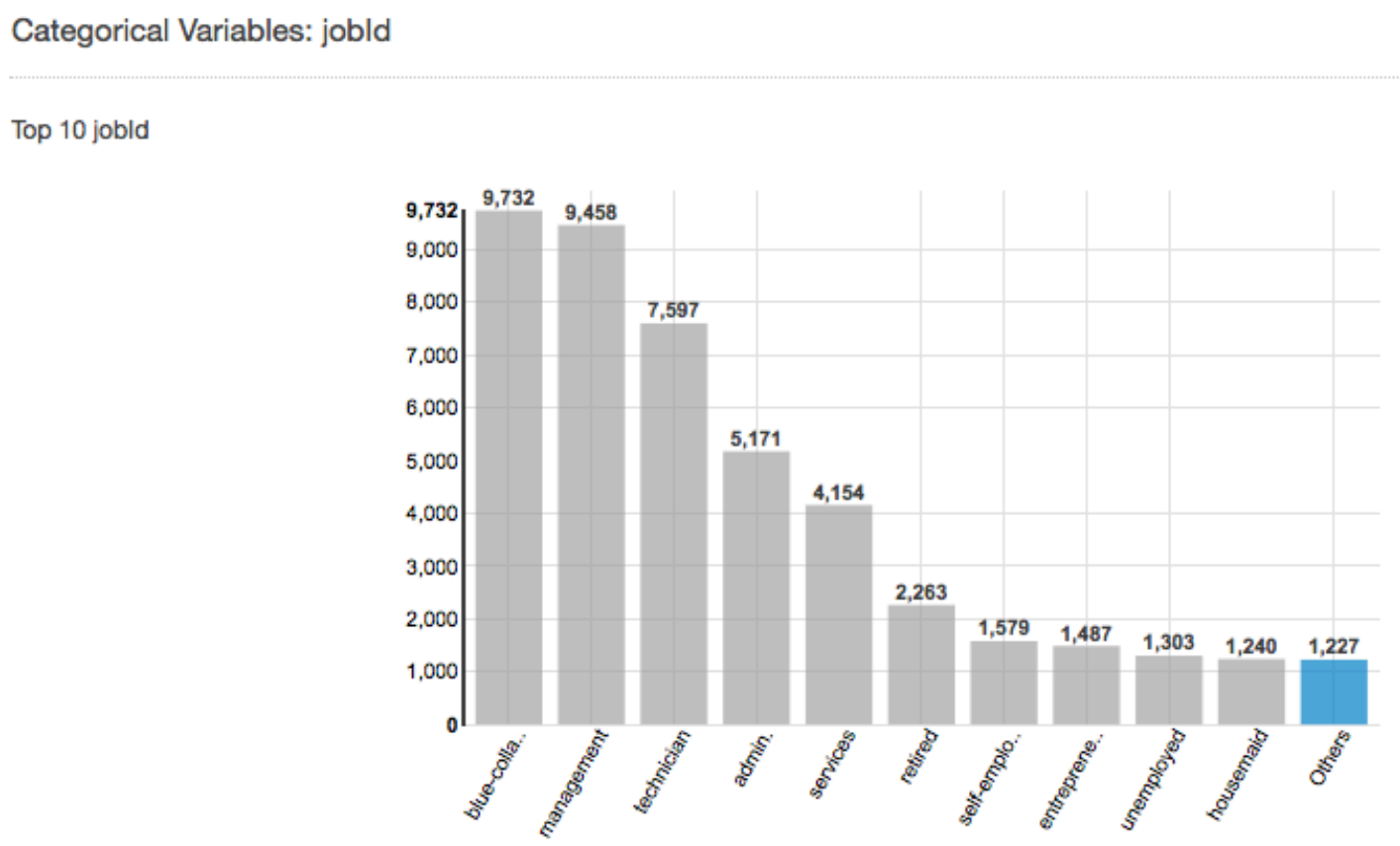
- 속성 열에는 변수 이름이 표시됩니다. 이 예제에서 이 열에는 “review”라고 표시됩니다.
- 대상과의 상관 관계 열은 대상이 지정된 경우에만 존재합니다. 상관 관계는 이 속성이 대상에 대해 제공하는 정보의 양을 측정합니다. 상관 관계가 높을수록 이 속성을 통해 대상에 대해 더 많은 정보를 얻을 수 있습니다. 상관 관계는 텍스트 속성의 단순화된 표현과 대상 간의 상호 정보 측면에서 측정됩니다.
- 전체 단어 열에는 각 레코드를 토큰화하여 생성된 단어 수가 표시되며 단어를 공백으로 구분합니다. 이 예제에서 이 열의 이름은 “12”입니다.

- 고유 단어 열에는 속성의 고유 단어 수가 표시됩니다. 이 예제에서 이 열의 이름은 “10”입니다.
- 속성(범위) 내 단어 수 열에는 속성의 단일 행에 있는 단어 수가 표시됩니다. 이 예제에서 이 열은 “0-6”입니다.
- 단어 길이(범위) 열에는 단어의 문자 수 범위가 표시됩니다. 이 예제에서 이 열은 “2-11”로 표시됩니다.
- 가장 눈에 띄는 단어 열에는 속성에 나타나는 단어의 순위 목록이 표시됩니다. 대상 속성이 있는 경우 대상과의 상관 관계를 기준으로 단어의 순위가 매겨집니다. 즉, 상관 관계가 가장 높은 단어가 먼저 나열됩니다. 데이터에 대상이 없는 경우 단어의 엔트로피를 기준으로 순위가 매겨집니다.

범주형 및 이진 속성의 분포 이해

범주형 또는 바이너리 속성과 관련된 미리 보기 링크를 클릭하면 해당 속성의 분포 뿐만 아니라 속성의 각 범주형 값에 대한 입력 파일의 샘플 데이터를 볼 수 있습니다.

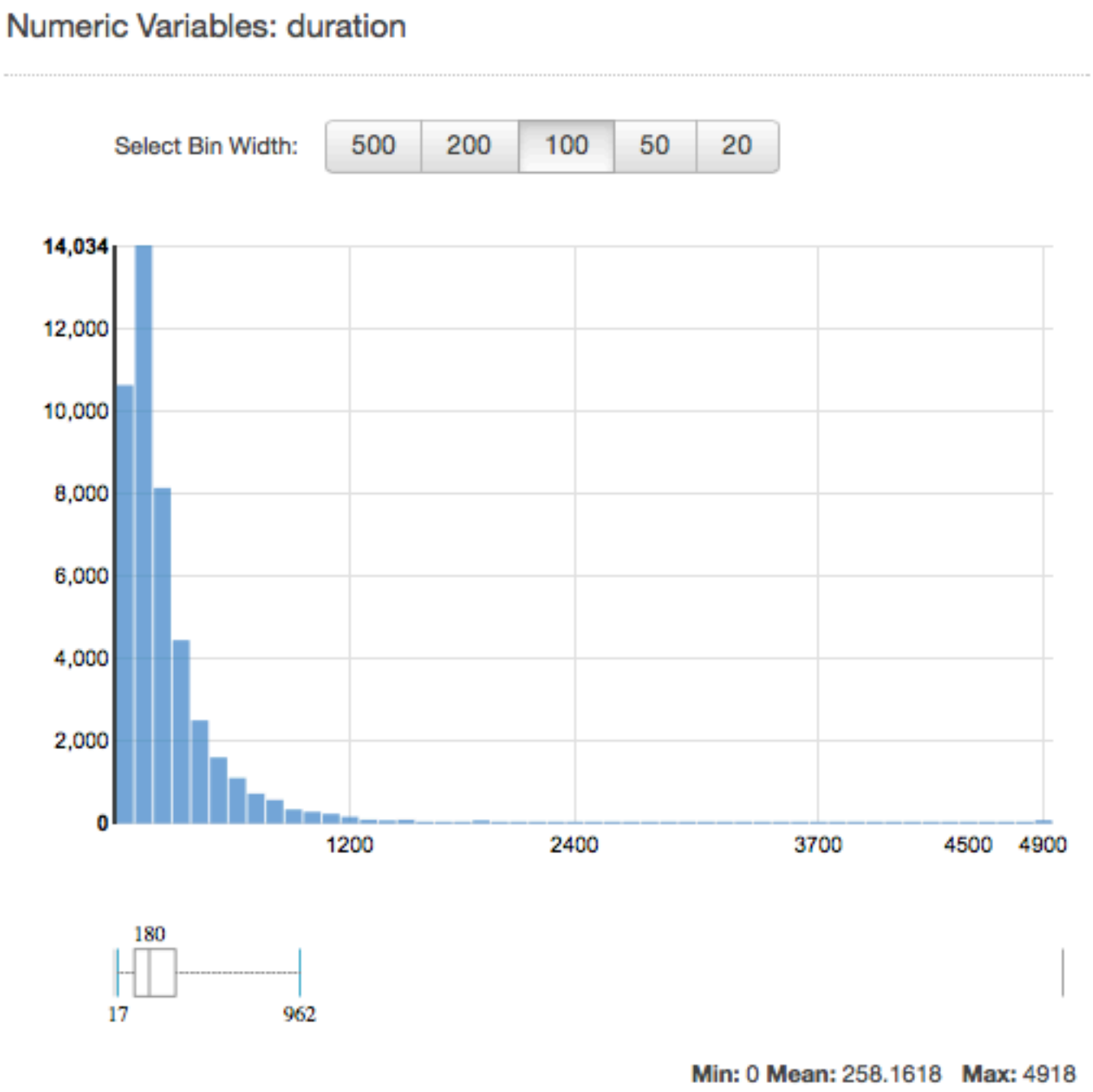
예를 들어 다음 스크린샷에서는 범주형 속성 JoBid에 대한 분포를 보여줍니다. 분포에는 상위 10개 범주형 값이 표시되며 다른 모든 값은 “기타”로 그룹화됩니다. 입력 파일에서 해당 값을 포함하는 입력 파일의 관측치 수와 입력 데이터 파일의 샘플 관측치를 볼 수 있는 링크를 사용하여 상위 10개 범주형 값 각각의 순위를 매깁니다.



숫자 속성의 분포에 대한 이해

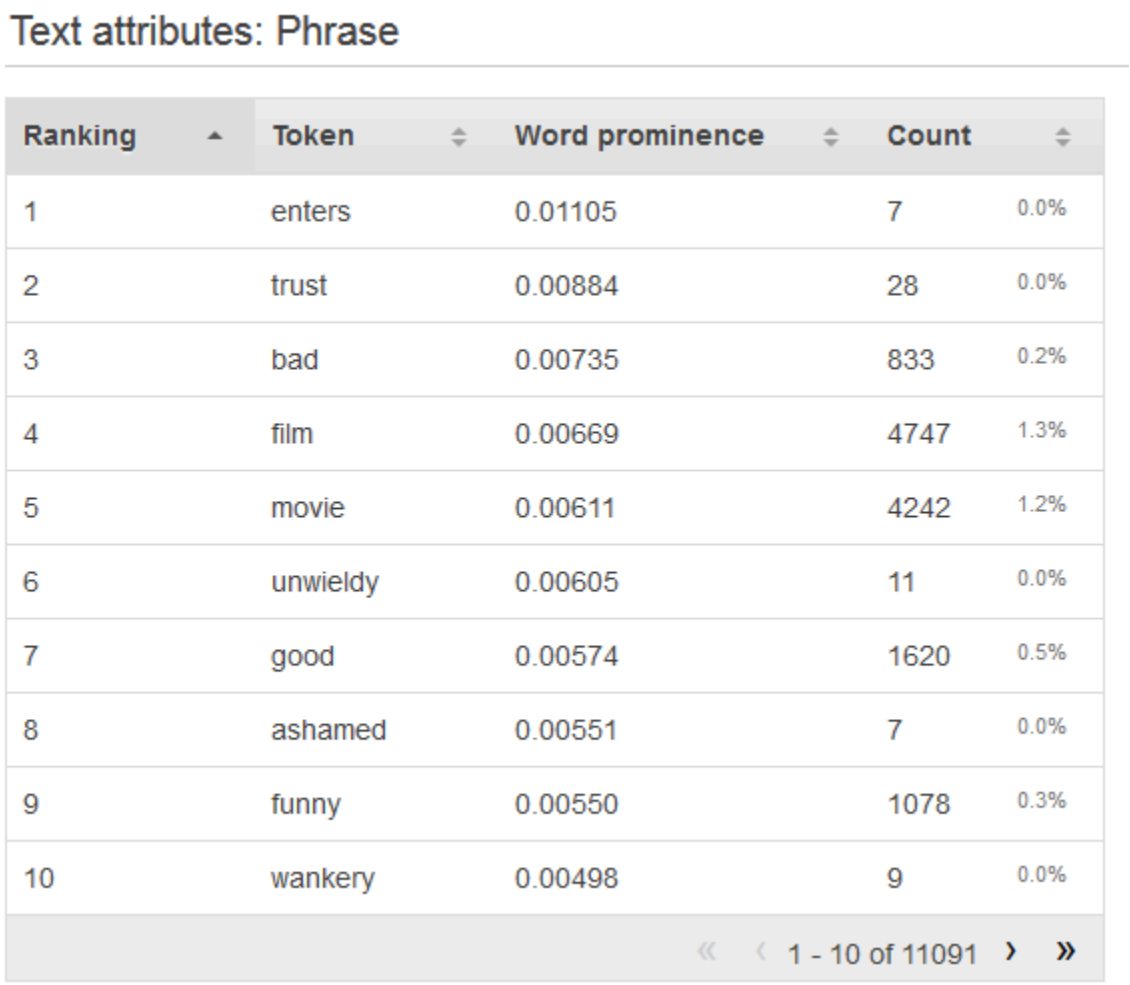
숫자 속성의 분포를 보려면 속성의 미리 보기 링크를 클릭합니다. 숫자 속성의 분포를 볼 때 빈 크기를 500, 200, 100, 50 또는 20 중에서 선택할 수 있습니다. 빈 크기가 클수록 표시되는 막대 그래프 수가 줄어듭니다. 또한 빈 크기가 크면 분포의 분해능이 약해집니다. 반대로 버킷 크기를 20으로 설정하면 표시된 분포의 해상도가 높아집니다.

다음 스크린샷과 같이 최소값, 평균값 및 최대값도 표시됩니다.



텍스트 속성 분포에 대한 이해

텍스트 속성의 분포를 보려면 속성의 미리 보기 링크를 클릭합니다. 텍스트 속성의 분포를 볼 때 다음 정보가 표시됩니다.



순위 결정

텍스트 토큰은 전달하는 정보의 양(정보가 가장 많은 것부터 가장 적은 것까지)을 기준으로 순위가 매겨집니다.

토큰

토큰은 입력 텍스트에서 통계 행과 관련된 단어를 표시합니다.

단어 돌출

대상 속성이 있는 경우 대상과의 상관 관계를 기준으로 단어의 순위가 매겨지므로 상관 관계가 가장 높은 단어를 먼저 나열합니다. 데이터에 대상이 없는 경우 단어의 엔트로피, 즉 전달할 수 있는 정보의 양을 기준으로 단어의 순위가 매겨집니다.

카운트 수

카운트 수는 해당 토큰이 나타난 입력 레코드 수를 나타냅니다.

카운트 백분율

카운트 백분율은 토큰이 나타난 입력 데이터 행의 백분율을 나타냅니다.

Amazon ML에서 Amazon S3 사용

Amazon Simple Storage Service(Amazon S3)는 인터넷 스토리지 서비스입니다. Amazon S3를 사용하면 인터넷을 통해 언제 어디서든 원하는 양의 데이터를 저장하고 검색할 수 있습니다. Amazon ML은 Amazon S3를 다음 작업을 위한 기본 데이터 리포지토리로 사용합니다.

- 입력 파일에 액세스하여 ML 모델을 학습시키고 평가하기 위한 데이터 소스 객체를 생성하기 위해.
- 입력 파일에 액세스하여 배치 예측을 생성하기 위해.
- ML 모델을 사용하여 배치 예측을 생성하는 경우, 예측 파일을 지정한 S3 버킷으로 출력하기 위해.
- Amazon Redshift 또는 Amazon Relational Database Service(Amazon RDS)에 저장한 데이터를 .csv 파일로 복사하여 Amazon S3에 업로드하기 위해.

Amazon ML이 이러한 작업을 수행할 수 있도록 Amazon ML에 Amazon S3 데이터에 액세스할 권한을 부여해야 합니다.

> **Note**
>
> 서버 측 암호화된 파일만 허용하는 S3 버킷에는 배치 예측 파일을 출력할 수 없습니다. 요청에 `s3:x-amz-server-side-encryption` 헤더가 없는 경우 정책에 `s3:PutObject` 작업에 대한 `Deny` 효과가 포함되어 있지 않은지 확인하여 암호화되지 않은 파일의 업로드를 버킷 정책에서 허용하는지 확인합니다. S3 서버 측 암호화 버킷 정책에 대한 자세한 내용은 Simple Storage Service 사용 설명서의 서버 측 암호화를 사용하여 데이터 보호 단원을 참조하세요.

Amazon S3에 데이터 업로드

Amazon ML이 Amazon S3 위치의 데이터를 읽기 때문에 Amazon Simple Storage Service(Amazon S3)에 입력 데이터를 업로드해야 합니다. Amazon S3로 직접 데이터를 업로드하거나(예: 컴퓨터에서) Amazon ML에서 Amazon Redshift 또는 Amazon Redational Database Service(RDS)에 저장한 데이터를 .csv 파일로 복사하여 Amazon S3에 업로드할 수 있습니다.

Redshift 또는 RDS에서 데이터를 복사하는 방법에 대한 자세한 내용은 각각 ML에서 Redshift 사용 또는 ML에서 RDS 사용 단원을 참조하세요.

이 단원의 나머지 부분에서는 입력 데이터를 컴퓨터에서 Amazon S3로 직접 업로드하는 방법을 설명합니다. 이 단원의 절차를 따르기 전에 데이터를 .csv 파일에 넣어 놓아야 합니다. ML에서 사용할 수 있도록 .csv 파일의 형식을 올바르게 지정하는 방법에 대한 자세한 내용은 ML의 데이터 형식 이해 단원을 참조하세요.

컴퓨터에서 Amazon S3로 데이터를 업로드하려면

1. AWS Management Console에 로그인하고 https://console.aws.amazon.com/s3에서 S3 콘솔을 엽니다.
2. 버킷을 생성하거나 기존 버킷을 선택합니다.
   a. 버킷을 생성하려면 버킷 생성을 선택합니다. 버킷의 이름을 지정하고 지역을 선택한 다음 (사용 가능한 지역 선택 가능) 생성을 선택합니다. 자세한 내용은 Simple Storage 시작 안내서의 버킷 생성 단원을 참조하세요.
   b. 기존 버킷을 사용하려면 모든 버킷 목록에서 버킷을 선택하여 해당 버킷을 검색합니다. 버킷 이름이 나타나면 선택한 다음 업로드를 선택합니다.
3. 업로드 대화 상자에서 파일 추가를 선택합니다.
4. 입력 데이터 .csv 파일이 있는 폴더로 이동한 다음 열기를 선택합니다.

권한

Amazon ML이 S3 버킷 중 하나에 액세스할 수 있는 권한을 부여하려면 버킷 정책을 편집해야 합니다.

S3의 버킷의 데이터를 읽을 수 있는 권한을 ML에 부여하는 방법에 대한 자세한 내용은 S3의 데이터를 읽을 수 있는 권한을 ML에 부여 단원을 참조하세요.

ML에 S3의 버킷에 배치 예측 결과를 출력할 권한을 부여하는 방법에 대한 자세한 내용은 S3에 예측을 출력할 수 있는 권한을 ML에 부여 단원을 참조하세요.

S3 리소스에 대한 액세스 권한 관리에 대한 자세한 내용은 S3 개발자 안내서를 참조하세요.

Amazon Redshift의 데이터에서 Amazon ML 데이터 소스 생성

Redshift에 저장된 데이터가 있는 경우 머신 러닝(ML) 콘솔의 데이터 소스 생성 마법사를 사용하여 데이터 소스 객체를 생성할 수 있습니다. Amazon Redshift 데이터에서 데이터 소스를 생성할 때는 데이터를 포함하고 있는 클러스터와 데이터를 검색하기 위한 SQL 쿼리를 지정합니다. ML은 클러스터에서 Redshift `Unload` 명령을 간접적으로 호출하여 쿼리를 실행합니다. Amazon ML은 선택한 Amazon Simple Storage Service(Amazon S3) 위치에 결과를 저장한 다음, Amazon S3에 저장된 데이터를 사용

하여 데이터 소스를 생성합니다. 데이터 소스, Amazon Redshift 클러스터 및 S3 버킷이 모두 같은 지역에 있어야 합니다.

> **Note**
>
> Amazon ML은 프라이빗 VPC의 Amazon Redshift 클러스터에서 데이터 소스를 생성하는 것은 지원하지 않습니다. 클러스터는 공용 IP 주소를 가지고 있어야 합니다.

주제

데이터 소스 생성 마법사에 대한 필수 파라미터

Amazon ML이 사용자 대신 Amazon Redshift 데이터베이스에 연결하고 데이터를 읽을 수 있게 허용하려면 다음 항목을 제공해야 합니다.

- Redshift `ClusterIdentifier`
- Amazon Redshift 데이터베이스 이름
- Amazon Redshift 데이터베이스 보안 인증 정보(사용자 이름 및 암호)
- Amazon ML Amazon Redshift AWS Identity and Access Management (IAM) 역할
- Amazon Redshift SQL 쿼리
- (선택 사항) Amazon ML 스키마의 위치
- Amazon S3 스테이징 위치(데이터 소스를 생성하기 전에 Amazon ML이 데이터를 넣는 위치)

또한 Redshift 데이터 소스를 생성하는 IAM 사용자 또는 역할에 `iam:PassRole` 권한이 있음을 확인해야 합니다(콘솔을 통해 또는 `CreateDatasourceFromRedshift` 작업을 사용하여).

Redshift **`ClusterIdentifier`**

Amazon ML이 클러스터를 찾아서 연결할 수 있도록 하려면 이 파라미터(대소문자 구분)를 사용합니다. Amazon Redshift 콘솔에서 클러스터 식별자(이름)를 얻을 수 있습니다. 클러스터에 대한 자세한 내용은 Redshift 클러스터 섹션을 참조하세요.

Redshift 데이터베이스 이름

데이터 소스로 사용할 데이터를 포함하고 있는 Amazon Redshift 클러스터의 데이터베이스를 Amazon ML에 알리려면 이 파라미터를 사용합니다.

Redshift 데이터베이스 보안 인증 정보

보안 쿼리가 실행될 컨텍스트에서 Amazon Redshift 데이터베이스 사용자의 사용자 이름과 암호를 지정하려면 이 파라미터를 사용합니다.

> **Note**
>
> Amazon Redshift 데이터베이스에 연결하려면 Amazon ML에 Amazon Redshift 사용자 이름과 암호가 필요합니다. Amazon S3에 데이터를 언로드한 후 Amazon ML은 암호를 재사용하거나 저장하지 않습니다.

ML Redshift 역할

Amazon Redshift 클러스터에 대한 보안 그룹 및 Amazon S3 스테이징 위치에 대한 버킷 정책을 구성하기 위해 Amazon ML이 사용해야 하는 IAM 역할의 이름을 지정하려면 이 파라미터를 사용합니다.

Amazon Redshift에 액세스할 수 있는 IAM 역할이 없으면 Amazon ML이 대신 역할을 생성할 수 있습니다. Amazon ML은 역할을 만들 때 고객 관리형 정책을 생성하여 IAM 역할에 연결합니다. Amazon ML이 생성하는 정책은 사용자가 지정하는 클러스터에만 액세스할 수 있는 권한을 Amazon ML에 부여합니다.

Amazon Redshift에 액세스할 수 있는 IAM 역할이 이미 있는 경우에는 역할의 ARN을 입력하거나 드롭다운 목록에서 해당 역할을 선택할 수 있습니다. Amazon Redshift 액세스 권한이 있는 IAM 역할은 드롭다운 목록 상단에 나열됩니다.

IAM 역할에 다음과 같은 내용이 포함되어 있어야 합니다.

JSON

```
{
    "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/*" }
        }
    }]
}
```

고객 관리형 정책에 대한 자세한 내용은 IAM 사용 설명서의 고객 관리형 정책 단원을 참조하세요.

Redshift SQL 쿼리

Amazon ML이 데이터를 선택하기 위해 Amazon Redshift 데이터베이스에서 실행하는 SQL SELECT 쿼리를 지정하려면 이 파라미터를 사용합니다. ML은 Redshift 언로드 작업을 사용하여 쿼리 결과를 S3 위치에 안전하게 복사합니다.

> **Note**
>
> Amazon ML은 입력 레코드가 임의의 순서로 되어 있을 때(셔플링됨) 가장 효과적으로 작동합니다. Redshift 랜덤 함수를 사용하면 Redshift SQL 쿼리의 결과를 쉽게 셔플링할 수 있습니다. 예를 들어 원본 쿼리가 다음과 같다고 가정해 보겠습니다.
>
> ```
> "SELECT col1, col2, … FROM training_table"
> ```
>
> 다음과 같이 쿼리를 업데이트하여 임의 셔플링을 포함할 수 있습니다.
>
> ```
> "SELECT col1, col2, … FROM training_table ORDER BY random()"
> ```

스키마 위치(선택 사항)

Amazon ML이 내보낼 Amazon Redshift 데이터의 스키마에 대한 Amazon S3 경로를 지정하려면 이 파라미터를 사용합니다.

데이터 소스에 대한 스키마를 제공하지 않는 경우 Amazon ML 콘솔은 Amazon Redshift SQL 쿼리의 데이터 스키마를 기반으로 Amazon ML 스키마를 자동으로 생성합니다. Amazon ML 스키마는 Amazon Redshift 스키마보다 데이터 유형 수가 적기 때문에 일대일 변환이 아닙니다. Amazon ML 콘솔은 다음 변환 스키마를 사용하여 Amazon Redshift 데이터 형식을 Amazon ML 데이터 형식으로 변환합니다

Amazon Redshift 데이터 형식.	Amazon Redshift 별칭	Amazon ML 데이터 유형
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
DATE		TEXT
TIMESTAMP	TIMESTAMP WITHOUT TIME ZONE	TEXT

ML `Binary` 데이터 형식으로 변환되려면 데이터의 Redshift 부울 값이 ML 이진 값을 지원해야 합니다. 부울 데이터 형식에 지원되지 않는 값이 있는 경우 Amazon ML은 이 값을 가능한 한 가장 특정한 데이터 형식으로 변환합니다. 예를 들어 Redshift 부울에 값 `0`, `1`, `2`이 있는 경우 ML은 부울을 `Numeric` 데이터 형식으로 변환합니다. 지원되는 이진 값에 대한 자세한 내용은 AttributeType 필드 사용 단원을 참조하세요.

ML이 데이터 형식을 파악할 수 없는 경우 `Text`이 기본값입니다.

Amazon ML이 스키마를 변환한 후 데이터 소스 생성 마법사에서 할당된 Amazon ML 데이터 형식을 검토하고 교정할 수 있으며 Amazon ML이 데이터 소스를 생성하기 전에 스키마를 수정할 수 있습니다.

S3 스테이징 위치

Amazon ML이 Amazon Redshift SQL 쿼리 결과를 저장하는 Amazon S3 스테이징 위치의 이름을 지정하려면 이 파라미터를 사용합니다. 데이터 소스를 생성한 후 Amazon ML은 Amazon Redshift로 반환하는 대신 스테이징 위치의 데이터를 사용합니다.

> **Note**
>
> Amazon ML이 Amazon ML Amazon Redshift 역할에 정의된 IAM 역할을 맡기 때문에 Amazon ML은 지정된 Amazon S3 스테이징 위치에 있는 모든 객체에 액세스할 수 있는 권한을 갖게 됩니다. 따라서 중요한 정보가 포함되지 않은 파일만 Amazon S3 스테이징 위치에 저장하는 것이 좋습니다. 예를 들어 루트 버킷이 `s3://mybucket/`인 경우 `s3://mybucket/AmazonMLInput/`처럼 ML이 액세스할 수 있게 하려는 파일만 저장할 위치를 생성하는 것이 좋습니다.

Amazon Redshift 데이터로 데이터 소스 생성(콘솔)

Amazon ML 콘솔에서 두 가지 방법으로 Amazon Redshift 데이터를 사용하여 데이터 소스를 생성할 수 있습니다. 데이터 소스 생성 마법사를 완료하여 데이터 소스를 만들거나, Amazon Redshift 데이터에서 생성한 데이터 소스가 이미 있는 경우 기존 데이터 소스를 복사하고 설정을 수정할 수 있습니다. 데이터 소스를 복사하면 여러 유사한 데이터 소스를 쉽게 생성할 수 있습니다.

API를 사용하여 데이터 소스를 만드는 데 대한 자세한 내용은 CreateDataSourceFromRedshift 단원을 참조하세요.

다음 절차의 파라미터에 대한 자세한 내용은 데이터 소스 생성 마법사에 대한 필수 파라미터 단원을 참조하세요.

주제

- 데이터 소스 생성(콘솔)
- 데이터 소스 복사(콘솔)

데이터 소스 생성(콘솔)

Amazon Redshift의 데이터를 Amazon ML 데이터 소스에 언로드하려면 데이터 소스 생성 마법사를 사용합니다.

Amazon Redshift의 데이터에서 데이터 소스를 생성하려면

1. https://console.aws.amazon.com/machinelearning/에서 머신 러닝 콘솔을 엽니다.
2. ML 대시보드의 개체에서 새로 만들기...를 선택한 후 데이터 소스를 선택합니다.
3. 입력 데이터 페이지에서 Redshift를 선택합니다.
4. 데이터 소스 생성 마법사의 클러스터 식별자에서 클러스터 이름을 입력합니다.
5. 데이터베이스 이름에서 Redshift 데이터베이스의 이름을 입력합니다.
6. 데이터베이스 사용자 이름에 데이터베이스 사용자 이름을 입력합니다.
7. 데이터베이스 암호에서 데이터베이스 암호를 입력합니다.
8. IAM 역할에 대해 IAM 역할을 선택합니다. 역할이 아직 없는 경우 새 역할 생성을 선택합니다. Amazon ML이 대신 Amazon Redshift IAM 역할을 생성합니다.
9. Redshift 설정을 테스트하려면 액세스 테스트(IAM 역할 옆에 있음)를 선택합니다. Amazon ML이 제공된 설정을 통해 Amazon Redshift에 연결할 수 없는 경우 데이터 소스 생성 작업을 계속 진행할 수 없습니다. 문제 해결에 대한 도움말은 오류 문제 해결 단원을 참조하세요.
10. SQL 쿼리에서 SQL 쿼리를 입력합니다.
11. 스키마 위치에서 ML이 스키마를 생성할지 여부를 선택합니다. 자체적으로 스키마를 생성한 경우 스키마 파일에 대한 Amazon S3 경로를 입력합니다.
12. S3 스테이징 위치에서 ML이 Redshift에서 언로드하는 데이터를 배치할 버킷에 대한 S3 경로를 입력합니다.
13. (선택 사항) 데이터 소스 이름에서 데이터 소스 이름을 입력합니다.
14. 확인을 선택합니다. Amazon ML이 자신이 Amazon Redshift 데이터베이스에 연결할 수 있는지 확인합니다.
15. 스키마 페이지에서 모든 속성에 대한 데이터 형식을 검토하고 필요에 따라 수정합니다.
16. 계속을 선택합니다.
17. 이 데이터 소스를 사용하여 ML 모델을 만들거나 평가하려는 경우 이 데이터 세트를 사용하여 ML 모델을 만들거나 평가할 계획입니까?에 대해 예를 선택합니다. 예를 선택한 경우 대상 행을 선택합니다. 대상에 대한 자세한 내용은 targetAttributeName 필드 사용 단원을 참조하세요.

    이미 생성한 모델과 함께 이 데이터 소스를 사용하여 예측을 생성하려는 경우 아니요를 선택합니다.
18. 계속을 선택합니다.
19. 데이터에 식별자가 포함되어 있습니까?에 대해 데이터에 행 식별자가 없는 경우 아니요를 선택합니다.

데이터에 행 식별자가 포함되어 있는 경우 예를 선택합니다. 행 식별자에 대한 자세한 내용은 rowID 필드 사용 단원을 참조하세요.

20. 검토를 선택합니다.
21. 검토 페이지에서 설정을 검토한 다음 완료를 선택합니다.

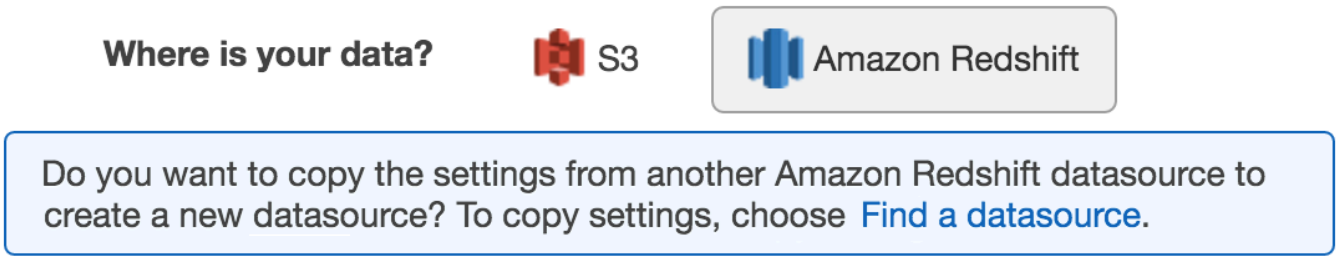
데이터 소스를 생성한 후 이를 사용하여 create an ML model 작업을 수행할 수 있습니다. 모델을 이미 생성했으면 데이터 소스를 사용하여 evaluate an ML model 또는 generate predictions 작업을 수행할 수 있습니다.

데이터 소스 복사(콘솔)

기존 데이터 소스와 비슷한 데이터 소스를 생성하려고 할 때 Amazon ML 콘솔을 사용하여 기존 데이터 소스를 복사하고 설정을 수정할 수 있습니다. 예를 들어 기존 데이터 소스로 시작한 다음 데이터에 더 가깝게 일치하도록 데이터 스키마를 수정하거나, Amazon Redshift에서 데이터를 언로드하는 데 사용되는 SQL 쿼리를 변경하거나, Amazon Redshift 클러스터에 액세스할 다른 AWS Identity and Access Management (IAM) 사용자를 지정할 수 있습니다.

Amazon Redshift 데이터 소스를 복사하고 수정하려면

1. https://console.aws.amazon.com/machinelearning/에서 머신 러닝 콘솔을 엽니다.
2. ML 대시보드의 개체에서 새로 만들기...를 선택한 후 데이터 소스를 선택합니다.
3. 입력 데이터 페이지의 데이터는 어디에 있습니까?에 대해 Redshift를 선택합니다. Amazon Redshift 데이터에서 생성한 데이터 소스가 이미 있으면 다른 데이터 소스의 설정을 복사할 수 있는 옵션을 갖게 됩니다.



Amazon Redshift 데이터에서 생성한 데이터 소스가 없으면 이 옵션이 표시되지 않습니다.

4. 데이터 소스 찾기를 선택합니다.

5. 복사하려는 데이터 소스를 선택하고 설정 복사를 선택합니다. Amazon ML이 대부분의 데이터 소스 설정을 기존 데이터 소스의 설정으로 자동으로 채워줍니다. 기존 데이터 소스의 데이터베이스 암호, 스키마 위치 또는 데이터 소스 이름은 복사하지 않습니다.
6. 자동으로 채워진 설정을 원하는 설정으로 수정합니다. 예를 들어 Amazon ML이 Amazon Redshift에서 언로드하는 데이터를 변경하려면 SQL 쿼리를 변경합니다.
7. 데이터베이스 암호에서 데이터베이스 암호를 입력합니다. Amazon ML은 암호를 저장하거나 재사용하지 않으므로 항상 암호를 제공해야 합니다.
8. (선택 사항) 스키마 위치의 경우 ML에서 권장 스키마를 생성해 주길 원합니다를 미리 선택합니다. 이미 스키마를 생성한 경우 생성하여 S3에 저장한 스키마를 사용하고 싶습니다를 선택하고 S3에 있는 스키마 파일에 대한 경로를 입력합니다.
9. (선택 사항) 데이터베이스 이름에서 데이터 소스 이름을 입력합니다. 그렇지 않으면 Amazon ML이 대신 새 데이터 소스 이름을 생성해줍니다.
10. 확인을 선택합니다. Amazon ML이 자신이 Amazon Redshift 데이터베이스에 연결할 수 있는지 확인합니다.
11. (선택 사항) ML이 스키마를 유추한 경우 스키마 페이지에서 모든 속성에 대한 데이터 형식을 검토하고 필요에 따라 수정합니다.
12. 계속을 선택합니다.
13. 이 데이터 소스를 사용하여 ML 모델을 만들거나 평가하려는 경우 이 데이터 세트를 사용하여 ML 모델을 만들거나 평가할 계획입니까?에 대해 예를 선택합니다. 예를 선택한 경우 대상 행을 선택합니다. 대상에 대한 자세한 내용은 targetAttributeName 필드 사용 단원을 참조하세요.

    이미 생성한 모델과 함께 이 데이터 소스를 사용하여 예측을 생성하려는 경우 아니요를 선택합니다.
14. 계속을 선택합니다.
15. 데이터에 식별자가 포함되어 있습니까?의 경우 데이터에 행 식별자가 포함되어 있지 않은 경우 아니요를 선택합니다.

    데이터에 행 식별자가 포함되어 있는 경우 예를 선택하고 식별자로 사용하려는 행을 선택합니다. 행 식별자에 대한 자세한 내용은 rowID 필드 사용 단원을 참조하세요.
16. 검토를 선택합니다.
17. 설정을 검토한 다음, 완료를 선택합니다.

데이터 소스를 생성한 후 이를 사용하여 create an ML model 작업을 수행할 수 있습니다. 모델을 이미 생성했으면 데이터 소스를 사용하여 evaluate an ML model 또는 generate predictions 작업을 수행할 수 있습니다.

Amazon Redshift 문제 해결

Amazon Redshift 데이터 소스, ML 모델 및 평가를 생성하면 Amazon Machine Learning(Amazon ML)이 Amazon ML 콘솔에서 Amazon ML 객체의 상태를 보고합니다. Amazon ML에서 오류 메시지를 반환하는 경우 다음 정보와 리소스를 사용하여 문제를 해결합니다.

ML에 대한 일반적인 FAQ는 머신 러닝 FAQ 단원을 참조하세요. 머신 러닝 포럼에서 답을 검색하고 질문을 올릴 수도 있습니다.

주제

오류 문제 해결

로그 파일 형식이 잘못되었습니다. 유효한 IAM 역할을 제공합니다. 예: arn:aws:iam::YourAccountID:role/YourRedshiftRole.

원인

IAM 역할의 Amazon 리소스 이름(ARN) 형식이 잘못되었습니다.

솔루션

데이터 소스 생성 마법사에서 사용자 역할에 대한 ARN을 수정합니다. 역할 ARN 형식 지정에 대한 자세한 내용은 IAM 사용 설명서의 IAM ARN 단원을 참조하세요. IAM 역할 ARN의 경우 지역은 선택 사항입니다.

역할이 잘못되었습니다. Amazon ML이 <role ARN> IAM 역할을 맡을 수 없습니다. 유효한 IAM 역할을 제공하고 Amazon ML에서 액세스할 수 있도록 합니다.

원인

Amazon ML이 역할을 맡도록 역할이 설정되지 않았습니다.

솔루션

IAM 콘솔에서 ML이 해당 역할을 맡도록 허용하는 신뢰 정책을 갖도록 역할을 편집합니다.

이 <user ARN> 사용자는 <role ARN> IAM 역할을 전달할 권한이 없습니다.

원인

IAM 사용자에게 Amazon ML에 역할을 전달할 수 있는 권한 정책이 없습니다.

솔루션

Amazon ML에 역할을 전달할 수 있도록 허용하는 권한 정책을 IAM 사용자에게 연결합니다. IAM 콘솔에서 IAM 사용자에게 권한 정책을 연결할 수 있습니다.

계정 간 IAM 역할 전달은 허용되지 않습니다. IAM 역할이 이 계정에 속해야 합니다.

원인

다른 IAM 계정에 속한 역할을 전달할 수 없습니다.

솔루션

역할을 생성하는 데 사용한 AWS 계정으로 로그인합니다. IAM 콘솔에서 IAM 역할을 확인할 수 있습니다.

지정된 역할에 작업을 수행할 수 있는 권한이 없습니다. Amazon ML에 필요한 권한을 제공하는 정책이 포함된 역할을 제공합니다.

원인

IAM 역할에 요청한 작업을 수행할 수 있는 권한이 없습니다.

솔루션

IAM 콘솔에서 역할에 연결된 권한 정책을 편집하여 필요한 권한을 제공합니다.

Amazon ML이 해당 Amazon Redshift 클러스터에서 지정된 IAM 역할을 사용하여 보안 그룹을 구성할 수 없습니다.

원인

IAM 역할에 Amazon Redshift 보안 클러스터를 구성하는 데 필요한 권한이 없습니다.

솔루션

IAM 콘솔에서 역할에 연결된 권한 정책을 편집하여 필요한 권한을 제공합니다.

Amazon ML이 클러스터에 보안 그룹을 구성하려고 시도할 때 오류가 발생했습니다. 나중에 다시 시도해 주세요.

원인

Amazon ML이 Amazon Redshift 클러스터에 연결하려고 할 때 문제가 발생했습니다.

솔루션

데이터 소스 생성 마법사에서 제공한 IAM 역할에 필요한 모든 권한이 있는지 확인합니다.

클러스터 ID 형식이 잘못되었습니다. 클러스터 ID는 문자로 시작해야 하며 영숫자 및 하이픈만 포함해야 합니다. 두 개의 연속 하이픈이 포함되거나 하이픈으로 끝날 수 없습니다.

원인

Amazon Redshift 클러스터 ID 형식이 올바르지 않습니다.

솔루션

데이터 소스 생성 마법사에서 클러스터 ID가 영숫자와 하이픈만 포함되도록 수정하고 두 개의 연속 하이픈을 포함하거나 하이픈으로 끝나지 않도록 클러스터 ID를 수정합니다.

<Amazon Redshift 클러스터 이름> 클러스터가 없거나 클러스터가 Amazon ML 서비스와 같은 지역에 있지 않습니다. 이 Amazon ML과 같은 지역에서 클러스터를 지정합니다.

원인

클러스터가 Amazon ML 데이터 소스를 생성하고 있는 지역에 있지 않기 때문에 Amazon ML이 Amazon Redshift 클러스터를 찾을 수 없습니다.

솔루션

클러스터가 Redshift 콘솔 클러스터 페이지에 있는지, Redshift 클러스터가 위치한 동일한 지역에서 데이터 소스를 생성하고 있는지, 데이터 소스 생성 마법사에서 지정된 클러스터 ID가 올바른지 확인합니다.

Amazon ML이 Amazon Redshift 클러스터의 데이터를 읽을 수 없습니다. 올바른 Amazon Redshift 클러스터 ID를 제공합니다.

원인

Amazon ML이 사용자가 지정한 Amazon Redshift 클러스터의 데이터를 읽을 수 없습니다.

솔루션

데이터 소스 생성 마법사에서 올바른 Redshift 클러스터 ID를 지정하고, Redshift 클러스터가 있는 동일한 지역에서 데이터 소스를 생성하고 있는지, 클러스터가 Redshift 클러스터 페이지에 나열되어 있는지 확인합니다.

<Amazon Redshift 클러스터 이름> 클러스터는 공개 액세스가 불가능합니다.

원인

클러스터가 공개 액세스가 불가능하고 공용 IP 주소가 없기 때문에 Amazon ML이 클러스터에 액세스할 수 없습니다.

솔루션

클러스터를 공개 액세스가 가능하게 만들고 공용 IP 주소를 부여합니다. 클러스터를 공개 액세스가 가능하게 만드는 방법에 대한 자세한 내용은 Redshift 관리 안내서의 클러스터 수정 단원을 참조하세요.

<Redshift> 클러스터 상태가 Amazon ML에 사용 불가능합니다. Amazon Redshift 콘솔을 사용하여 이 클러스터 상태 문제를 확인하고 해결합니다. 클러스터의 상태가 “사용 가능”이어야 합니다.

원인

Amazon ML이 클러스터 상태를 확인할 수 없습니다.

솔루션

클러스터가 사용 가능한지 확인합니다. 클러스터 상태 확인에 대한 자세한 내용은 Amazon Redshift 관리 안내서의 클러스터 상태 개요 가져오기를 참조하세요. 클러스터를 사용 가능한 상태가 되도록 재부팅하는 방법에 대한 자세한 내용은 Redshift 관리 안내서의 클러스터 재부팅 단원을 참조하세요.

이 클러스터에 <데이터베이스 이름> 데이터베이스가 없습니다. 데이터베이스 이름이 올바른지 확인하거나 다른 클러스터 및 데이터베이스를 지정합니다.

원인

Amazon ML이 지정된 클러스터에서 지정된 데이터베이스를 찾을 수 없습니다.

솔루션

데이터 소스 생성 마법사에서 입력한 데이터베이스 이름이 올바른지 확인하거나 올바른 클러스터 및 데이터베이스 이름을 지정합니다.

Amazon ML이 데이터베이스에 액세스할 수 없습니다. 데이터베이스 사용자 <사용자 이름>에 유효한 암호를 제공합니다.

원인

Amazon ML이 Amazon Redshift 데이터베이스에 액세스할 수 있도록 데이터 소스 생성 마법사에서 입력한 암호가 올바르지 않습니다.

솔루션

Amazon Redshift 데이터베이스 사용자에 올바른 암호를 제공합니다.

Amazon ML이 쿼리 검증을 시도할 때 오류가 발생했습니다.

원인

SQL 쿼리에 문제가 있습니다.

솔루션

쿼리가 유효한 SQL인지 확인합니다.

SQL 쿼리를 실행하는 동안 오류가 발생했습니다. 데이터베이스 이름과 제공된 쿼리를 확인합니다. 근본 원인: {serverMessage}.

원인

Amazon Redshift가 쿼리를 실행할 수 없었습니다.

솔루션

데이터 소스 생성 마법사에서 올바른 데이터베이스 이름을 지정했고 쿼리가 유효한 SQL인지 확인합니다.

SQL 쿼리를 실행하는 동안 오류가 발생했습니다. 근본 원인: {serverMessage}.

원인

Amazon Redshift가 지정된 표를 찾을 수 없습니다.

솔루션

데이터 소스 생성 마법사에서 지정한 표가 Amazon Redshift 클러스터 데이터베이스에 있는지, 올바른 클러스터 ID, 데이터베이스 이름 및 SQL 쿼리를 입력했는지 확인합니다.

AWS Support에 문의

AWS 프리미엄 서포트가 있는 경우 지원 센터에서 기술 지원 사례를 생성할 수 있습니다.

Amazon RDS 데이터베이스의 데이터를 사용하여 Amazon ML 데이터 소스 생성

Amazon ML을 사용하면 Amazon Realtional Database Service(Amazon RDS)의 MySQL 데이터베이스에 저장된 데이터로부터 데이터 소스 객체를 생성할 수 있습니다. 이 작업을 수행하면 Amazon ML이 지정한 SQL 쿼리를 실행하는 AWS Data Pipeline 객체를 생성하고 선택한 S3 버킷에 출력을 배치합니다. Amazon ML은 이 데이터를 사용하여 데이터 소스를 생성합니다.

> **Note**
>
> Amazon ML은 VPC의 MySQL 데이터베이스만 지원합니다.

Amazon ML에서 입력 데이터를 읽으려면 먼저 해당 데이터를 Amazon Simple Storage Service(Amazon S3)로 내보내야 합니다. API를 사용하여 자동으로 내보내기를 수행하도록 Amazon ML을 설정할 수 있습니다. RDS는 API에서만 사용할 수 있으며 콘솔에서는 사용할 수 없습니다.

Amazon ML이 Amazon RDS의 MySQL 데이터베이스에 연결되고 사용자를 대신하여 데이터를 읽도록 하려면 다음을 제공해야 합니다.

- RDS DB 인스턴스 식별자
- MySQL 데이터베이스 이름
- 데이터 파이프라인을 생성, 활성화 및 실행하는 데 사용되는 AWS Identity and Access Management (IAM) 역할
- 데이터베이스 사용자 자격 증명:
  - 사용자 이름

  - 암호
- AWS Data Pipeline 보안 정보:
  - IAM 리소스 역할
  - IAM 서비스 역할
- Amazon RDS 보안 정보:
  - 서브넷 ID
  - 보안 그룹 ID
- 데이터 소스를 생성하는 데 사용하려는 데이터를 지정하는 SQL 쿼리
- 쿼리 결과를 저장하는 데 사용되는 S3 출력 위치(버킷)
- (선택 사항) 데이터 스키마의 위치

또한 CreateDataSourceFromRDS 작업을 사용하여 RDS 데이터 소스를 생성하는 IAM 사용자 또는 역할에 `iam:PassRole` 권한이 있는지 확인해야 합니다. 자세한 내용은 Amazon ML 리소스에 대한 액세스 제어 - IAM 사용 단원을 참조하십시오.

주제

- RDS 데이터베이스 인스턴스 식별자
- MySQL 데이터베이스 이름
- 데이터베이스 사용자 자격 증명
- AWS Data Pipeline 보안 정보
- Amazon RDS 보안 정보
- MySQL 쿼리
- S3 출력 위치

RDS 데이터베이스 인스턴스 식별자

RDS DB 인스턴스 식별자는 Amazon ML이 Amazon RDS와 상호작용할 때 사용해야 하는 데이터베이스 인스턴스를 식별하기 위해 사용자가 제공하는 고유한 이름입니다. RDS DB 인스턴스 식별자는 Amazon RDS 콘솔 내에서 찾을 수 있습니다.

MySQL 데이터베이스 이름

MySQL 데이터베이스 이름은 RDS DB 인스턴스의 MySQL 데이터베이스 이름을 지정합니다.

데이터베이스 사용자 자격 증명

RDS DB 인스턴스에 연결하려면 제공한 SQL 쿼리를 실행할 수 있는 충분한 권한을 가진 데이터베이스 사용자의 사용자 이름과 암호를 제공해야 합니다.

AWS Data Pipeline 보안 정보

안전한 AWS Data Pipeline 액세스를 가능하게 하려면 IAM 리소스 역할과 IAM 서비스 역할의 이름을 제공해야 합니다.

EC2 인스턴스는 Amazon RDS에서 Amazon S3로 데이터를 복사하는 리소스 역할을 맡고 있습니다. 이 리소스 역할을 생성하는 가장 쉬운 방법은 `DataPipelineDefaultResourceRole` 템플릿을 사용하여 **`machinelearning.aws.com`**를 신뢰할 수 있는 서비스로 등록하는 것입니다. 템플릿에 대한 자세한 내용은 데이터 파이프라인 개발자 안내서의 IAM 역할 설정 단원을 참조하세요.

역할을 직접 생성하는 경우 다음과 같은 내용이 포함되어야 합니다.

JSON

```
{
    "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 Pipeline은 Amazon RDS에서 Amazon S3로 데이터를 복사하는 과정을 모니터링하는 서비스 역할을 맡고 있습니다. 이 리소스 역할을 생성하는 가장 쉬운 방법은 `DataPipelineDefaultRole` 템플릿을 사용하여 `machinelearning.aws.com`를 신뢰할 수 있는

서비스로 등록하는 것입니다. 템플릿에 대한 자세한 내용은 데이터 파이프라인 개발자 안내서의 IAM 역할 설정 단원을 참조하세요.

Amazon RDS 보안 정보

안전한 RDS 액세스를 가능하게 하려면 `VPC Subnet ID` 및 `RDS Security Group IDs`를 제공해야 합니다. 또한 `Subnet ID` 파라미터가 가리키는 VPC 서브넷에 대해 적절한 수신 규칙을 설정하고 이러한 권한이 있는 보안 그룹의 ID를 제공해야 합니다.

MySQL 쿼리

`MySQL SQL Query` 파라미터는 MySQL 데이터베이스에서 실행하려는 SQL SELECT 쿼리를 지정합니다. 쿼리 결과는 지정한 S3 출력 위치(버킷)로 복사됩니다.

> **Note**
>
> 기계 학습 기술은 입력 레코드가 임의 순서(셔플링됨)로 표시될 때 가장 잘 작동합니다. `rand()` 함수를 사용하면 MySQL 쿼리 결과를 쉽게 셔플링할 수 있습니다. 예를 들어 원본 쿼리가 다음과 같다고 가정해 보겠습니다.
> "SELECT col1, col2, … FROM training_table"
> 다음과 같이 쿼리를 업데이트하면 임의 셔플링을 추가할 수 있습니다.
> "SELECT col1, col2, … FROM training_table ORDER BY rand()"

S3 출력 위치

`S3 Output Location` 파라미터는 MySQL 쿼리 결과가 출력되는 “스테이징” S3 위치의 이름을 지정합니다.

> **Note**
>
> Amazon RDS에서 데이터를 내보낸 후에는 Amazon ML에 이 위치에서 데이터를 읽을 권한이 있는지 확인해야 합니다. 이러한 권한을 설정하는 방법에 대한 자세한 내용은 “Amazon S3에서 데이터를 읽을 수 있도록 Amazon ML에 권한 부여” 단원을 참조하세요.

모델 학습

ML 모델을 학습시키는 프로세스에는 학습할 학습 데이터가 포함된 ML 알고리즘(즉, 학습 알고리즘)을 제공하는 작업이 포함됩니다. ML 모델이라는 용어는 학습 프로세스에서 생성되는 모델 아티팩트를 나타냅니다.

학습 데이터에는 정답이 포함되어야 하며, 이를 대상 또는 대상 속성이라고 합니다. 학습 알고리즘은 학습 데이터에서 대상(예측하려는 답)에 입력 데이터 속성을 매핑하는 패턴을 찾아 내서, 이들 패턴을 캡처하는 ML 모델을 출력합니다.

그러면 ML 모델을 사용하여 해당 대상을 모르는 새로운 데이터에서 예측을 얻을 수 있습니다. 예를 들어 이메일이 스팸인지 스팸이 아닌 지를 예측하도록 ML 모델을 학습시키고 싶다고 가정해 보겠습니다. 대상을 알고 있는 이메일이 포함된 학습 데이터(즉, 이메일이 스팸인지 스팸이 아닌 지를 나타내는 레이블)를 Amazon ML에 제공하면 됩니다. Amazon ML은 이 데이터를 사용하여 ML 모델을 학습시켜 새 이메일이 스팸인지 아니면 스팸이 아닌 지를 예측하는 모델을 만듭니다.

ML 모델 및 ML 알고리즘에 대한 일반 정보는 기계 학습 개념 단원을 참조하세요.

주제

ML 모델 유형

Amazon ML은 바이너리 분류, 멀티클래스 분류 및 회귀라는 세 가지 유형의 ML 모델을 지원합니다. 선택해야 하는 모델 유형은 예측하려는 대상의 유형에 따라 달라집니다.

바이너리 분류 모델

바이너리 분류 문제에 대한 ML 모델은 이진 결과(가능한 두 클래스 중 하나)를 예측합니다. Amazon ML은 바이너리 분류 모델을 학습시키기 위해 로지스틱 회귀라는 업계 표준 학습 알고리즘을 사용합니다.

바이너리 분류 문제의 예

- “이 이메일은 스팸인가요, 아니면 스팸이 아닌가요?”

- “고객이 이 제품을 구매할까요?”
- “이 제품은 책인가요 아니면 가축인가요?”
- “이 리뷰는 고객이 작성했나요, 아니면 로봇이 작성했나요?”

멀티클래스 분류 모델

멀티클래스 분류 문제에 대한 ML 모델을 사용하면 여러 클래스에 대한 예측(둘 이상의 결과 중 하나 예측)을 생성할 수 있습니다. 멀티클래스 모델을 학습시키기 위해 Amazon ML은 다항 로지스틱 회귀라는 업계 표준 학습 알고리즘을 사용합니다.

멀티클래스 문제의 예

- 예: “이 제품은 책인가요, 영화인가요, 의류인가요?”
- “이 영화는 로맨틱 코미디인가요, 다큐멘터리인가요, 스릴러인가요?”
- “이 고객이 가장 관심을 갖고 있는 제품 범주는 무엇인가요?”

회귀 모델

회귀 문제에 대한 ML 모델은 숫자 값을 예측합니다. 회귀 모델을 학습시키기 위해 Amazon ML은 선형 회귀라고 하는 업계 표준 학습 알고리즘을 사용합니다.

회귀 문제의 예

- “내일 시애틀의 기온은 어떻게 될까요?”
- “이 제품의 경우 몇 대가 판매될 예정인가요?”
- “이 집은 어떤 가격에 팔릴까요?”

학습 프로세스

ML 모델을 학습시키려면 다음을 지정해야 합니다.

- 입력 학습 데이터 소스
- 예측 대상이 포함된 데이터 속성의 이름
- 필수 데이터 변환 지침
- 학습 알고리즘을 제어하기 위한 학습 파라미터

학습 프로세스 중에 Amazon ML이 사용자가 학습 데이터 소스에 지정한 대상 유형에 따라 올바른 학습 알고리즘을 자동으로 선택합니다.

학습 파라미터

일반적으로 기계 학습 알고리즘은 학습 프로세스 및 결과 ML 모델의 특정 속성을 제어하는 데 사용할 수 있는 파라미터를 받아들입니다. 머신 러닝에서는 이를 학습 파라미터라고 합니다. 이들 파라미터는 Amazon ML 콘솔, API 또는 명령줄 인터페이스(CLI)를 사용하여 설정할 수 있습니다. 파라미터를 설정하지 않으면 Amazon ML은 광범위한 기계 학습 작업에 잘 작동하는 것으로 알려진 기본값을 사용합니다.

다음 학습 파라미터에 대해 값을 지정할 수 있습니다.

- 최대 모델 크기
- 학습 데이터의 최대 전달 횟수.
- 셔플 유형
- 정규화 유형
- 정규화 정도

Amazon ML 콘솔에서는 학습 파라미터가 기본적으로 설정되어 있습니다. 기본 설정은 대부분의 ML 문제에 적합하지만 다른 값을 선택하여 성능을 미세 조정할 수 있습니다. 학습률과 같은 기타 특정 학습 파라미터는 데이터를 기반으로 구성됩니다.

다음에 이어지는 단원에서는 학습 파라미터에 대한 자세한 정보를 제공합니다.

최대 모델 크기

최대 모델 크기는 Amazon ML이 ML 모델을 학습하는 동안 생성하는 패턴의 총 크기(바이트 단위)입니다.

기본적으로 Amazon ML은 100MB 모델을 생성합니다. 크기를 다르게 지정하여 더 작거나 큰 모델을 생성하도록 Amazon ML에 지시할 수 있습니다. 사용 가능한 크기 범위는 ML 모델 유형 단원을 참조하세요.

Amazon ML은 모델 크기를 채울 만큼 충분한 패턴을 찾지 못하면 더 작은 모델을 만듭니다. 예를 들어, 최대 모델 크기를 100MB로 지정했는데 Amazon ML이 총 50MB에 불과한 패턴만 찾아내면 결과 모델은 50MB가 됩니다. Amazon ML은 지정된 크기에 맞는 것보다 더 많은 패턴을 발견하면 학습된 모델의 품질에 가장 영향을 미치지 않는 패턴을 트리밍하여 최대 커트라인을 적용합니다.

모델 크기를 선택하면 모델의 예측 품질과 사용 비용 간의 균형을 조절할 수 있습니다. 모델이 작으면 Amazon ML에서 최대 크기 한도 내에 맞춰 많은 패턴을 제거하여 예측 품질에 영향을 미칠 수 있습니다. 반면 모델이 클수록 실시간 예측을 위한 쿼리 비용이 더 많이 듭니다.

> **Note**
>
> ML 모델을 사용하여 실시간 예측을 생성하는 경우 모델 크기에 따라 결정되는 소액의 용량 예약 요금이 부과됩니다. 자세한 내용은 Amazon EKS 요금 단원을 참조하세요.

모델은 입력 데이터가 아니라 패턴을 저장하기 때문에 입력 데이터 세트가 크다고 해서 반드시 모델 크기가 커지는 것은 아닙니다. 패턴이 적고 단순하면 결과 모델은 작아집니다. 원시 속성(입력 열) 또는 파생 특성(Amazon ML 데이터 변환의 출력)이 많은 입력 데이터는 학습 프로세스 중에 더 많은 패턴을 발견하고 저장하게 될 것입니다. 몇 번의 실험을 통해 데이터와 문제에 적합한 모델 크기를 선택하는 것이 가장 좋습니다. Amazon ML 모델 학습 로그(콘솔이나 API를 통해 다운로드 가능)에는 학습 프로세스 중에 발생한 모델 트리밍(있는 경우)에 대한 메시지가 포함되어 있어 잠재적인 적중률 예측 품질을 추정할 수 있습니다.

데이터의 최대 전달 횟수.

최상의 결과를 얻으려면 Amazon ML에서 패턴을 찾기 위해 데이터를 여러 번 전달해야 할 수 있습니다. 기본적으로 Amazon ML은 10회의 전달을 수행하지만 숫자를 최대 100개로 설정하여 기본값을 변경할 수 있습니다. Amazon ML은 진행 과정에서 패턴 품질(모델 컨버전스)을 추적하고 더 이상 발견할 데이터 포인트나 패턴이 없을 경우 자동으로 학습을 중단합니다. 예를 들어, 전달 횟수를 20으로 설정했는데 Amazon ML이 15번의 전달이 끝날 때까지 새로운 패턴을 찾을 수 없다는 것을 발견하면 15회 전달이 되면 학습이 중지됩니다.

일반적으로 관측치가 몇 개뿐인 데이터 세트의 경우 모델 품질을 높이려면 일반적으로 데이터를 더 많이 전달해야 합니다. 데이터 세트가 클수록 유사한 데이터 포인트가 많이 포함되는 경우가 많기 때문에 많은 수의 전달이 필요하지 않습니다. 데이터에 대해 더 많은 데이터 전달을 선택하는 것은 두 가지 영향을 미칩니다. 모델 학습에 더 많은 시간이 걸리고 비용도 더 많이 든다는 것입니다.

학습 데이터의 셔플 유형

Amazon ML에서는 학습 데이터를 셔플링해야 합니다. 셔플링은 데이터 순서를 혼합하기 때문에 SGD 알고리즘이 연속적으로 너무 많은 관측치에서 한 가지 유형의 데이터만 접하지 않습니다. 예를 들어, ML 모델을 학습시켜 제품 유형을 예측하고 학습 데이터에 영화, 장난감 및 비디오 게임 제품 유형이 포함된 경우 업로드하기 전에 제품 유형 열을 기준으로 데이터를 정렬하면 알고리즘이 데이터를 제품 유형별로 알파벳순으로 확인합니다. 알고리즘이 먼저 영화에 대한 모든 데이터를 확인하면 ML 모델이 영

화에 대한 패턴을 학습하기 시작합니다. 그런 다음 모델이 장난감에 대한 데이터를 접하면 해당 알고리즘이 수행하는 모든 업데이트는 장난감 제품 유형에 대한 모델에 적합하게 됩니다. 이러한 업데이트로 인해 영화에 적합한 패턴이 저하되는 경우에도 마찬가지입니다. 영화에서 장난감 유형으로 갑자기 전환하면 제품 유형을 정확하게 예측하는 방법을 학습하지 못하는 모델이 생성될 수 있습니다.

입력 데이터 소스를 학습 부분과 평가 부분으로 분리할 때 임의 분할 옵션을 선택한 경우에도 학습 데이터를 셔플링해야 합니다. 임의 분할 전략은 각 데이터 소스에 대해 데이터의 하위 집합을 무작위로 선택하지만 데이터 소스의 행 순서를 변경하지는 않습니다. 데이터 분할에 대한 자세한 내용은 데이터 분할 단원을 참조하세요.

콘솔을 사용하여 ML 모델을 생성할 때 Amazon ML은 기본적으로 유사 무작위 셔플링 기법을 사용하여 데이터를 셔플링합니다. 요청된 전달 횟수에 관계없이 Amazon ML은 ML 모델을 학습시키기 전에 데이터를 한 번만 셔플링합니다. ML에 데이터를 제공하기 전에 데이터를 셔플링했고 ML에서 데이터를 다시 셔플링하지 않도록 하려면 셔플 유형을 none로 설정하면 됩니다. 예를 들어 S3에 업로드하기 전에 .csv 파일의 레코드를 무작위로 셔플링했거나, RDS에서 데이터 소스를 생성할 때 MySQL 쿼리의 `rand()` 함수를 사용했거나, Redshift에서 데이터 소스를 생성할 때 Redshift SQL 쿼리의 `random()` 함수를 사용한 경우, 셔플 유형을 none로 설정해도 ML 모델의 예측 정확도에는 영향을 미치지 않습니다. 데이터를 한 번만 셔플링하면 ML 모델을 만드는 데 드는 런타임과 비용을 줄일 수 있습니다.

> **Important**
>
> Amazon ML API를 사용하여 ML 모델을 생성하면 Amazon ML이 기본적으로 데이터를 셔플링하지 않습니다. 콘솔 대신 API를 사용하여 ML 모델을 생성하는 경우 `sgd.shuffleType` 파라미터를 `auto`로 설정하여 데이터를 셔플링하는 것이 좋습니다.

정규화 유형 및 정도

데이터에 너무 많은 패턴이 포함되어 있으면 복잡한 ML 모델(입력 속성이 많은 모델)의 예측 성능이 저하됩니다. 패턴 수가 늘어날수록 모델이 실제 데이터 패턴 대신 의도하지 않은 데이터 아티팩트를 학습할 가능성도 커집니다. 이 경우 모델은 학습 데이터에서는 잘 작동하지만 새 데이터에 대해서는 일반화할 수 없습니다. 이러한 현상을 학습 데이터의 과적합이라고 합니다.

정규화를 사용하면 극단적인 가중치 값에 페널티를 부과하여 선형 모델이 학습 데이터 예에 과적합되는 것을 방지할 수 있습니다. L1 정규화에서는 적은 가중치를 갖게 될 특성의 가중치를 0으로 만들어 모델에서 사용되는 특성의 수가 감소됩니다. L1 정규화는 희소 모델을 생성하고 모델의 노이즈 양이 감소됩니다. L2 정규화 결과, 전체 가중치가 더 작아져서, 특성 간의 상관 관계가 높을 때 가중치가 안정화됩니다. `Regularization amount` 파라미터를 사용하면 L1 또는 L2 정규화의 정도를 제어할 수

있습니다. 너무 큰 `Regularization amount` 값을 지정하면 모든 특성의 가중치가 0이 될 수 있습니다.

최적의 정규화 값을 선택하고 조정하는 것은 기계 학습 연구에서 활발한 주제입니다. Amazon ML 콘솔의 기본값인 적당한 정도의 L2 정규화를 선택하면 도움이 될 것입니다. 고급 사용자는 세 가지 유형의 정규화(없음, L1 또는 L2)와 정도 중에서 선택할 수 있습니다. 정규화에 대한 자세한 내용은 정규화(수학)에서 확인하세요.

학습 파라미터: 유형 및 기본 값

다음 표에는 Amazon ML 학습 파라미터가 기본 값 및 각 파라미터의 허용 범위와 함께 나열되어 있습니다.

학습 파라미터	유형	기본 값	설명
maxMLModelSizeInBytes	정수	100,000,000바이트(100MiB)	허용 범위: 100,000(100KiB) ~ 2,147,483,648(2GiB) 입력 데이터에 따라 모델 크기가 성능에 영향을 미칠 수 있습니다.
sgd.maxPasses	정수	10	허용 범위: 1~100
sgd.shuffleType	문자열	auto	허용 값: `auto` 또는 `none`
sgd.l1RegularizationAmount	배정밀도 실수	0(기본적으로 L1은 사용되지 않음)	허용 범위: 0~MAX_DOUBLE 1E-4와 1E-8 사이의 L1 값은 양호한 결과를 생성하는 것으로 확인되었습니다. 값이 클수록 그다지 유용하지 않은 모델이 생성될 수 있습니다. L1과 L2를 모두 설정할 수는 없습니다. 두 선택기 중 하나만 선택해야 합니다.
sgd.l2RegularizationAmount	배정밀도 실수	1E-6(기본적으로 이 정도의 정규화	허용 범위: 0 ~ MAX_DOUBLE 1E-2 및 1E-6 사이의 L2 값은 양호한 결과를 생성하는 것으로 확인되었습

학습 파라미터	유형	기본 값	설명
		에는 L2가 사용됨)	니다. 값이 클수록 그다지 유용하지 않은 모델이 생성될 수 있습니다. L1과 L2를 모두 설정할 수는 없습니다. 두 선택기 중 하나만 선택해야 합니다.

ML 모델 생성

데이터 소스를 생성하고 나면 ML 모델을 생성할 준비가 된 것입니다. Amazon Machine Learning 콘솔을 사용하여 모델을 생성하는 경우 기본 설정을 사용하거나 사용자 지정 옵션을 적용하여 모델을 사용자 지정할 수 있습니다.

사용자 지정 옵션에는 다음이 포함됩니다.

- 평가 설정: Amazon ML이 ML 모델의 예측 품질을 평가하기 위해 입력 데이터의 일부를 예약하도록 선택할 수 있습니다. 평가에 대한 자세한 내용은 ML 모델 평가 단원을 참조하세요.
- 레시피: 레시피는 Amazon ML에 모델 학습에 사용할 수 있는 속성 및 속성 변환을 알려줍니다. ML 레시피에 대한 자세한 내용은 데이터 레시피를 사용한 특성 변환 단원을 참조하세요.
- 학습 파라미터: 파라미터는 학습 프로세스 및 결과 ML 모델의 특정 속성을 제어합니다. 학습 파라미터에 대한 자세한 내용은 학습 파라미터 단원을 참조하세요.

이러한 설정의 값을 선택하거나 지정하려면 ML 모델 생성 마법사를 사용할 때 사용자 지정 옵션을 선택합니다. ML에서 기본 설정을 적용하도록 하려면 기본값을 선택합니다.

ML 모델을 생성할 때 Amazon ML은 대상 속성의 속성 유형에 따라 사용할 학습 알고리즘 유형을 선택합니다. (대상 속성은 "정답"이 포함된 속성입니다.) 대상 속성이 이진인 경우 Amazon ML은 로지스틱 회귀 알고리즘을 사용하는 바이너리 분류 모델을 생성합니다. 대상 속성이 범주형인 경우 Amazon ML은 다항 로지스틱 회귀 알고리즘을 사용하는 다중 클래스 모델을 생성합니다. 대상 속성이 숫자인 경우 Amazon ML은 선형 회귀 알고리즘을 사용하는 회귀 모델을 생성합니다.

주제

- 사전 조건
- 기본 옵션을 사용하여 ML 모델 생성
- 사용자 지정 옵션을 사용하여 ML 모델 생성

사전 조건

Amazon ML 콘솔을 사용하여 ML 모델을 생성하기 전에 두 개의 데이터 소스를 생성해야 하는데, 하나는 모델 학습용이고 다른 하나는 모델 평가용입니다. 아직 데이터소 스를 두 개 생성하지 않은 경우 자습서의 2단계: 학습 데이터 세트 생성 단원을 참조하세요.

기본 옵션을 사용하여 ML 모델 생성

ML에서 다음을 수행하려면 기본값 옵션을 선택합니다.

- 입력 데이터를 분할하여 처음 70%는 학습에 사용하고 나머지 30%는 평가에 사용합니다.
- 학습 데이터 소스에서 수집한 통계(입력 데이터 소스의 70%)를 기반으로 레시피를 추천합니다.
- 기본 학습 파라미터 선택

기본 옵션을 선택하려면

1. ML 콘솔에서 머신 러닝을 선택한 다음 ML 모델을 선택합니다.
2. ML 모델 요약 페이지에서 새 ML 모델 생성을 선택합니다.
3. 입력 데이터 페이지에서 S3 데이터를 가리키는 데이터 소스를 이미 생성했습니다가 선택되었는지 확인합니다.
4. 표에서 데이터 소스를 선택한 다음 계속을 선택합니다.
5. ML 모델 설정 페이지의 ML 모델 이름에서 ML 모델의 이름을 입력합니다.
6. 학습 및 평가 설정에 대해 기본값이 선택되었는지 확인합니다.
7. 이 평가의 이름 지정에서 평가 이름을 입력한 다음 검토를 선택합니다. ML이 마법사의 나머지 부분을 생략하고 검토 페이지로 이동합니다.
8. 데이터를 검토하고, 모델 및 평가에 적용하지 않으려는 데이터 소스에서 복사해온 태그를 모두 삭제한 다음 완료를 선택합니다.

사용자 지정 옵션을 사용하여 ML 모델 생성

ML 모델을 사용자 지정하면 다음을 수행할 수 있습니다.

- 나만의 레시피를 제공. 자체 레시피를 제공하는 방법에 대한 자세한 내용은 레시피 형식 참조 단원을 참조하세요.
- 학습 파라미터 선택 학습 파라미터에 대한 자세한 내용은 학습 파라미터 단원을 참조하세요.

- 기본 70/30 비율 이외의 학습/평가 분할 비율을 선택하거나 평가를 위해 이미 준비한 다른 데이터 소스를 제공합니다. 데이터 분리에 대한 자세한 내용은 데이터 분할 단원을 참조하세요.

이들 설정의 기본 값을 선택할 수도 있습니다.

기본 옵션을 사용하여 이미 모델을 생성한 상태에서 모델의 예측 성능을 향상시키려면 사용자 지정 옵션을 사용하여 몇 가지 사용자 지정 설정의 새 모델을 생성합니다. 예를 들어 레시피에 특성 변환을 더 추가하거나 학습 파라미터의 전달 수를 늘릴 수 있습니다.

사용자 지정 옵션을 사용하여 모델을 생성하려면

1. ML 콘솔에서 머신 러닝을 선택한 다음 ML 모델을 선택합니다.
2. ML 모델 요약 페이지에서 새 ML 모델 생성을 선택합니다.
3. 데이터 소스를 이미 생성한 경우 입력 데이터 페이지에서 S3 데이터를 가리키는 데이터 소스를 이미 생성했습니다를 선택합니다. 표에서 데이터 소스를 선택한 다음 계속을 선택합니다.

   데이터 소스를 생성해야 하는 경우 내 데이터가 S3에 있고 데이터 소스를 생성해야 합니다를 선택하고 계속을 선택합니다. 데이터 소스 생성 마법사로 리디렉션됩니다. 데이터가 S3에 있는지 아니면 Redshift에 있는 지를 지정한 다음 확인을 선택합니다. 데이터 소스 생성 절차를 완료합니다.

   데이터 소스를 만든 후에는 ML 모델 생성 마법사의 다음 단계로 리디렉션됩니다.
4. ML 모델 설정 페이지의 ML 모델 이름에서 ML 모델 이름을 입력합니다.
5. 학습 및 평가 설정 선택에서 사용자 지정을 선택한 다음 계속을 선택합니다.
6. 레시피 페이지에서 customize a recipe를 할 수 있습니다. 레시피를 사용자 지정하지 않으려면 Amazon ML이 자동으로 추천합니다. 계속을 선택합니다.
7. 고급 설정 페이지에서 최대 ML 모델 크기, 최대 데이터 전달 횟수, 학습 데이터의 셔플 유형, 정규화 유형 및 정규화 정도를 지정합니다. 이들을 지정하지 않으면 Amazon ML은 기본 학습 파라미터를 사용합니다.

   이러한 파라미터 및 기본값에 대한 자세한 내용은 학습 파라미터 단원을 참조하세요.

   계속을 선택합니다.
8. 평가 페이지에서 ML 모델을 즉시 평가할지 여부를 지정합니다. 지금 ML 모델을 평가하지 않으려면 검토를 선택합니다.

   지금 ML 모델을 평가하려면:

   a. 이 평가 이름 지정에서 평가 이름을 입력합니다.

b. 평가 데이터 선택에 대해 ML에서 평가를 위해 입력 데이터의 일부를 예약할지, 예약할 경우 데이터 소스를 분할할지, 아니면 평가를 위해 다른 데이터 소스를 제공할지 선택합니다.

c. 검토를 선택합니다.

9. 검토 페이지에서 선택 내용을 편집하고 데이터 소스에서 복사한 태그 중 모델 및 평가에 적용하지 않으려는 태그를 모두 삭제한 다음 완료를 선택합니다.

모델을 생성한 후에는 4단계: ML 모델의 예측 성능 검토 및 점수 임계값 설정 단원을 참조하세요.

기계 학습을 위한 데이터 변환

기계 학습 모델의 성능은 모델을 학습시키는 데 사용되는 데이터에 비례합니다. 우수한 학습 데이터의 주요 특징은 학습 및 일반화에 최적화된 방식으로 제공된다는 것입니다. 이 최적의 형식으로 데이터를 모으는 프로세스를 업계에서는 특성 변환이라고 합니다.

주제

특성 변환의 중요성

신용 카드 거래가 사기인지 아닌 지를 결정하는 역할을 하는 기계 학습 모델을 생각해 보세요. 애플리케이션 배경 지식과 데이터 분석을 바탕으로 입력 데이터에 포함시켜야 할 중요한 데이터 필드(또는 특성)를 결정할 수 있습니다. 예를 들어 거래 금액, 판매자 이름, 주소, 신용 카드 소유자 주소 등을 학습 프로세스에 제공하는 것이 중요합니다. 반면, 임의로 생성된 거래 ID는 정보를 담고 있지 않으며 (실제로 무작위임을 알고 있는 경우) 유용하지 않습니다.

어떤 필드를 포함시킬지 결정했으면 이들 특성을 학습 프로세스에 도움이 되도록 변환합니다. 변환은 입력 데이터에 배경 경험을 추가하여 기계 학습 모델이 이러한 경험을 활용할 수 있도록 합니다. 예를 들어 다음과 같은 판매자 주소는 문자열로 표시됩니다.

"123 Main Street, Seattle, WA 98101"

주소 자체로는 표현력이 제한적이므로 정확한 주소와 관련된 패턴을 학습하는 데만 유용합니다. 그러나 주소를 구성 요소로 나누면 "주소"(123 Main Street), "도시"(Seattle), "주"(WA) 및 "우편번호"(98101)와 같은 추가 특성을 생성할 수 있습니다. 이제 학습 알고리즘을 통해 서로 다른 거래를 그룹화하여 더 광범위한 패턴을 발견할 수 있습니다. 아마도 일부 판매자의 우편번호는 다른 판매자에 비해 사기 행위가 더 심할 수 있습니다.

특성 변환 접근 방식 및 프로세스에 대한 자세한 내용은 기계 학습 개념 단원을 참조하세요.

데이터 레시피를 사용한 특성 변환

Amazon ML로 ML 모델을 생성하기 전에 특성을 변환하는 두 가지 방법이 있습니다. Amazon ML에 표시하기 전에 입력 데이터를 직접 변환하거나 Amazon ML의 내장된 데이터 변환을 사용할 수 있습니다. 일반적인 변환에 대해 미리 형식이 지정된 지침인 Amazon ML 레시피를 사용할 수 있습니다. 레시피를 통해 다음 작업을 할 수 있습니다.

- 내장된 일반 기계 학습 변환 목록에서 선택하여 이를 개별 변수 또는 변수 그룹에 적용
- 기계 학습 프로세스에 사용할 수 있는 입력 변수와 변환을 선택

Amazon ML 레시피를 사용하면 여러 가지 이점이 있습니다. Amazon ML이 자동으로 데이터 변환을 수행하므로 사용자가 직접 구현하지 않아도 됩니다. 또한 Amazon ML이 입력 데이터를 읽는 동안 변환을 적용하고 결과를 디스크에 저장하는 중간 단계 없이 학습 프로세스에 결과를 제공하기 때문에 속도가 빠릅니다.

레시피 형식 참조

Amazon ML 레시피에는 기계 학습 프로세스의 일부로 데이터를 변환하기 위한 지침이 포함되어 있습니다. 레시피는 JSON과 유사한 구문을 사용하여 정의되지만 일반적인 JSON 제한 외에 추가 제한이 있습니다. 레시피에는 다음과 같은 섹션이 있으며, 해당 섹션은 다음과 같은 순서대로 표시되어야 합니다.

- 그룹을 사용하면 여러 변수를 그룹화하여 변환을 쉽게 적용할 수 있습니다. 예를 들어 웹 페이지의 자유 텍스트 부분(제목, 본문)과 관련된 모든 변수를 그룹으로 만든 다음 이들 부분을 모두 한 번에 변환할 수 있습니다.
- 할당을 사용하면 이름이 지정된 중간 변수를 만들 수 있는데, 이들은 처리 시 재활용이 가능합니다.
- 출력은 학습 프로세스에서 사용할 변수와 이러한 변수에 적용할 변환(있는 경우)을 정의합니다.

Groups

변수 그룹을 정의하여 그룹 내의 모든 변수를 일괄하여 변환하거나 이러한 변수를 변환하지 않고 기계 학습에 사용할 수 있습니다. 기본적으로 Amazon ML은 다음과 같은 그룹을 생성합니다.

ALL_TEXT, ALL_NUMERIC, ALL_CATEGORICAL, ALL_BINARY – 데이터 소스 스키마에 정의된 변수를 기반으로 하는 유형별 그룹.

> **Note**
>
> ALL_INPUTS로는 그룹을 생성할 수 없습니다.

이들 변수는 정의하지 않고도 레시피의 출력 섹션에서 사용할 수 있습니다. 기존 그룹에서 변수를 더하거나 빼서 또는 변수 모음에서 직접 변수를 더하거나 빼서 사용자 지정 그룹을 만들 수도 있습니다. 다음 예제에서는 세 가지 접근 방식과 그룹화 할당의 구문을 모두 보여줍니다.

```
"groups": {

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

}
```

그룹 이름은 영문자로 시작해야 하며 길이는 1~64자일 수 있습니다. 그룹 이름이 영문자로 시작하지 않거나 특수 문자(, ' " \t \r \n ( ) \)가 포함된 경우 레시피에 포함시키려면 이름을 따옴표로 묶어야 합니다.

할당

편의성과 가독성을 위해 중간 변수에 한 가지 이상의 변환을 할당할 수 있습니다. 예를 들어 email_subject라는 텍스트 변수가 있고 여기에 소문자 변환을 적용한 경우 결과 변수의 이름을 email_subect_lowercase로 지정하면 레시피의 다른 곳에서 해당 변수를 쉽게 추적할 수 있습니다. 또한 할당을 체인으로 연결하여 여러 변환을 지정된 순서로 적용할 수도 있습니다. 다음 예제에서는 레시피 구문의 단일 및 체인 할당을 보여줍니다.

```
"assignments": {

"email_subject_lowercase": "lowercase(email_subject)",

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

}
```

중간 변수 이름은 영문자로 시작해야 하며 길이는 1~64자일 수 있습니다. 이름이 영문자로 시작하지 않거나 특수 문자(, ' " \t \r \n ( ) \)가 포함된 경우에는 이름을 따옴표로 묶어야 레시피에 포함될 수 있습니다.

결과

출력 섹션은 학습 프로세스에 사용할 입력 변수와 해당 변수에 적용할 변환을 제어합니다. 출력 섹션이 비어 있거나 존재하지 않는 경우 학습 프로세스에 데이터가 전달되지 않으므로 오류가 발생합니다.

가장 간단한 출력 섹션에는 사전 정의된 ALL_INPUTS 그룹이 포함되어 있는데, 이는 ML이 데이터 소스에 정의된 모든 변수를 학습에 사용하도록 지시합니다.

```
"outputs": [

"ALL_INPUTS"

]
```

Amazon ML에 다음 그룹의 모든 변수를 사용하도록 지시하여 출력 섹션에서 미리 정의된 다른 그룹을 참조할 수도 있습니다.

```
"outputs": [

"ALL_NUMERIC",

"ALL_CATEGORICAL"

]
```

출력 섹션은 사용자 지정 그룹도 참조할 수 있습니다. 다음 예제에서는 이전 예제의 그룹화 할당 섹션에 정의된 사용자 지정 그룹 중 하나만 기계 학습에 사용됩니다. 다른 모든 변수는 삭제됩니다.

```
"outputs": [

"All_Categorical_plus_one_other"

]
```

출력 섹션은 할당 섹션에 정의된 변수 할당도 참조할 수 있습니다.

```
"outputs": [

"email_subject_lowercase"

]
```

그리고 입력 변수 또는 변환을 출력 섹션에서 직접 정의할 수 있습니다.

```
"outputs": [

"var1",

"lowercase(var2)"

]
```

출력은 학습 프로세스에서 사용할 수 있을 것으로 예상되는 모든 변수와 변환된 변수를 명시적으로 지정해야 합니다. 예를 들어 var1과 var2의 데카르트 곱(Cartesian product)을 출력에 포함시킨다고 가정해 보겠습니다. 원시 변수 var1과 var2도 모두 포함시키려면 출력 섹션에 원시 변수를 추가해야 합니다.

```
"outputs": [

"cartesian(var1,var2)",

"var1",

"var2"

]
```

가독성을 높이기 위해 변수와 함께 주석 텍스트를 추가하여 출력에 주석을 포함시킬 수 있습니다.

```
"outputs": [

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

"age // explicitly include the original numeric variable along with the
```

```
binned version"

]
```

출력 섹션 내에서 이러한 접근 방식을 모두 혼합하여 사용할 수 있습니다.

> **Note**
> 레시피를 추가할 때는 Amazon ML 콘솔에서 댓글을 달 수 없습니다.

전체 레시피 예제

다음 예제는 이전 예제에서 소개된 여러 내장 데이터 프로세서를 참조합니다.

```
{

"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"

]

}
```

제안된 레시피

Amazon ML에서 새 데이터 소스를 생성하고 해당 데이터 소스에 대한 통계를 계산하면 Amazon ML이 데이터 소스에서 새 ML 모델을 생성하는 데 사용할 수 있는 추천 레시피도 생성합니다. 추천 데이터 소스는 데이터와 이 데이터에 존재하는 대상 속성을 기반으로 하며, ML 모델을 생성하고 미세 조정하는 데 유용한 출발점을 제공합니다.

Aazon ML 콘솔에서 제안된 레시피를 사용하려면 새로 만들기 드롭다운 목록에서 데이터 스스 또는 데이터 소스 및 ML 모델을 선택합니다. ML 모델 설정에 대해서는 ML 모델 생성 마법사의 ML 모델 설정 단계에서 기본 또는 사용자 지정 학습 및 평가 설정을 선택하면 됩니다. 기본 옵션을 선택하면 Amazon ML에서 자동으로 제안된 레시피를 사용합니다. 사용자 지정 옵션을 선택하면 다음 단계의 레시피 편집기에 제안된 레시피가 표시되며 필요에 따라 이를 확인하거나 수정할 수 있습니다.

> **Note**
>
> Amazon ML에서는 통계 계산이 완료되기 전에 데이터 소스를 생성한 다음 즉시 이를 사용하여 ML 모델을 생성할 수 있습니다. 이 경우 사용자 지정 옵션에서 제안된 레시피를 볼 수 없지만 해당 단계를 계속 진행하여 Amazon ML에서 모델 학습에 기본 레시피를 사용하도록 할 수는 있습니다.

Amazon ML API에서 제안된 레시피를 사용하려면 레시피와 RecipeURI API 파라미터 모두에 빈 문자열을 전달하면 됩니다. Amazon ML API를 사용하여 제안된 레시피를 검색하는 것은 불가능합니다.

데이터 변환 참조

주제

n-gram 변환

n-gram 변환은 텍스트 변수를 입력으로 받아 (사용자가 구성할 수 있는) n개 단어의 창을 슬라이딩하는 것에 해당하는 문자열을 생성하여 프로세스에서 출력을 생성합니다. 예를 들어 "I really enjoyed reading this book"라는 텍스트 문자열을 생각해 보세요.

창 크기=1로 n-gram 변환을 지정하면 간단히 해당 문자열에 있는 모든 개별 단어를 얻을 수 있습니다.

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

창 크기 =2로 n-gram 변환을 지정하면 두 단어 조합과 한 단어 조합이 모두 제공됩니다.

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

창 크기 = 3으로 n-gram 변환을 지정하면 이 목록에 세 단어 조합이 추가되어 다음과 같은 결과가 나타납니다.

```
{"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"}
```

크기가 2~10단어 범위인 n-gram을 요청할 수 있습니다. 크기가 1인 n-gram은 데이터 스키마에서 유형이 텍스트로 표시된 모든 입력에 대해 암시적으로 생성되므로 요청하지 않아도 됩니다. 마지막으로, n-gram은 공백 문자의 입력 데이터를 분리하여 생성된다는 점을 명심하세요. 즉, 예를 들어 문장 부호 문자는 워드 토큰의 일부로 간주됩니다. "red, green, blue" 문자열에 대해 창 크기가 2인 n-gram을 생성하면 {"red,", "green,", "blue,", "red, green", "green, blue"}가 출력됩니다. 필요하지 않은 경우 구두점 제거 프로세서(이 문서의 뒷부분에서 설명)를 사용하여 구두점 기호를 제거할 수 있습니다.

변수 var1에 대해 창 크기가 3인 n-gram을 계산하려면:

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

Orthogonal Sparse Bigram(OSB) 변환

OSB 변환은 텍스트 문자열 분석을 지원하기 위한 것으로 bi-gram 변환(창 크기가 2인 n-gram)의 대안입니다. OSB는 크기가 n인 창을 텍스트 위로 슬라이딩하고 창의 첫 번째 단어가 포함된 모든 단어 쌍을 출력하면 생성됩니다.

각 OSB를 빌드할 때 OSB를 구성하는 단어를 "_" (밑줄) 문자로 연결하고 생략된 모든 토큰은 OSB에 밑줄을 하나 더 추가하여 표시합니다. 따라서 OSB는 창에 표시되는 토큰 뿐만 아니라 동일한 창 내에서 건너뛰었던 토큰 수를 나타내는 값도 인코딩합니다.

예를 들어, "The quick brown fox jumps over the lazy dog"라는 문자열과 크기가 4인 OSB를 생각해 보세요. 다음 예제에는 네 단어로 된 창 여섯 개와 문자열 끝에서 나온 마지막 두 개의 짧은 창, 그리고 각 창에서 생성된 OSB도 나와 있습니다.

Window, {생성된 OSB}

```
"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}
```

Orthogonal sparse bigram은 n-gram의 대안으로, 일부 상황에서는 더 잘 작동할 수 있습니다. 데이터에 큰 텍스트 필드(단어가 10개 이상)가 있는 경우 어떤 것이 더 효과적인지 실험해 보세요. 큰 텍스트 필드를 구성하는 요소는 상황에 따라 달라질 수 있습니다. 그러나 텍스트 필드가 클수록 OSB는 특수 건너뛰기 기호(밑줄)로 인해 텍스트를 고유하게 나타내는 것으로 경험적으로 나타났습니다.

입력 텍스트 변수에 대한 OSB 변환의 경우 창 크기를 2~10으로 요청할 수 있습니다.

변수 var1에 대해 창 크기가 5인 OSB를 계산하려면:

"osb(var1, 5)"

소문자 변환

소문자 변환 프로세서는 텍스트 입력을 소문자로 변환합니다. 예를 들어 "The Quick Brown Fox Jumps Over the Lazy Dog"라고 입력하면 프로세서가 "the quick brown fox jumps over the lazy dog"라고 출력합니다.

var1 변수에 소문자 변환을 적용하려면:

"lowercase(var1)"

구두점 제거 변환

Amazon ML은 암시적으로 데이터 스키마에 텍스트로 표시된 입력을 공백으로 분할합니다. 문자열에 있는 구두점은 주변의 공백에 따라 인접한 단어 토큰이 되거나 완전히 별도의 토큰이 됩니다. 이것이 바람직하지 않은 경우 구두점 제거 변환을 사용하여 생성된 특성에서 구두점 기호를 제거할 수 있습니다. 예를 들어, "Welcome to AML - please fasten your seat-belts!" 라는 문자열이 주어진다면 다음과 같은 토큰 세트가 암시적으로 생성됩니다.

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

이 문자열에 구두점 제거 프로세서를 적용하면 다음과 같은 결과가 나옵니다.

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

접두사 및 접미사 문장 부호만 제거된다는 점에 유의하세요. 토큰 중간에 나타나는 문장 부호(예: "seat-belts"의 하이픈)는 제거되지 않습니다.

변수 var1에 구두점 제거를 적용하려면:

"no_punct(var1)"

Quantile Binning 변환

quantile binning 프로세서는 숫자 변수 및 bin 숫자라는 파라미터의 두 가지 입력을 받아서 범주형 변수를 출력하는 프로세스입니다. 관측된 값을 그룹화하여 변수 분포의 비선형성을 발견하기 위한 것입니다.

많은 경우에 숫자 변수와 대상 간의 관계는 선형적이지 않습니다(숫자 변수 값이 대상에 따라 점차 증가하거나 감소하지 않음). 이 경우 다양한 범위의 숫자 특성을 나타내는 범주형 특성으로 숫자 특성을 비닝하는 것이 유용할 수 있습니다. 각 범주형 특성 값(빈)은 대상과의 선형 관계를 갖도록 모델링될 수 있습니다. 예를 들어 연속적인 숫자 특성 account_age가 책을 구입할 가능성과 선형적으로 관련이 없다는 것을 사용자가 알고 있다고 가정하고 설명해 보겠습니다. 사용자는 대상과의 관계를 더욱 정확하게 캡처할 수 있는 범주형 특성으로 age를 비닝할 수 있습니다.

quantile binning 프로세서를 사용하면 age 변수의 모든 입력 값 분포를 기반으로 동일한 크기의 n개 빈을 설정한 다음 각 숫자를 빈이 포함된 텍스트 토큰으로 대체하도록 Amazon ML에 지시할 수 있습니다. 숫자 변수에 대한 최적의 빈 수는 변수의 특징과 대상과의 관계에 따라 달라지며, 이는 실험을 통해 가장 효과적으로 결정됩니다. ML은 제안된 레시피의 데이터 통계를 기반으로 숫자 특성에 가장 적합한 빈 수를 추천합니다.

모든 숫자 입력 변수에 대해 5~1000개의 분위수 빈을 계산하도록 요청할 수 있습니다.

다음 예제에서는 숫자 변수 var1 대신 50개의 빈을 계산하고 사용하는 방법을 보여줍니다.

"quantile_bin(var1, 50)"

정규화 변환

정규화 변환기는 평균이 0이고 분산이 1이 되도록 숫자 변수를 정규화합니다. 대상과 관련된 정보를 제공하는 특성이든 아니든 상관없이 규모가 가장 큰 변수가 ML 모델을 지배할 수 있기 때문에 숫자 변수를 정규화하면 학습 프로세스에 도움이 될 수 있습니다.

이 변환을 숫자 변수 var1에 적용하려면 레시피에 다음을 추가합니다.

normalize(var1)

이 변환기는 또한 모든 숫자 변수에 대해 사용자 정의 숫자 변수 그룹 또는 미리 정의된 그룹(ALL_NUMERIC)을 입력으로 사용할 수 있습니다.

normalize(ALL_NUMERIC)

참고

숫자 변수에 정규화 프로세서를 반드시 사용해야 하는 것은 아닙니다.

Cartesian Product 변환

데카르트 변환은 둘 이상의 텍스트 또는 범주형 입력 변수의 순열을 생성합니다. 이 변환은 변수 간 상호작용이 의심되는 경우에 사용됩니다. 예를 들어 자습서: Amazon ML을 사용하여 마케팅 제안에 대한 응답 예측에서 사용되는 은행 마케팅 데이터 세트를 예로 들어 보겠습니다. 이 데이터 세트를 사용하여 경제 및 인구 통계 정보를 기반으로 은행 프로모션에 긍정적인 반응을 보일지 예측하고자 합니다. 우리는 그 사람의 직업 유형이 어느 정도 중요하다고 의심할 수 있으며(아마도 특정 분야에 취업하는 것과 가용 자금을 확보하는 것 사이에 상관 관계가 있을 수 있음), 최고 수준의 교육을 받은 것도 중요하다고 생각할 수 있습니다. 또한 이 두 변수의 상호작용에 강한 신호가 있다는 것을 더 깊이 직감할 수 있을 것입니다. 예를 들어 대학 학위를 취득한 기업가 고객에게 특히 적합하다는 것을 예로 들 수 있습니다.

Cartesian Product 변환은 범주형 변수 또는 텍스트를 입력으로 사용하여 이러한 입력 변수 간의 상호작용을 캡처하는 새로운 특성을 생성합니다. 특히, 각 학습 예제에 대해 기능 조합을 만들어 독립형 특성으로 추가합니다. 예를 들어, 단순화된 입력 행이 다음과 같다고 가정해 보겠습니다.

target, education, job

0, university.degree, technician

0, high.school, services

1, university.degree, admin

데카르트 변환을 범주형 변수인 education 및 직무 필드에 적용하도록 지정하면 education_job_interaction이라는 결과 특성이 다음과 같이 표시됩니다.

target, education_job_interaction

0, university.degree_technician

0, high.school_services

1, university.degree_admin

데카르트 변환은 토큰 시퀀스를 작업할 때 훨씬 더 강력합니다. 인수 중 하나가 암시적 또는 명시적으로 토큰으로 분할된 텍스트 변수인 경우처럼 말이죠. 예를 들어, 책을 교과서로 분류하고 교과서가 아닌 것으로 분류하는 작업을 생각해 보세요. 직관적으로 우리는 책 제목에 교과서임을 알려주는 무언가가 있다고 생각할 수도 있고(교과서 제목에 특정 단어가 더 자주 등장할 수도 있음) 책의 제본용 표지에 예측 가능한 부분이 있다고 생각할 수도 있지만(교과서는 양장본일 가능성이 더 높음), 실제로 가장 예측 가능한 것은 제목과 제본용 표지의 일부 단어의 조합입니다. 실제 예제의 경우 다음 표는 입력 변수 binding과 title에 데카르트 프로세서를 적용한 결과를 보여줍니다.

TextI	Title	Binding	Cartesian product of no_punct(Title) and Binding
1	Economics : Principles, Problems, Policies	Hardcover	{"Economics_Hardcover", "Principles_Hardcover", "Problems_Hardcover", "Policies_Hardcover"}
0	The Invisible Heart: An Economics Romance	Softcover	{"The_Softcover", "Invisible_Softcover", "Heart_Softcover", "An_Softcover", "Economics_Softcover", "Romance_Softcover"}
0	Fun With Problems	Softcover	{"Fun_Softcover", "With_Softcover", "Problems_Softcover"}

다음 예제에서는 var1과 var2에 데카르트 변환기를 적용하는 방법을 보여줍니다.

cartesian(var1, var2)

데이터 재배열

데이터 재배열 특성을 사용하면 입력 데이터가 가리키는 일부 데이터만 기반으로 하는 데이터 소스를 만들 수 있습니다. 예를 들어 ML 콘솔에서 ML 모델 생성 마법사를 사용하여 ML 모델을 생성하고 기본 평가 옵션을 선택하면 ML이 자동으로 데이터의 30%를 ML 모델 평가를 위해 예약하고 나머지 70%는 학습에 사용합니다. 이 기능은 Amazon ML의 데이터 재배열 특성을 통해 활성화됩니다.

Amazon ML API를 사용하여 데이터 소스를 생성하는 경우 새 데이터 소스의 기반이 되는 입력 데이터 부분을 지정할 수 있습니다. `DataRearrangement` 파라미터의 지침을 `CreateDataSourceFromS3`, `CreateDataSourceFromRedshift` 또는 `CreateDataSourceFromRDS` API에 전달하면 됩니다 DataRearrangement 문자열의 내용은 데이터의 시작 위치와 끝 위치를 포함하고 있는 JSON 문자열

로, 백분율, 보완 플래그 및 분할 전략으로 표현됩니다. 예를 들어, 다음 데이터 재배열 문자열은 데이터의 처음 70%가 데이터 소스를 만드는 데 사용되도록 지정합니다.

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

데이터 재배열 파라미터

Amazon ML이 데이터 소스를 생성하는 방식을 변경하려면 다음 파라미터를 사용합니다.

PercentBegin(선택 사항)

percentBegin을 사용하면 데이터 소스의 데이터가 시작되는 위치를 지정할 수 있습니다. percentBegin 및 percentEnd를 포함시키지 않으면 ML은 데이터 소스를 생성할 때 모든 데이터를 포함시킵니다.

유효한 값은 0 ~ 100(포함)입니다.

PercentEnd(선택 사항)

percentEnd를 사용하면 데이터 소스의 데이터가 끝나는 위치를 지정할 수 있습니다. percentBegin 및 percentEnd를 포함시키지 않으면 ML은 데이터 소스를 생성할 때 모든 데이터를 포함시킵니다.

유효한 값은 0 ~ 100(포함)입니다.

Complement(선택 사항)

complement 파라미터는 ML에 percentBegin ~ percentEnd의 범위에 포함되지 않은 데이터를 사용하여 데이터 소스를 생성하도록 지시합니다. 이 complement 파라미터는 학습 및 평가를 위한 보완적인 데이터 소스를 생성해야 하는 경우에 유용합니다. 보완적인 데이터 소스를 만들려면 complement 파라미터와 함께 percentBegin 및 percentEnd에 동일한 값을 사용합니다.

예를 들어, 다음 두 데이터 소스는 데이터를 공유하지 않으므로 모델을 학습하고 평가하는 데 사용할 수 있습니다. 첫 번째 데이터 소스에는 데이터의 25%가 있고, 두 번째 데이터 소스에는 75%가 있습니다.

평가용 데이터소스:

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

학습용 데이터 소스:

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

유효 값은 `true`및 `false`입니다.

Strategy(선택 사항)

ML이 데이터 소스의 데이터를 분할하는 방식을 변경하려면 `strategy` 파라미터를 사용합니다.

`strategy` 파라미터의 기본값은 `sequential`입니다. 즉, ML이 데이터 소스의 `percentBegin` 및 `percentEnd` 파라미터 사이에 있는 모든 데이터 레코드를 입력 데이터에 나타나는 순서대로 가져옵니다.

다음 두 `DataRearrangement` 줄은 순차적으로 정렬된 학습 및 평가 데이터 소스의 예입니다.

평가용 데이터 소스: `{"splitting":{"percentBegin":70, "percentEnd":100, "strategy":"sequential"}}`

학습용 데이터 소스: `{"splitting":{"percentBegin":70, "percentEnd":100, "strategy":"sequential", "complement":"true"}}`

무작위로 선택한 데이터에서 데이터 소스를 만들려면 `strategy` 파라미터를 `random`로 설정하고 무작위 데이터 분할의 시드 값으로 사용되는 문자열을 입력합니다. 예를 들어 데이터에 대한 S3 경로를 무작위 시드 문자열로 사용할 수 있습니다. 무작위 분할 전략을 선택하면 ML이 각 데이터 행에 유사 난수를 할당한 다음, `percentBegin` ~ `percentEnd` 사이의 숫자가 할당된 행을 선택합니

다. 바이트 오프셋을 시드로 사용하여 유사 난수를 할당하므로 데이터를 변경하면 분할이 달라집니다. 기존 순서는 모두 보존됩니다. 무작위 분할 전략을 사용하면 학습 데이터와 평가 데이터의 변수가 비슷하게 분포됩니다. 입력 데이터에 암시적인 정렬 순서가 있어서 학습 및 평가 데이터 소스에 유사하지 않은 데이터 레코드가 포함되는 경우에 유용합니다.

다음 두 `DataRearrangement` 줄은 순차적으로 정렬되지 않은 학습 및 평가 데이터 소스의 예입니다.

평가용 데이터 소스:

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

학습용 데이터 소스:

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

유효 값은 `sequential`및 `random`입니다.

(선택 사항) Strategy:RandomSeed

ML이 randomSeed를 사용하여 데이터를 분할합니다. API의 기본 시드는 빈 문자열입니다. 무작위 분할 전략의 시드를 지정하려면 문자열을 전달합니다. 무작위 시드에 대한 자세한 내용은 머신 러닝 개발자 안내서의 데이터 무작위 분할 단원을 참조하세요.

ML에서 교차 검증을 사용하는 방법을 보여주는 샘플 코드는 Github 기계 학습 샘플 단원을 참조하세요.

ML 모델 평가

항상 모델을 평가하여 새 데이터 및 미래 데이터에 대한 대상 예측에 적합한지 판단해야 합니다. 미래 인스턴스에는 알 수 없는 대상 값이 있으므로 이미 대상 답변을 알고 있는 데이터에 대해 ML 모델의 정확도 지표를 확인하고 이 평가를 미래 데이터에 대한 예측 정확도를 위한 프록시로 사용해야 합니다.

모델을 제대로 평가하려면 학습 데이터 소스의 대상(실측 자료)으로 레이블이 지정된 데이터 샘플을 추출해야 합니다. 학습에 사용된 것과 동일한 데이터를 사용하여 모델을 평가하는 것은 유용하지 않은데, 이는 일반화하는 것과 대조적으로 학습 데이터를 "기억"할 수 있는 모델에 대한 보상을 하기 때문입니다. ML 모델 학습을 마치면 대상 값을 알고 있는 보류된 관측치를 모델에 전송합니다. 그런 다음 ML 모델에서 반환된 예측치를 알려진 대상 값과 비교합니다. 마지막으로 예측값과 실제 값이 얼마나 잘 일치하는지 알려주는 요약 지표를 계산합니다.

ML에서는 평가를 생성하여 ML 모델을 평가합니다. ML 모델에 대한 평가를 생성하려면 평가하려는 ML 모델과 학습에 사용되지 않은 레이블이 지정된 데이터가 필요합니다. 먼저 보류된 데이터로 Amazon ML 데이터 소스를 생성하여 평가용 데이터 소스를 생성합니다. 평가에 사용되는 데이터는 학습에 사용된 데이터와 동일한 스키마를 가져야 하며 대상 변수의 실제 값을 포함하고 있어야 합니다.

모든 데이터가 단일 파일 또는 디렉터리에 있는 경우 Amazon ML 콘솔을 사용하여 데이터를 분할할 수 있습니다. ML 모델 생성 마법사의 기본 경로는 입력 데이터 소스를 분할하여 처음 70%는 학습 데이터 소스에 사용하고 나머지 30%는 평가 데이터 소스에 사용합니다. ML 모델 생성 마법사의 사용자 지정 옵션을 사용하여 분할 비율을 사용자 지정할 수도 있습니다. 이 옵션에서는 학습에 사용할 무작위 70% 샘플을 선택하고 나머지 30%는 평가에 사용하도록 선택할 수 있습니다. 사용자 지정 분할 비율을 추가로 지정하려면 데이터 소스 생성 API의 데이터 재배열 문자열을 사용합니다. 평가 데이터 소스와 ML 모델이 있으면 평가를 만들고 평가 결과를 검토할 수 있습니다.

주제

ML 모델 인사이트 정보

ML 모델을 평가할 때 Amazon ML은 모델의 예측 정확도를 검토하기 위한 업계 표준 지표와 다양한 인사이트 정보를 제공합니다. Amazon ML의 평가 결과에는 다음이 포함됩니다.

- 모델의 전반적인 성공을 보고하기 위한 예측 정확도 지표
- 예측 정확도 지표를 넘어 모델의 정확도를 탐구하는 데 도움이 되는 시각화
- 점수 임계값 설정의 영향을 검토하는 기능(바이너리 분류에만 해당)
- 평가의 유효성을 확인하기 위한 기준에 대한 경보

지표 및 시각화의 선택은 평가 중인 ML 모델 유형에 따라 달라집니다. 이러한 시각화를 검토하여 모델이 비즈니스 요구 사항에 부합할 만큼 성능이 좋은지 확인하는 것이 중요합니다.

바이너리 모델 인사이트

예측 해석

많은 바이너리 분류 알고리즘의 실제 출력은 예측 점수입니다. 점수는 주어진 관측치가 긍정 클래스에 속한다는 시스템의 확실성을 나타냅니다(실제 목표 값은 1입니다). 바이너리 분류 모델은 0에서 1 범위의 점수를 출력합니다. 이 점수의 소비자는 관측치를 1로 분류할지 또는 0으로 분류할 지를 결정하기 위해 분류 임계값 또는 커트라인을 선택하여 점수를 해석하고 점수를 이 값과 비교합니다. 커트라인보다 점수가 높은 관측치는 대상= 1로 예측되고, 커트라인보다 점수가 낮으면 대상= 0으로 예측됩니다.

Amazon ML에서 기본 점수 커트라인은 0.5입니다. 비즈니스 요구 사항에 맞게 이 커트라인을 업데이트하도록 선택할 수 있습니다. 콘솔의 시각화를 사용하면 커트라인 선택이 애플리케이션에 어떤 영향을 미치는지 이해할 수 있습니다.

ML 모델 정확도 측정

Amazon ML은 (수신기 작동 특성) 곡선하면적(AUC)이라는 바이너리 분류 모델에 대한 업계 표준 정확도 지표를 제공하고 있습니다. AUC에서는 부정적인 사례보다 긍정적인 사례에 대해 더 높은 점수를 예측하는 모델의 기능을 측정합니다. 점수 커트라인과 무관하므로 임계값을 지정하지 않고도 AUC 지표에서 모델의 예측 정확도를 파악할 수 있습니다.

AUC 지표에서는 0 ~ 1의 십진수 값을 반환합니다. AUC 값이 1에 가까우면 정확성이 높은 ML 모델을 가리킵니다. 값이 0.5에 가까우면 무작위로 추측하는 것보다 나을 것이 없는 ML 모델을 가리킵니다. 0에 가까운 값은 보기 드문 것으로, 일반적으로 데이터에 문제가 있음을 나타냅니다. 기본적으로 AUC가 0에 가까우면 ML 모델이 올바른 패턴을 학습했지만 이를 사용하여 현실과 다른 예측을 하고 있다는 의

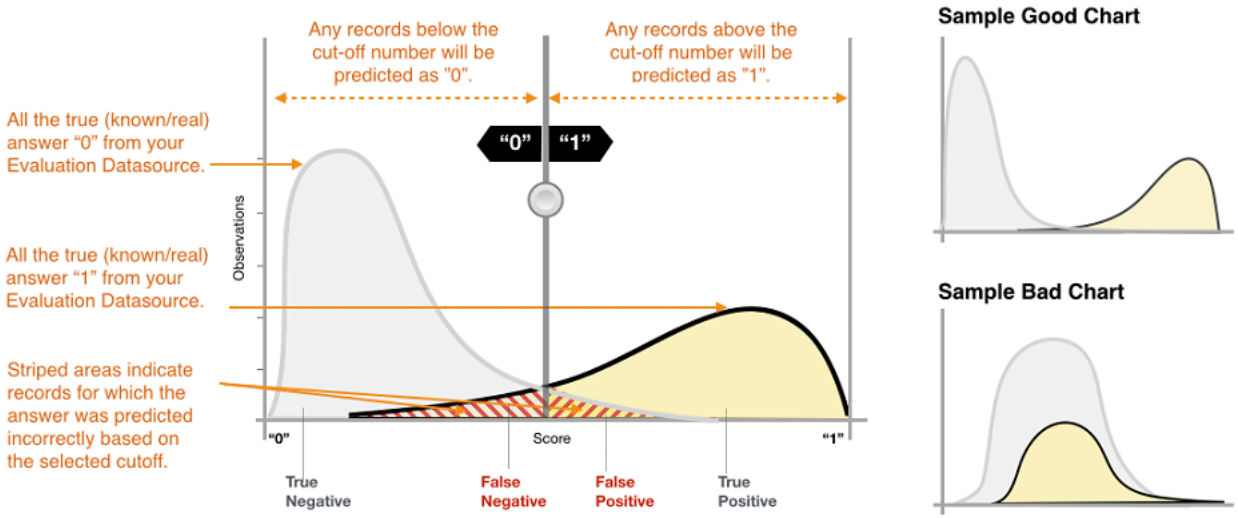
미입니다('0'이 '1'로 예측되거나 그 반대의 경우도 마찬가지임). AUC에 대한 자세한 내용은 Wikipedia의 수신기 작동 특성 페이지를 참조하세요.

바이너리 모델의 기준 AUC 지표는 0.5입니다. 이 값은 1 또는 0 답을 무작위로 예측하는 가상 ML 모델의 값입니다. 바이너리 ML 모델의 성능이 이 값보다 좋아야 가치를 인정받기 시작할 수 있습니다.

성능 시각화 사용

ML 모델의 정확성을 살펴보려면 ML 콘솔의 평가 페이지의 그래프를 검토하면 됩니다. 이 페이지에는 다음 두 가지 히스토그램이 나와 있습니다. a) 실제 긍정 점수 히스토그램(대상이 1)이고 다른 하나는 b) 평가 데이터의 실제 부정 점수 히스토그램(대상이 0)입니다.

예측 정확도가 양호한 ML 모델은 실제 1에 더 높은 점수를 예측하고 실제 0에 더 낮은 점수를 예측합니다. 완벽한 모델이라면 x축의 서로 다른 두 끝에 두 개의 히스토그램을 표시하여 실제 긍정은 모두 높은 점수를, 실제 부정은 모두 낮은 점수를 받았음을 보여줄 것입니다. 하지만 ML 모델도 실수를 하기 때문에 일반적인 그래프를 보면 두 히스토그램이 특정 점수에서 겹치는 것을 볼 수 있습니다. 성능이 매우 낮은 모델은 긍정 클래스와 부정 클래스를 구분하지 못하고 두 클래스 모두 히스토그램이 대부분 겹칩니다.



시각화를 사용하면 두 가지 유형의 올바른 예측과 두 가지 유형의 잘못된 예측에 해당하는 예측의 수를 확인할 수 있습니다.

올바른 예측

- 참 긍정(TP): Amazon ML이 값을 1로 예측했으며, 실제 값은 1입니다.
- 참 부정(TN): Amazon ML이 값을 0으로 예측했고, 실제 값은 0입니다.

잘못된 예측

- 거짓 긍정(FP): Amazon ML이 값을 1로 예측했지만 실제 값은 0입니다.
- 거짓 부정(FN): Amazon ML이 값을 0으로 예측했지만 실제 값은 1입니다.

> **Note**
>
> TP, TN, FP, FN의 수는 선택한 점수 임계값에 따라 달라지는데, 이러한 수치 중 하나를 최적화하려면 나머지 수치를 절충해야 합니다. 일반적으로 TP 수가 많으면 FP 수가 많아지고 TN 수는 낮아집니다.

점수 커트라인 조정

ML 모델은 수치형 예측 점수를 생성한 다음 커트라인을 적용하여 이 점수를 바이너리 0/1 레이블로 변환하는 방식으로 작동합니다. 점수 커트라인을 변경하여 실수가 발생했을 때의 모델 동작을 조정할 수 있습니다. ML 콘솔의 평가 페이지에서 다양한 점수 커트라인의 영향을 검토하고 모델에 사용하려는 점수 커트라인을 저장할 수 있습니다.

점수 커트라인 임계값을 조정할 때는 두 가지 오류 유형 간의 균형을 준수합니다. 커트라인을 왼쪽으로 이동하면 더 많은 참 긍정을 얻을 수 있지만 거짓 긍정 오류의 수가 증가한다는 단점이 있습니다. 오른쪽으로 이동하면 거짓 긍정 오류가 덜 발생하지만 일부 참 긍정 오류를 놓칠 수 있다는 단점이 있습니다. 예측 애플리케이션의 경우 적절한 커트라인 점수를 선택하여 어떤 종류의 오류가 더 허용되는지 결정할 수 있습니다.

고급 지표 검토

Amazon ML은 ML 모델의 예측 정확도를 측정하기 위해 정확도, 정밀도, 재현율, 거짓 긍정률과 같은 추가 지표를 제공합니다.

정확도

정확도(ACC)는 올바른 예측의 비율을 측정합니다. 범위는 0 ~ 1입니다. 값이 클수록 예측 정확도가 높음을 나타냅니다.

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

정밀도

정밀도는 긍정으로 예측되는 사례 중 실제 긍정의 비율을 측정합니다. 범위는 0 ~ 1입니다. 값이 클수록 예측 정확도가 더 높습니다.

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

재현율

재현율은 긍정으로 예측되는 실제 긍정의 비율을 측정합니다. 범위는 0 ~ 1입니다. 값이 클수록 예측 정확도가 더 높습니다.

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

거짓 긍정률

거짓 긍정률(FPR)은 거짓 경보율 또는 긍정으로 예측되는 실제 부정 비율을 측정합니다. 범위는 0 ~ 1입니다. 값이 작을수록 예측 정확도가 더 높습니다.

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

비즈니스 문제에 따라 이러한 지표의 특정 하위 집합에 대해 효과적으로 수행되는 모델에 더 관심을 가질 수 있습니다. 예를 들어 다음과 같이 두 비즈니스 애플리케이션은 ML 모델에 대해 매우 다른 요구 사항을 가질 수 있습니다.

- 한 애플리케이션은 실제로 긍정(높은 정밀도)인 긍정 예측에 대해 매우 높은 수준의 확신을 가져야 하며, 일부 긍정 사례를 부정(보통 수준의 재현율)으로 잘못 분류할 수 있어야 합니다.
- 다른 애플리케이션은 가능한 많은 수의 긍정 사례(높은 재현율)를 정확하게 예측해야 하며, 긍정(보통 수준의 정밀도)으로 잘못 분류된 일부 부정 사례를 수용해야 합니다.

Amazon ML에서는 이전 고급 지표의 특정 값에 해당하는 점수 커트라인을 선택할 수 있습니다. 또한 특정 지표에 대해 최적화할 때 발생하는 장단점도 보여줍니다. 예를 들어, 높은 정밀도에 해당하는 커트라인을 선택하면 일반적으로 재현율을 낮추면서 균형을 맞춰야 합니다.

> **Note**
>
> ML 모델별로 미래 예측을 분류하는 데 적용하려면 점수 커트라인을 저장해야 합니다.

멀티클래스 모델 인사이트 정보

예측 해석

많은 멀티클래스 분류 알고리즘의 실제 출력은 예측 점수 세트입니다. 점수는 주어진 관측치가 각 클래스에 속한다는 사실에 대한 시스템의 확실성을 나타냅니다. 바이너리 분류 문제와는 달리 예측을 하기 위해 점수 커트라인을 선택할 필요가 없습니다. 예측된 대답은 가장 높은 예측 점수를 가진 클래스(예: 레이블)입니다.

ML 모델 정확도 측정

멀티클래스에 사용되는 일반적인 지표는 바이너리 분류 사례에서 사용되는 지표와 동일합니다. Amazon ML에서 멀티클래스 지표의 예측 정확도를 평가하는 경우 매크로 평균 F1 점수가 사용됩니다.

매크로 평균 F1 점수

F1 점수는 이진 지표의 정밀도와 재현율을 모두 고려하는 바이너리 분류 지표입니다. 이는 정밀도와 재현율 사이의 조화 평균에 해당됩니다. 범위는 0 ~ 1입니다. 값이 클수록 예측 정확도가 더 높습니다.

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

매크로 평균 F1 점수는 멀티클래스 사례의 모든 클래스에 대한 F1 점수의 비가중치 평균입니다. 평가 데이터 세트의 클래스 발생 빈도는 고려하지 않습니다. 값이 클수록 예측 정확도가 더 높습니다. 다음 예제는 평가 데이터 소스의 K 클래스를 보여줍니다.

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

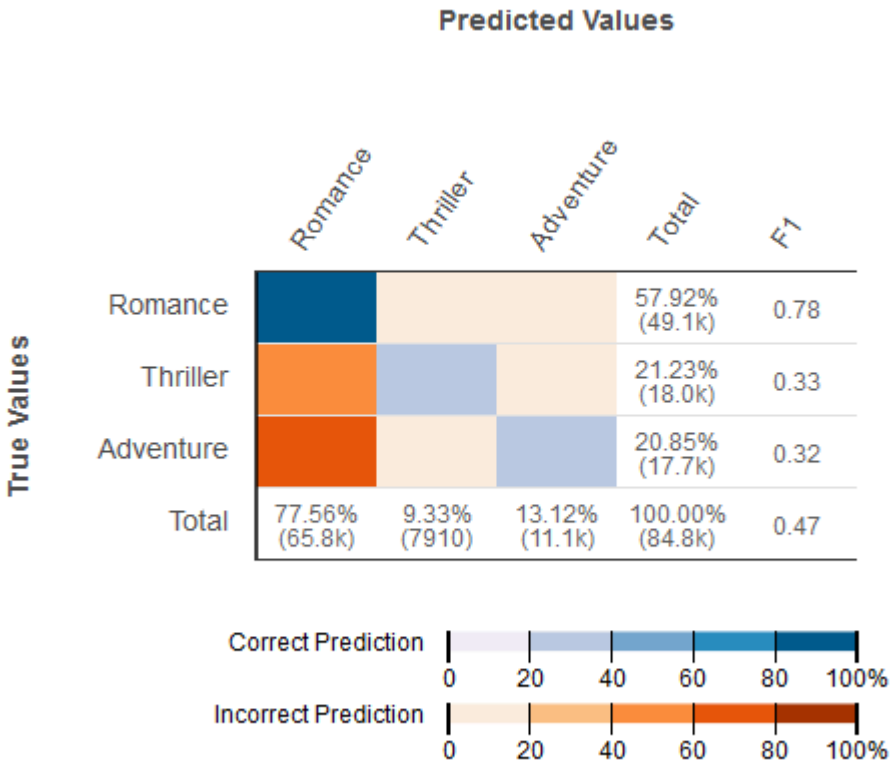
베이스라인 매크로 평균 F1 점수

Amazon ML은 멀티클래스 모델을 위한 기준 지표를 제공합니다. 이는 항상 가장 빈번한 클래스를 정답으로 예측하는 가상 멀티클래스 모델의 매크로 평균 F1 점수입니다. 예를 들어 영화의 장르를 예측하고 학습 데이터에서 가장 흔한 장르가 로맨스인 경우 기준 모델은 항상 장르를 로맨스로 예측합니다. ML 모델을 이 기준선과 비교하여 ML 모델이 이 상수 답을 예측하는 ML 모델보다 나은지 검증할 수 있습니다.

성능 시각화 사용

ML은 멀티클래스 분류 예측 모델의 정확도를 시각화하는 방법으로 오차 행렬을 제공합니다. 오차 행렬은 관측치의 예측 클래스와 실제 클래스를 비교하여 각 클래스에 대한 정답 및 오답 예측의 수 또는 백분율을 표로 보여줍니다.

예를 들어 영화를 장르별로 분류하려는 경우 예측 모델은 해당 장르(클래스)가 로맨스라고 예측할 수 있습니다. 하지만 실제 장르는 스릴러일 수도 있습니다. 다음 그림과 같이 멀티클래스 분류 ML 모델의 정확도를 평가할 때 Amazon ML은 이러한 오분류를 식별하여 그 결과를 오차 행렬에 표시합니다.



오차 행렬에는 다음 정보가 표시됩니다.

- 각 클래스에 대한 정답 및 오답 예측 수: 오차 행렬의 각 행은 실제 클래스 중 하나에 대한 지표에 해당합니다. 예를 들어, 첫 번째 행은 실제로 로맨스 장르에 속하는 영화의 경우 멀티클래스 ML 모델이 사례 중 80% 이상에 대해 정확한 예측을 내린다는 것을 보여줍니다. 이 경우 장르를 스릴러로 잘못 예측하는 경우는 20% 미만이고 어드벤처는 20% 미만입니다.
- 클래스별 F1 점수: 마지막 열에는 각 클래스의 F1 점수가 표시됩니다.
- 평가 데이터의 참 클래스 빈도: 두 번째 ~ 마지막 열은 평가 데이터 세트에서 평가 데이터의 관측치 중 57.92%가 로맨스, 21.23%가 스릴러, 20.85%가 어드벤처임을 보여줍니다.

- 평가 데이터에 대한 예측 클래스 빈도: 마지막 행은 예측에서 각 클래스의 빈도를 보여줍니다. 관측치의 77.56%는 로맨스로, 9.33%는 스릴러로, 13.12%는 각각 어드벤처로 각각 예측됩니다.

Amazon ML 콘솔은 오차 행렬의 클래스를 최대 10개까지 표시할 수 있는 시각적 표시를 제공하는데, 평가 데이터에서 빈도가 가장 높은 클래스부터 가장 빈도가 낮은 클래스 순으로 나열되어 있습니다. 평가 데이터에 10개 이상의 클래스가 있는 경우 오차 행렬에서 가장 자주 발생하는 상위 9개 클래스가 표시되고 다른 모든 클래스는 "others"라는 클래스로 축소됩니다. 또한 Amazon ML은 멀티클래스 시각화 페이지의 링크를 통해 오차 행렬 전체를 다운로드할 수 있는 기능도 제공합니다.

회귀 모델 인사이트 정보

예측 해석

회귀 ML 모델의 출력은 대상의 모델 예측에 대한 숫자 값입니다. 예를 들어 주택 가격을 예측하는 경우 모델의 예측은 254,013과 같은 값이 될 수 있습니다.

> **Note**
>
> 예측 범위가 학습 데이터의 대상 범위와 다를 수 있습니다. 예를 들어 주택 가격을 예측하고 있는데 학습 데이터의 대상이 0 ~ 450,000 범위의 값을 갖고 있다고 가정해 보겠습니다. 예측되는 대상이 같은 범위에 있을 필요는 없으며 양수 값(450,000 초과) 또는 음수 값(0 미만)을 취할 수 있습니다. 사용 중인 애플리케이션에 적합한 범위를 벗어나는 예측 값을 처리하는 방법을 계획하는 것이 중요합니다.

ML 모델 정확도 측정

회귀 작업의 경우 Amazon ML은 업계 표준 제곱 평균 제곱 오차(RMSE) 지표를 사용합니다. 이러한 지표는 예측 수치 대상과 실제 수치 대답(실측 정보) 간의 거리 측정에 해당됩니다. RMSE 값이 작을수록 모델의 예측 정확도가 높아집니다. 예측이 완벽하게 정확한 모델의 RMSE는 0입니다. 다음 예제에서는 N개의 레코드가 포함된 평가 데이터를 보여줍니다.

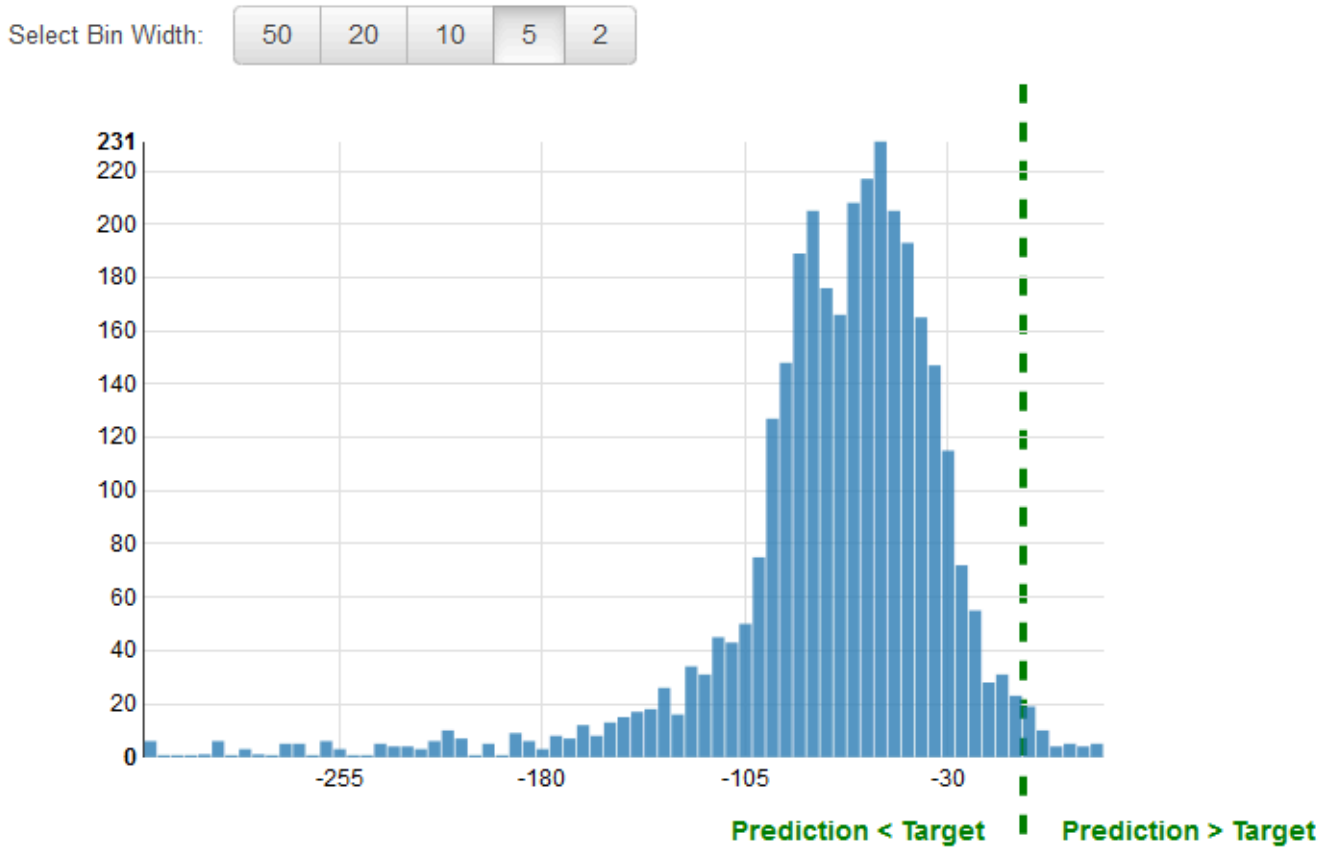
$$RMSE = \sqrt{1/N \sum_{i=1}^{N} (actual\ target - predicted\ target)^2}$$

기본 RMSE

Amazon ML은 회귀 모델을 위한 기준 지표를 제공합니다. 이는 항상 대상의 평균을 예측하여 답을 제시하는 가상 회귀 모델용 RMSE입니다. 예를 들어, 주택 구매자의 연령을 예측하고 학습 데이터에 포함된 관측치의 평균 연령이 35세인 경우 기준 모델은 항상 답을 35세로 예측합니다. ML 모델을 이 기준과 비교하여 ML 모델이 이 상수 답을 예측하는 ML 모델보다 나은지 검증할 수 있습니다.

성능 시각화 사용

회귀 문제에 대해 잔차를 검토하는 것이 일반적입니다. 평가 데이터에서 관측치에 대한 잔차는 실제 대상과 예측된 대상 간의 차이입니다. 잔차는 대상 중 모델이 예측할 수 없는 부분을 나타냅니다. 긍정 잔차는 모델이 대상을 과소평가하고 있다는 것을 나타냅니다(실제 대상이 예측된 대상보다 큼). 부정 잔차는 모델이 과대평가하고 있다는 것을 나타냅니다(실제 대상이 예측된 대상보다 작음). 종 모양으로 분포되고 0에 중심을 둔, 평가 데이터에 대한 잔차 히스토그램은 모델이 임의의 방식으로 오류를 만들고 대상 값의 특정 범위를 체계적으로 예측할 수 없다는 것을 나타냅니다. 잔차가 0에 중심을 둔 종 모양을 형성하지 않는다면 몇 가지 구조가 모델의 예측 오차에 포함된 것입니다. 모델에 변수를 더 추가하면 모델이 현재 모델이 캡처하지 않은 패턴을 캡처하는 데 도움이 될 수 있습니다. 다음 그림에서는 중심이 0이 아닌 잔차를 보여줍니다.



과적합 방지

ML 모델을 만들고 학습시킬 때 목표는 최상의 예측을 하는 모델을 선택하는 것입니다. 즉, 최상의 설정(ML 모델 설정 또는 하이퍼파라미터)을 가진 모델을 선택해야 합니다. Amazon Machine Learning에서는 사용자가 설정할 수 있는 네 가지 하이퍼파라미터, 즉 전달 수, 정규화, 모델 크기, 셔플 유형이 있습니다. 하지만 평가 데이터에서 "최상의" 예측 성능을 제공하는 모델 파라미터 설정을 선택하면 모델이 과적합될 수 있습니다. 과적합은 학습 및 평가 데이터 소스에서 발생하는 패턴을 모델에 기억시켰지만 데이터의 패턴을 일반화하지 못한 경우 발생합니다. 학습 데이터에 평가에 사용된 모든 데이터가 포함된 경우 자주 발생합니다. 과적합 모델은 평가 중에는 잘 작동하지만 보이지 않는 데이터에 대해서는 정확한 예측을 하지 못합니다.

과적합 모델이 최적 모델로 선택되지 않도록 추가 데이터를 예약하여 ML 모델의 성능을 검증할 수 있습니다. 예를 들어 데이터를 학습용 60%, 평가용 20%, 검증용 추가 20%로 나눌 수 있습니다. 평가 데이터에 적합한 모델 파라미터를 선택한 후 검증 데이터를 사용하여 두 번째 평가를 실행하여 ML 모델이 검증 데이터에서 얼마나 잘 수행되는지 확인합니다. 모델이 검증 데이터에 대한 기대치를 충족한다고 해서 모델이 데이터를 과적합시킨 것은 아닙니다.

세 번째 데이터 세트를 검증에 사용하면 적절한 ML 모델 파라미터를 선택하여 과적합을 방지할 수 있습니다. 하지만 학습 프로세스의 데이터를 평가와 검증 모두에 적용하면 학습에 사용할 수 있는 데이터가 줄어듭니다. 학습에 최대한 많은 데이터를 사용하는 것이 항상 최선이기 때문에 데이터 세트가 작은 경우에는 특히 문제가 됩니다. 이 문제를 해결하려면 교차 검증을 수행하면 됩니다. NFC 검증에 대한 자세한 내용은 교차 검증 단원을 참조하세요.

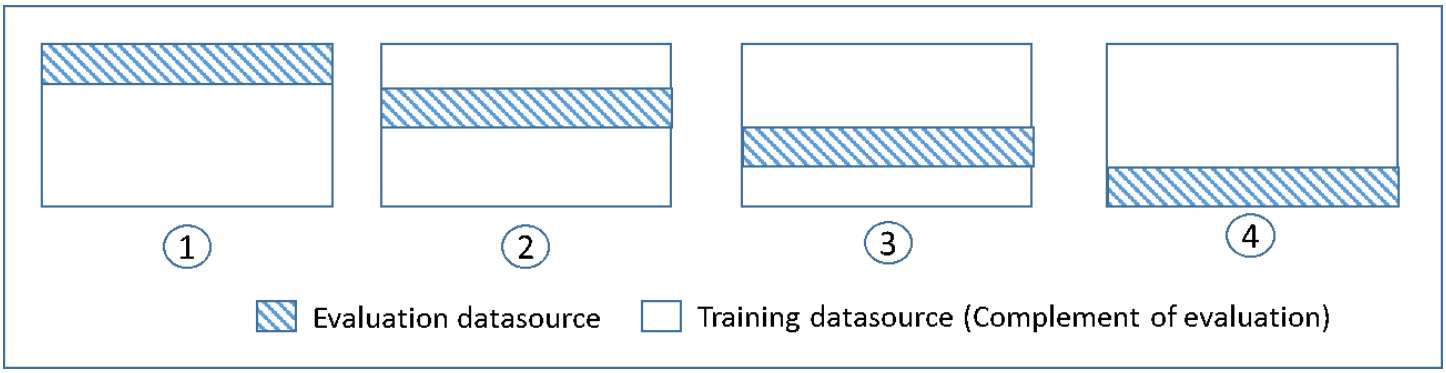
교차 검증

교차 검증은 사용 가능한 입력 데이터의 하위 집합에서 여러 ML 모델을 학습시키고 보완적인 데이터 하위 집합에서 평가하여 ML 모델을 평가하는 기법입니다. 교차 검증을 사용하면 과적합(예: 패턴 일반화 실패)을 탐지할 수 있습니다.

Amazon ML에서는 k중 교차 검증 방법을 사용하여 교차 검증을 수행할 수 있습니다. k-겹 교차 검증에서는 입력 데이터를 k개의 데이터 하위 집합(폴드라고도 함)으로 분할합니다. 부분 집합 중 하나(k-1)를 제외한 모든 하위 집합에 대해 ML 모델을 학습시킨 다음 학습에 사용되지 않은 부분 집합에서 모델을 평가합니다. 이 프로세스는 k번 반복되며, 매번 다른 하위 집합을 평가용으로 예약하고 학습에서 제외합니다.

다음 다이어그램에서는 4중 교차 검증을 통해 생성되고 훈련된 4개 모델 각각에 대해 생성된 학습 하위 집합과 보완 평가 하위 집합의 예를 보여줍니다. 모델 1은 데이터의 처음 25%를 평가에 사용하고 나머

지 75%는 학습에 사용합니다. 모델 2는 두 번째 부분 집합인 25%(25% ~ 50%)를 평가에 사용하고 나머지 세 가지 데이터 하위 집합은 학습에 사용하는 식입니다.



각 모델은 보완적인 데이터 소스를 사용하여 학습 및 평가됩니다. 평가 데이터 소스의 데이터에는 학습 데이터 소스에 없는 모든 데이터가 포함되며 이에만 국한됩니다. `createDatasourceFromS3`, `createDatasourceFromRedShift` 및 `createDatasourceFromRDS` API의 `DataRearrangement` 파라미터를 사용하여 이러한 각 하위 집합에 대한 데이터 소스를 생성합니다. `DataRearrangement` 파라미터에서 각 세그먼트의 시작 위치와 끝 위치를 지정하여 데이터 소스에 포함시킬 데이터 하위 집합을 지정합니다. 4k중 교차 검증에 필요한 보완 데이터 소스를 만들려면 다음 예제와 같이 `DataRearrangement` 파라미터를 지정합니다.

모델 1:

평가용 데이터 소스:

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

학습용 데이터 소스:

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

모델 2:

평가용 데이터 소스:

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

학습용 데이터 소스:

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

모델 3:

평가용 데이터 소스:

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

학습용 데이터 소스:

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

모델 4:

평가용 데이터 소스:

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

학습용 데이터 소스:

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

4중 교차 검증을 수행하면 모델 4개, 모델 학습을 위한 데이터 소스 4개, 모델 평가를 위한 데이터 소스 4개, 모델당 1개씩 총 4개의 평가가 생성됩니다. Amazon ML은 각 평가에 대해 모델 성능 지표를 생성합니다. 예를 들어 바이너리 분류 문제에 대한 4중 교차 검증에서 각 평가는 곡선하면적(AUC) 지표를 보고합니다. 4개의 AUC 지표의 평균을 계산하여 전체 성능 측정값을 얻을 수 있습니다. AUC 지표에 대한 자세한 내용은 ML 모델 정확도 측정 단원을 참조하세요.

교차 검증을 생성하고 모델 점수를 평균화하는 방법을 보여주는 샘플 코드에 대해서는 ML 샘플 코드 단원을 참조하세요.

모델 조정

모델을 교차 검증한 후 모델이 표준에 맞지 않는 경우 다음 모델에 대한 설정을 조정할 수 있습니다. 과적합에 대한 자세한 내용은 모델 적합성: 과소적합과 과적합 비교 단원을 참조하세요. 정규화에 대한 자세한 내용은 정규화 단원을 참조하세요. 정규화 설정 변경에 대한 자세한 내용은 사용자 지정 옵션을 사용하여 ML 모델 생성 단원을 참조하세요.

평가 경보

Amazon ML은 모델을 올바르게 평가했는지 검증하는 데 도움이 되는 인사이트 정보를 제공합니다. 평가 결과 검증 기준 중 하나라도 충족되지 않는 경우 Amazon ML 콘솔은 다음과 같이 위반된 검증 기준을 표시하여 경보를 표시합니다.

- ML 모델 평가는 보류된 데이터에 대해 수행됩니다.

  Amazon ML은 학습 및 평가에 동일한 데이터 소스를 사용하는 경우 경보를 표시합니다. Amazon ML을 사용하여 데이터를 분할하면 이 유효성 기준을 충족하게 됩니다. Amazon ML을 사용하여 데이터를 분할하지 않는 경우 학습 데이터 소스가 아닌 다른 데이터 소스로 ML 모델을 평가해야 합니다.

- 예측 모델 평가에 충분한 데이터 사용

  평가 데이터의 관측치/레코드 수가 학습 데이터 소스에 있는 관측치 수의 10% 미만인 경우 Amazon ML에서 경고합니다. 모델을 제대로 평가하려면 충분히 큰 데이터 샘플을 제공하는 것이 중요합니다. 이 기준을 통해 데이터를 너무 적게 사용하고 있는지 확인할 수 있습니다. ML 모델을 평가하는 데 필요한 데이터의 양은 주관적입니다. 더 나은 측정이 없을 경우를 대비하여 여기서는 10%를 임시방편으로 선택합니다.

- 스키마가 일치

  Amazon ML은 학습 데이터 소스와 평가 데이터 소스의 스키마가 동일하지 않을 경우 경보를 표시합니다. 평가 데이터 소스에 없는 특정 속성이 있거나 추가 속성이 있는 경우 Amazon ML은 이 경보를 표시합니다.

- 평가 파일의 모든 레코드가 예측 모델 성능 평가에 사용

  평가를 위해 제공된 모든 레코드가 실제로 모델을 평가하는 데 사용되었는지 아는 것이 중요합니다. 평가 데이터 소스의 일부 레코드가 유효하지 않아 정확도 지표 계산에 포함되지 않은 경우 Amazon ML에서 경보를 보냅니다. 예를 들어 평가 데이터 소스의 일부 관측치에 대상 변수가 누락된 경우 Amazon ML은 이러한 관측치에 대한 ML 모델의 예측이 정확한지 확인할 수 없습니다. 이 경우 타겟 값이 누락된 레코드는 유효하지 않은 것으로 간주됩니다.

- 대상 변수의 분포

  Amazon ML은 학습 및 평가 데이터 소스의 대상 속성 분포를 보여 주므로 두 데이터 소스에서 대상이 비슷하게 분포되어 있는지 검토할 수 있습니다. 평가 데이터에 대한 대상 분포와 대상 분포가 다른 학습 데이터를 기반으로 모델을 학습한 경우, 통계가 매우 다른 데이터를 대상으로 계산되므로 평가 품질이 저하될 수 있습니다. 데이터를 학습 데이터와 평가 데이터에 비슷하게 분포시키고 이러한 데이터 세트가 예측 시 모델이 접하게 될 데이터를 최대한 모방하도록 하는 것이 가장 좋습니다.

이 경보가 트리거되면 무작위 분할 전략을 사용하여 데이터를 학습 데이터 소스와 평가 데이터 소스로 분할해 봅니다. 드문 경우이긴 하지만 데이터를 무작위로 분할했는데도 이 경보가 목표 분포 차이에 대해 잘못 경고할 수도 있습니다. Amazon ML은 대략적인 데이터 통계를 사용하여 데이터 분포를 평가하며, 때때로 이 경보가 오류로 트리거됩니다.

예측 생성 및 해석

Amazon ML은 예측을 생성하는 두 가지 메커니즘, 즉 비동기식 메커니즘(배치 기반)과 동기식 메커니즘(한 번에 하나씩)을 제공합니다.

관측치가 많은데 관측치에 대한 예측을 모두 한꺼번에 얻으려는 경우 비동기식 예측 또는 배치 예측을 사용합니다. 이 프로세스는 데이터 소스를 입력으로 사용하고 선택한 S3 버킷에 저장된 .csv 파일로 예측을 출력합니다. 예측 결과에 액세스하려면 배치 예측 프로세스가 완료될 때까지 기다려야 합니다. Amazon ML이 배치 파일에서 처리할 수 있는 데이터 소스의 최대 크기는 1TB(레코드가 약 1억 개)입니다. 데이터 소스가 1TB보다 크면 작업이 실패하고 Amazon ML에서 오류 코드를 반환합니다. 이를 방지하려면 데이터를 여러 배치로 나눕니다. 일반적으로 레코드가 더 긴 경우 레코드 1억 개가 처리되기 전에 1TB 한도에 도달하게 됩니다. 이 경우 AWS 지원 팀에 문의하여 배치 예측에 필요한 작업 규모를 늘리는 것이 좋습니다.

짧은 지연 시간으로 예측을 얻으려는 경우, 동기 또는 실시간 예측을 사용합니다. 실시간 예측 API는 JSON 문자열로 직렬화된 단일 입력 관측치를 받아들이고 API 응답의 일부로 예측 및 관련 메타데이터를 동기적으로 반환합니다. API를 두 번 이상 동시에 직접 호출하여 동기식 예측을 병렬로 가져올 수 있습니다. 실시간 예측 API의 처리량 제한에 대한 자세한 내용은 ML API 참조의 실시간 예측 한도를 참조하세요.

주제

- 배치 예측 생성
- 배치 예측 지표 검토
- 배치 예측 출력 파일 읽기
- 실시간 예측 요청

배치 예측 생성

배치 예측을 생성하려면 머신 러닝(ML) 콘솔 또는 API를 사용하여 `BatchPrediction` 객체를 생성합니다. `BatchPrediction` 객체는 ML이 ML 모델 및 입력 관측치 세트를 사용하여 생성하는 예측 세트를 설명합니다. `BatchPrediction` 객체를 생성하면 ML이 예측을 계산하는 비동기 워크플로를 시작합니다.

배치 예측을 얻는 데 사용하는 데이터 소스와 예측을 위해 쿼리하는 ML 모델을 학습시키는 데 사용한 데이터 소스에 동일한 스키마를 사용해야 합니다. 한 가지 예외는 Amazon ML이 대상을 예측하기 때

문에 배치 예측의 데이터 소스에 대상 속성을 포함시킬 필요가 없다는 것입니다. 대상 속성을 제공하는 경우 Amazon ML은 해당 값을 무시합니다.

배치 예측 생성(콘솔)

Amazon ML 콘솔을 사용하여 배치 예측을 생성하려면 배치 예측 생성 마법사를 사용합니다.

배치 예측을 생성하려면(콘솔)

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/machinelearning/ Amazon Machine Learning 콘솔을 엽니다.
2. ML 대시보드의 객체에서 새로 만들기...를 선택한 다음 배치 예측을 선택합니다.
3. 배치 예측을 생성하는 데 사용할 Amazon ML 모델을 선택합니다.
4. 이 모델을 사용할 것인지 확인하려면 계속을 선택합니다.
5. 예측을 생성하려는 데이터 소스를 선택합니다. 대상 속성을 포함시킬 필요는 없지만 데이터 소스는 모델과 동일한 스키마를 가져야 합니다.
6. 계속을 선택합니다.
7. S3 목적지에서 S3 버킷의 이름을 입력합니다.
8. 검토를 선택합니다.
9. 설정을 검토한 후 배치 예측 생성을 선택합니다.

배치 예측 생성(API)

ML API를 사용하여 `BatchPrediction` 객체를 생성하려면 다음 파라미터를 제공해야 합니다.

데이터 소스 ID

예측하려는 관측치를 가리키는 데이터 소스의 ID입니다. 예를 들어 `s3://examplebucket/input.csv`이라는 파일의 데이터를 예측하려면 데이터 파일을 가리키는 데이터 소스 객체를 만든 다음 이 파라미터와 함께 해당 데이터 소스의 ID를 전달하면 됩니다.

배치 예측 ID

배치 예측에 할당할 ID입니다.

ML 모델 ID

Amazon ML이 예측을 위해 쿼리해야 하는 ML 모델의 ID입니다.

출력 Uri

예측 결과를 저장할 S3 버킷의 URI입니다. Amazon ML에 이 버킷에 데이터를 쓸 권한이 있어야 합니다.

`OutputUri` 파라미터는 다음 예와 같이 슬래시('/') 문자로 끝나는 S3 경로를 참조해야 합니다.

s3://examplebucket/examplepath/

S3 권한 구성에 대한 자세한 내용은 Amazon S3에 예측을 출력할 수 있는 권한을 Amazon ML에 부여 단원을 참조하세요.

(선택 사항) 배치 예측 이름

(선택 사항) 배치 예측에 사용할 수 있는 이름입니다.

배치 예측 지표 검토

머신 러닝(ML)은 배치 예측을 생성한 후 `Records seen`과 `Records failed to process`의 두 가지 지표를 제공합니다. `Records seen`은 ML이 배치 예측을 실행할 때 살펴본 레코드 수를 알려줍니다. `Records failed to process`는 ML에서 처리할 수 없는 레코드 수를 알려줍니다.

Amazon ML에서 실패한 레코드를 처리할 수 있도록 하려면 데이터 소스를 생성하는 데 사용된 데이터의 레코드 형식을 확인하고 필요한 속성이 모두 있고 모든 데이터가 올바른지 확인합니다. 데이터를 수정한 후에는 배치 예측을 다시 생성하거나 실패한 레코드로 새 데이터 소스를 생성한 다음 새 데이터 소스를 사용하여 새 배치 예측을 생성할 수 있습니다.

배치 예측 지표 검토(콘솔)

ML 콘솔에서 지표를 확인하려면 배치 예측 요약 페이지를 열고 처리된 정보 섹션을 살펴봅니다.

배치 예측 지표 및 세부 정보 검토(API)

ML API를 사용하여 레코드 지표를 비롯한 `BatchPrediction` 객체에 대한 세부 정보를 검색할 수 있습니다. Amazon ML은 다음과 같은 배치 예측 API 직접 호출을 제공합니다.

- `CreateBatchPrediction`
- `UpdateBatchPrediction`
- `DeleteBatchPrediction`

- GetBatchPrediction
- DescribeBatchPredictions

자세한 내용은 ML API 참조 단원을 참조하세요.

배치 예측 출력 파일 읽기

배치 예측 출력 파일을 검색하려면 다음 단계를 수행합니다.

1. 배치 예측 매니페스트 파일을 찾습니다.
2. 매니페스트 파일을 읽고 출력 파일의 위치를 확인합니다.
3. 예측이 포함된 출력 파일을 검색합니다.
4. 출력 파일의 콘텐츠를 해석합니다. 콘텐츠는 예측을 생성하는 데 사용된 ML 모델의 유형에 따라 달라집니다.

다음에 이어지는 단원에서는 이러한 단계에 대해 자세히 설명합니다.

배치 예측 매니페스트 파일 찾기

배치 예측의 매니페스트 파일에는 입력 파일을 예측 출력 파일에 매핑하는 정보가 들어 있습니다.

매니페스트 파일을 찾으려면 배치 예측 객체를 만들 때 지정한 출력 위치부터 시작합니다. 완료된 배치 예측 객체를 쿼리하여 ML API 또는 https://console.aws.amazon.com/machinelearning/ 중 하나를 사용하면 이 파일의 S3 위치를 검색할 수 있습니다.

매니페스트 파일은 출력 위치에 추가된 정적 문자열 /batch-prediction/과, 매니페스트 파일 이름(배치 예측의 ID)과 여기에 확장자 .manifest가 추가된 경로로 구성된 경로의 출력 위치에 있습니다.

예를 들어 ID bp-example의 배치 예측 객체를 생성하고 S3 위치 s3://examplebucket/output/를 출력 위치로 지정했다면 여기에서 매니페스트 파일을 찾을 수 있을 것입니다.

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

매니페스트 파일 읽기

.manifest 파일의 콘텐츠는 JSON 맵으로 인코딩되어 있는데, 키는 S3 입력 데이터 파일 이름의 문자열에, 값은 관련 배치 예측 결과 파일의 문자열에 각각 해당됩니다. 입력/출력 파일 쌍마다 매핑 라인이 하나씩 있습니다. 예제를 계속 살펴보면, BatchPrediction 객체 생성을 위한 입력이 s3://

examplebucket/input/에 있는 data.csv 라는 단일 파일로 구성된 경우 다음과 같은 매핑 문자열이 표시될 수 있습니다.

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

BatchPrediction 객체 생성에 대한 입력이 data1.csv, data2.csv, data3.csv 라는 세 개의 파일로 구성되어 있고 이들 파일이 모두 S3 위치 s3://examplebucket/input/에 저장되어 있는 경우 다음과 같은 매핑 문자열이 표시될 수 있습니다.

```
{"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"}
```

배치 예측 출력 파일 검색

매니페스트 매핑에서 가져온 각 배치 예측 파일을 다운로드하여 로컬로 처리할 수 있습니다. 파일 형식은 CSV이고, gzip 알고리즘으로 압축되어 있습니다. 해당 파일 내에는 해당 입력 파일의 입력 관측치당 라인이 하나씩 있습니다.

예측을 배치 예측의 입력 파일과 결합하려면 두 파일을 레코드별로 간단히 병합하면 됩니다. 배치 예측의 출력 파일에는 항상 예측 입력 파일과 같은 수의 레코드가 같은 순서로 포함됩니다. 입력 관측치 처리에 실패하고 예측을 생성할 수 없는 경우 배치 예측 출력 파일의 해당 위치에 빈 줄이 생깁니다.

바이너리 분류 ML 모델용 배치 예측 파일의 콘텐츠 해석

바이너리 분류 모델용 배치 예측 파일의 열 이름은 최고응답 및 score로 지정됩니다.

최고응답 열에는 예측 점수를 커트라인 점수와 비교하여 얻은 예측 레이블("1" 또는 "0")이 들어 있습니다. 커트라인 점수에 대한 자세한 내용은 점수 커트라인 조정 단원을 참조하세요. ML 모델의 커트라인 점수는 Amazon ML API 또는 Amazon ML 콘솔의 모델 평가 기능을 사용하여 설정합니다. 커트라인 점수를 설정하지 않은 경우 Amazon ML은 기본 값인 0.5를 사용합니다.

score 열에는 이 예측에 대해 ML 모델에서 할당한 원시 예측 점수가 들어 있습니다. ML은 로지스틱 회귀 모델을 사용하므로 이 점수는 실제 ("1") 값에 해당하는 관측치의 확률을 모델링하려고 시도합니다.

참고로 점수는 과학적 표기법으로 보고되므로 다음 예제의 첫 번째 행에서 8.7642E-3 값은 0.0087642와 같습니다.

예를 들어 ML 모델의 커트라인 점수가 0.75인 경우 바이너리 분류 모델에 대한 배치 예측 출력 파일의 콘텐츠는 다음과 같을 수 있습니다.

```
bestAnswer,score

0,8.7642E-3

1,7.899012E-1


0,6.323061E-3

0,2.143189E-2


1,8.944209E-1
```

입력 파일에서 두 번째 및 다섯 번째 관측치의 예측 점수가 0.75를 넘었으므로 이들 관측치에 대한 bestAnswer 열에는 값 "1"이 표시되는 반면에 다른 관측치의 값은 "0"이 됩니다.

멀티클래스 분류 ML 모델용 배치 예측 파일의 콘텐츠 해석

멀티클래스 모델용 배치 예측 파일은 학습 데이터에 있는 각 클래스에 대해 하나의 열을 포함하고 있습니다. 열 이름은 배치 예측 파일의 헤더 라인에 표시됩니다.

멀티클래스 모델에서 예측을 요청하면 Amazon ML은 입력 파일의 각 관측치에 대해 입력 데이터 세트에 정의된 각 클래스별로 하나씩 여러 예측 점수를 계산합니다. 이는 "다른 클래스와 달리 이 관측치가 이 클래스에 속할 확률(0과 1 사이로 측정)은 얼마입니까?" 라고 묻는 것과 같습니다. 각 점수는 "관측치가 이 클래스에 속할 확률"로 해석될 수 있습니다. 예측 점수가 한 클래스 또는 다른 클래스에 속해 있는 관측치의 기본 확률을 모델링한 것이기 때문에 한 행에 있는 모든 예측 점수의 합계는 1이 됩니다. 클래스 하나를 모델의 예측 클래스로 선택해야 합니다. 확률이 가장 높은 클래스를 가장 좋은 답으로 선택하는 것이 가장 일반적일 것입니다.

예를 들어, 별 1개에서 5개까지의 척도를 기준으로 제품에 대한 고객의 평점을 예측한다고 생각해 보겠습니다. 클래스 이름을 1_star, 2_stars, 3_stars, 4_stars, 5_stars로 지정한다면 멀티클래스 예측 출력 파일은 다음과 같이 표시될 수 있습니다.

```
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
```

이 예제에서는 첫 번째 관측치가 3_stars 클래스에 대한 예측 점수가 가장 높으므로(예측 점수 = 4.77781E-1), 이 관측치에 대한 최선의 답은 클래스 3_stars임을 보여주는 것으로 결과를 해석할 수 있을 것입니다. 참고로 예측 점수는 과학 표기법으로 보고되므로 예측 점수 4.77781E-1은 0.477781과 같습니다.

확률이 가장 높은 클래스를 선택하고 싶지 않은 경우도 있을 수 있습니다. 예를 들어, 예측 점수가 가장 높더라도 해당 클래스가 가장 좋은 답으로 간주되지 않도록 최소 임계값을 이 이하로 설정하는 것이 좋습니다. 영화를 장르별로 분류하고 장르를 최선의 답으로 선언하기 전에 예측 점수가 5E-1 이상이어야 한다고 가정해 보겠습니다. 코미디의 경우 3E-1, 드라마의 경우 2.5E-1, 다큐멘터리의 경우 2.5E-1, 액션 영화의 경우 2E-1 등의 예측 점수를 얻습니다. 이 경우 ML 모델은 코미디가 가장 가능성이 높은 선택이라고 예측하지만 사용자는 코미디를 최선의 답으로 선택하지 않기로 결정합니다. 예측 점수 중 어느 것도 기준 예측 점수인 5E-1을 초과하지 않았기 때문에 장르를 자신 있게 예측하기에는 예측이 충분하지 않다고 판단하고 다른 것을 선택하기로 결정합니다. 그러면 애플리케이션에서 이 영화의 장르 필드를 "알 수 없음"으로 간주할 수 있습니다.

회귀 ML 모델용 배치 예측 파일의 콘텐츠 해석

회귀 모델용 배치 예측 파일에는 score라는 단일 열이 포함되어 있습니다. 이 열에는 입력 데이터의 각 관측치에 대한 원시 수치 예측이 들어 있습니다. 값은 과학적 표기법으로 보고되므로 다음 예제의 첫 번째 행에서 -1.526385E1의 점수 값은 -15.26835와 같습니다.

이 예제는 회귀 모델에서 수행된 배치 예측의 출력 파일을 보여줍니다.

```
score

-1.526385E1

-6.188034E0
```

```
-1.271108E1

-2.200578E1

8.359159E0
```

실시간 예측 요청

실시간 예측은 Amazon Machine Learning(Amazon ML)에 대한 동기식 직접 호출입니다. Amazon ML이 요청을 받으면 예측이 수행되고 응답은 즉시 반환됩니다. 실시간 예측은 일반적으로 대화식 웹, 모바일 또는 데스크톱 애플리케이션에서 예측 기능을 활성화하는 데 사용됩니다. ML을 통해 생성된 ML 모델에 지연 시간이 짧은 `Predict` API를 사용하여 실시간으로 예측을 쿼리할 수 있습니다. `Predict` 작업은 요청 페이로드에서 단일 입력 관측치를 수용하고 응답에서 예측을 동기식으로 반환합니다. 이를 통해, 입력 관측치의 위치를 가리키는 Amazon ML 데이터 소스 객체의 ID를 사용하여 간접적으로 호출되는 배치 예측 API와 별개로 설정되며, 모든 관측치에 대한 예측을 포함하는 파일에 URI를 비동기식으로 반환합니다. Amazon ML은 100밀리초 이내에 대부분의 실시간 예측 요청에 응답합니다.

Amazon ML 콘솔에서 비용 발생 없이 실시간 예측을 시도할 수 있습니다. 실시간 예측을 사용하기로 결정한 경우 먼저 실시간 예측 생성을 위한 엔드포인트를 생성해야 합니다. 이 작업은 ML 콘솔에서 또는 `CreateRealtimeEndpoint` API를 사용하여 수행할 수 있습니다. 엔드포인트를 생성한 후에는 실시간 예측 API를 사용하여 실시간 예측을 생성합니다.

> **Note**
>
> 모델에 대해 실시간 엔드포인트를 생성한 후 모델의 크기를 기준으로 용량 예약 요금이 부과되기 시작합니다. 자세한 내용은 요금 단원을 참조하세요. 콘솔에 실시간 엔드포인트를 생성하면 엔드포인트에서 지속적으로 발생하는 예상 요금의 세부 내역이 콘솔에 표시됩니다. 해당 모델에서 더 이상 실시간 예측을 얻을 필요가 없을 때 요금 부과를 중단하려면 콘솔 또는 `DeleteRealtimeEndpoint` 작업을 사용하여 실시간 엔드포인트를 제거합니다.

`Predict` 요청 및 응답의 예는 머신 러닝 API 참조의 예측 단원을 참조하세요. 모델을 사용하는 정확한 응답 형식의 예를 확인하려면 실시간 예측 시도 단원을 참조하세요.

주제

- 실시간 예측 시도
- 실시간 엔드포인트 생성

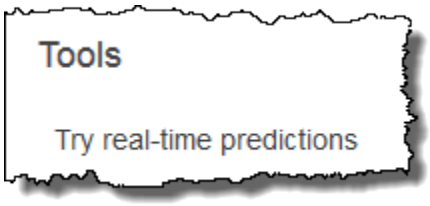
- 실시간 예측 엔드포인트 찾기(콘솔)
- 실시간 예측 엔드포인트 찾기(API)
- 실시간 예측 요청 생성
- 실시간 엔드포인트 삭제

실시간 예측 시도

실시간 예측을 사용할지 여부를 결정하는 데 도움을 주기 위해 Amazon ML은 실시간 예측 엔드포인트 설정과 관련된 추가 요금의 발생 없이 단일 데이터 레코드에 대한 예측을 시도할 수 있게 허용합니다. 실시간 예측을 시도하려면 ML 모델이 있어야 합니다. 대규모 실시간 예측을 생성하려면 머신 러닝 API 참조의 Predict API를 사용합니다.

실시간 예측을 시도하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/machinelearning/ Amazon Machine Learning 콘솔을 엽니다.
2. 탐색 모음의 머신 러닝 드롭다운에서 ML 모델을 선택합니다.
3. 자습서의 `Subscription propensity model`과 같이 실시간 예측을 시도하는 데 사용할 모델을 선택합니다.
4. ML 모델 보고서 페이지의 예측에서 요약을 선택한 다음 실시간 예측 시도를 선택합니다.



   Amazon ML이 이 모델을 학습시키는 데 사용한 데이터 레코드를 구성하는 변수 목록을 보여줍니다.
5. 양식의 각 필드에 데이터를 입력하거나 단일 데이터 레코드를 CSV 형식으로 텍스트 상자에 붙여넣어 계속 진행할 수 있습니다.

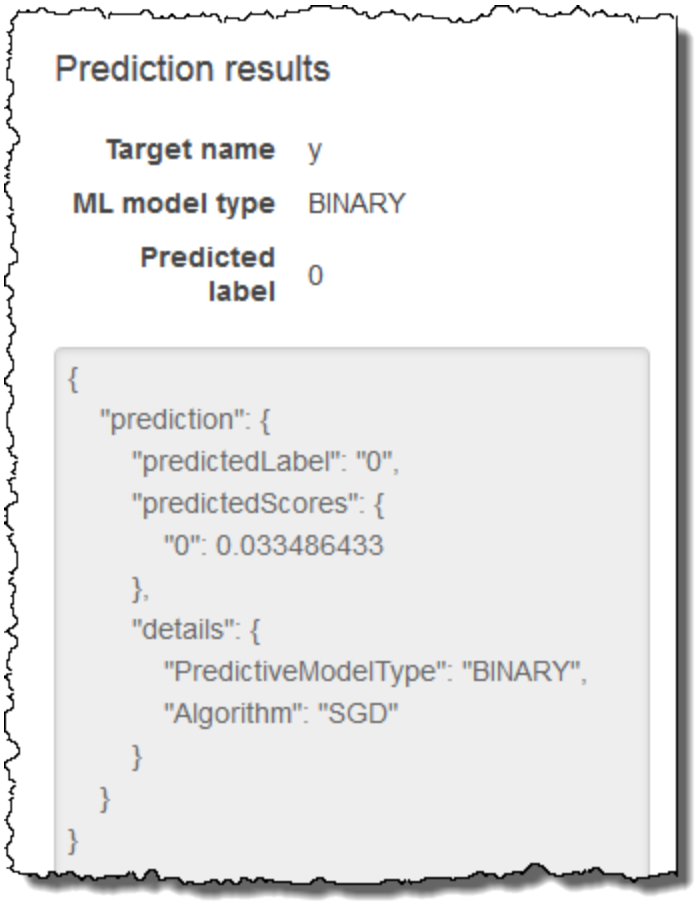
   해당 양식을 사용하려면 각 값 필드에 실시간 예측을 테스트하는 데 사용할 데이터를 입력합니다. 입력하는 데이터 레코드에 하나 이상의 데이터 속성에 대한 값이 포함되어 있지 않은 경우 입력 필드를 비워 둡니다.

   데이터 레코드를 제공하려면 레코드 붙여넣기를 선택합니다. 단일 CSV 형식의 데이터 행을 텍스트 필드에 붙여넣고 제출을 선택합니다. ML이 자동으로 값 필드를 채웁니다.

> **Note**
>
> 데이터 레코드의 데이터는 학습 데이터와 동일한 수의 열을 가져야 하며 동일한 순서로 배열되어야 합니다. 유일한 예외 사항은 대상 값을 생략해야 한다는 것입니다. 대상 값을 포함시킨 경우 Amazon ML이 무시합니다.

6. 페이지 하단에서 예측 생성을 선택합니다. Amazon ML이 즉시 예측을 반환합니다.

   예측 결과 창에 `Predict` API 직접 호출이 반환하는 예측 객체가 ML 모델 유형, 대상 변수 이름 및 예상 클래스 또는 값과 함께 표시됩니다. 결과 해석에 대한 자세한 내용은 바이너리 분류 ML 모델용 배치 예측 파일의 콘텐츠 해석 단원을 참조하세요.



실시간 엔드포인트 생성

실시간 예측을 생성하려면 실시간 엔드포인트를 생성해야 합니다. 실시간 엔드포인트를 생성하려면 먼저 실시간 예측을 생성할 ML 모델이 있어야 합니다. ML 콘솔을 사용하거나 `CreateRealtimeEndpoint` API를 호출하여 실시간 엔드포인트를 생성할 수 있습니다.

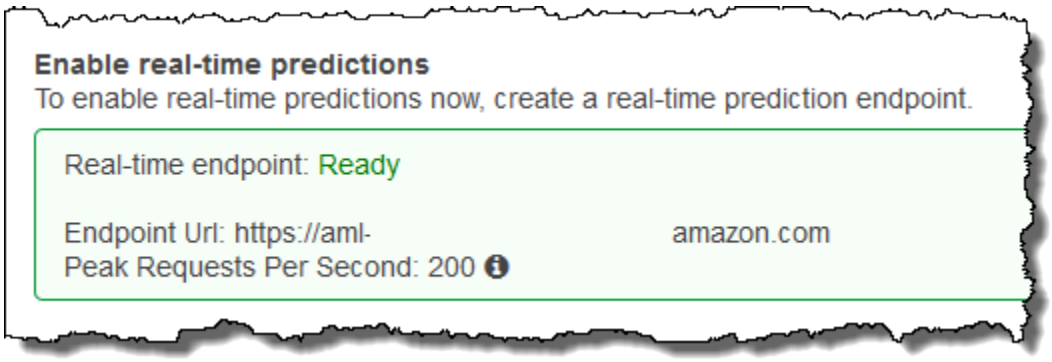
`CreateRealtimeEndpoint` API 사용에 대한 자세한 내용은 머신 러닝 API 참조의 https://docs.aws.amazon.com/machine-learning/latest/APIReference/API_CreateRealtimeEndpoint.html 단원을 참조하세요.

실시간 엔드포인트를 생성하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/machinelearning/ Amazon Machine Learning 콘솔을 엽니다.
2. 탐색 모음의 머신 러닝 드롭다운에서 ML 모델을 선택합니다.
3. 실시간 예측을 생성하려는 모델을 선택합니다.
4. ML 모델 요약 페이지의 예측 아래에서 실시간 엔드포인트 생성을 선택합니다.

   실시간 예측의 가격 책정 방법을 설명하는 대화 상자가 나타납니다.
5. 생성을 선택합니다. 실시간 엔드포인트 요청이 Amazon ML로 전송되어 대기열로 들어갑니다. 실시간 엔드포인트의 상태는 업데이트하는 중이 됩니다.

6. 실시간 엔드포인트가 준비되면 상태가 준비 상태로 변경되고 ML이 엔드포인트 URL을 표시합니다. 엔드포인트 URL을 사용하여 `Predict` API를 통해 실시간 예측을 생성합니다. `Predict` API 사용에 대한 자세한 내용은 머신 러닝 API 참조의 https://docs.aws.amazon.com/machine-learning/latest/APIReference/API_Predict.html 단원을 참조하세요.



실시간 예측 엔드포인트 찾기(콘솔)

ML 콘솔을 사용하여 ML 모델의 엔드포인트 URL을 찾으려면 모델의 ML 모델 요약 페이지로 이동합니다.

실시간 엔드포인트 URL을 찾으려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/machinelearning/ Amazon Machine Learning 콘솔을 엽니다.
2. 탐색 모음의 머신 러닝 드롭다운에서 ML 모델을 선택합니다.
3. 실시간 예측을 생성하려는 모델을 선택합니다.
4. ML 모델 요약 페이지에서 예측 섹션이 보일 때까지 스크롤을 아래로 내립니다.
5. 모델의 엔드포인트 URL은 실시간 예측에 나열되어 있습니다. 해당 URL을 실시간 예측 호출에 대한 엔드포인트 Url URL로 사용합니다. 엔드포인트를 사용하여 예측을 생성하는 방법에 대한 자세한 내용은 머신 러닝 API 참조의 https://docs.aws.amazon.com/machine-learning/latest/APIReference/API_Predict.html 단원을 참조하세요.

실시간 예측 엔드포인트 찾기(API)

`CreateRealtimeEndpoint` 작업을 사용하여 실시간 엔드포인트를 생성하면 응답에 URL 및 엔드포인트 상태가 반환됩니다. 콘솔을 사용하여 실시간 엔드포인트를 생성했거나 이전에 생성한 엔드포인트의 URL 및 상태를 검색하려는 경우 실시간 예측에 대해 쿼리하려는 모델의 ID로 `GetMLModel` 작업을 직접적으로 호출합니다. 엔드포인트 정보는 응답의 `EndpointInfo` 섹션에 포함되어 있습니다. 실시간 엔드포인트가 연결된 모델의 경우 `EndpointInfo`는 다음과 같은 모습일 수 있습니다.

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

실시간 엔드포인트가 없는 모델은 다음을 반환합니다.

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

```
}
```

실시간 예측 요청 생성

샘플 Predict 요청 페이로드는 다음과 같은 모습일 수 있습니다.

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

PredictEndpoint 필드는 EndpointInfo 구조의 EndpointUrl 필드와 일치해야 합니다. Amazon ML은 이 필드를 사용하여 실시간 예측 플릿의 해당 서버로 요청을 라우팅합니다.

MLModelId는 이전에 학습된 실시간 엔드포인트가 있는 모델의 식별자입니다.

Record는 변수 이름에 대한 변수 값의 맵입니다. 각 쌍은 관측치를 나타냅니다. Record 맵에는 ML 모델에 대한 입력이 포함되어 있습니다. 대상 변수가 없는 학습 데이터 세트의 단일 행 데이터와 유사합니다. 학습 데이터의 값 유형에 관계없이 Record에는 문자열-문자열 매핑이 포함됩니다.

> **Note**
>
> 값을 가지지 않은 변수를 생략할 수 있지만 이렇게 되면 예측 정확성은 감소할 수 있습니다. 더 많은 변수를 포함할수록 모델이 정확해집니다.

Predict 요청으로 반환된 응답의 형식은 예측에 대해 쿼리 중인 모델의 유형에 따라 다릅니다. 모든 사례에서 details 필드에는 특히 모델 유형이 있는 PredictiveModelType 필드를 비롯하여 예측 요청에 대한 정보가 포함되어 있습니다.

다음 예에서는 이진 모델에 대한 응답 이벤트를 보여줍니다.

```
{
    "Prediction":{
        "details":{
            "PredictiveModelType": "BINARY"
```

```
        },
        "predictedLabel": "0",
        "predictedScores":{
            "0": 0.47380468249320984
        }
    }
}
```

예측된 레이블이 포함된 `predictedLabel` 필드에 주목하세요(이 경우 0). Amazon ML은 예측 점수를 분류 기준과 비교하여 예측 레이블을 계산합니다.

- `GetMLModel` 작업에 응하여 `ScoreThreshold` 필드를 검사하거나 ML 콘솔의 모델 정보를 확인하여 현재 ML 모델과 연결된 분류 커트라인을 얻을 수 있습니다. 점수 임계값을 설정하지 않으면 Amazon ML은 기본 값인 0.5를 사용합니다.
- `predictedScores` 맵을 검사하면 바이너리 분류 모델의 정확한 예측 점수를 얻을 수 있습니다. 이 맵 내에서 예측된 레이블은 정확한 예측 점수와 짝을 이룹니다.

이진 예측에 대한 자세한 내용은 예측 해석 단원을 참조하세요.

다음 예에서는 회귀 모델에 대한 응답 이벤트를 보여줍니다. 예측된 숫자 값은 `predictedValue` 필드에서 찾을 수 있습니다.

```
{
    "Prediction":{
        "details":{
            "PredictiveModelType": "REGRESSION"
        },
        "predictedValue": 15.508452415466309
    }
}
```

다음 예에서는 멀티클래스 모델에 대한 응답 이벤트를 보여줍니다.

```
{
    "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
        }
    }
}
```

바이너리 분류 모델과 마찬가지로 예측된 레이블/클래스도 `predictedLabel` 필드에서 찾을 수 있습니다. `predictedScores` 맵을 확인하여 예측이 각 클래스와 얼마나 큰 관련이 있는지 더욱 깊이 이해할 수 있습니다. 이 맵에서 클래스의 점수가 높을수록 예측이 해당 클래스와 더욱 큰 관련이 있으며 가장 높은 값은 궁극적으로 `predictedLabel`로 선택됩니다.

멀티클래스 예측에 대한 자세한 내용은 멀티클래스 모델 인사이트 정보 단원을 참조하세요.

실시간 엔드포인트 삭제

실시간 예측을 완료했으면 추가 요금이 발생하지 않도록 실시간 엔드포인트를 삭제합니다. 엔드포인트를 삭제하는 즉시 요금 발생이 중지됩니다.

실시간 엔드포인트를 삭제하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/machinelearning/ Amazon Machine Learning 콘솔을 엽니다.
2. 탐색 모음의 머신 러닝 드롭다운에서 ML 모델을 선택합니다.
3. 실시간 예측이 더 이상 필요하지 않은 모델을 선택합니다.
4. ML 모델 보고서 페이지의 예측에서 요약을 선택합니다.
5. 실시간 엔드포인트 삭제를 선택합니다.
6. 실시간 엔드포인트 삭제 대화 상자에서 삭제를 선택합니다.

Amazon ML 객체 관리

Amazon ML은 Amazon ML 콘솔 또는 Amazon ML API를 통해 관리할 수 있는 다음 네 가지 객체를 제공합니다.

- 데이터 소스
- ML 모델
- 평가
- 배치 예측

각 객체는 기계 학습 애플리케이션을 빌드하는 수명 주기에서 서로 다른 용도로 사용되며, 각 객체에는 해당 객체에만 적용되는 특정 속성 및 기능이 있습니다. 이러한 차이점에도 불구하고 객체를 관리하는 방법은 비슷합니다. 예를 들어 객체를 나열하고, 설명을 검색하고, 업데이트하거나 삭제하는 데 거의 동일한 프로세스를 사용합니다.

다음에 이어지는 단원에서는 네 객체 모두에 공통적인 관리 작업을 설명하고 차이점에 대해 설명합니다.

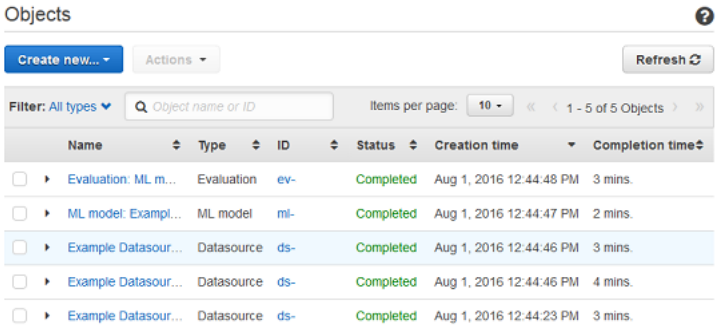
주제

객체 나열

Amazon Machine Learning(Amazon ML) 데이터 소스, ML 모델, 평가 및 배치 예측에 대한 자세한 내용을 보려면 해당 목록을 나열합니다. 각 객체에 대해 이름, 유형, ID, 상태 코드, 생성 시간이 표시됩니다. 특정 객체 유형별 세부 정보도 볼 수 있습니다. 예를 들어, 데이터 소스의 데이터 인사이트 정보를 볼 수 있습니다.

객체 나열(콘솔)

최근에 생성한 1,000개의 객체 목록을 보려면 ML 콘솔에서 객체 대시보드를 엽니다. 객체 대시보드를 표시하려면 ML 콘솔에 로그인합니다.



해당 객체 유형별 세부 정보를 포함하여 객체에 대한 자세한 세부 정보를 보려면 객체의 이름 또는 ID를 선택합니다. 예를 들어 데이터 소스에 대한 데이터 인사이트 정보를 보려면 데이터 소스 이름을 선택합니다.

객체 대시보드의 열에는 각 객체에 대한 다음 정보가 표시됩니다.

이름

객체의 이름.

유형

객체의 유형. 유효한 값에는 데이터 소스, ML 모델, 평가 및 배치 예측이 포함됩니다.

> **Note**
>
> 모델이 실시간 예측을 지원하도록 설정되었는지 확인하려면 이름 또는 모델 ID를 선택하여 ML 모델 요약 페이지로 이동합니다.

ID

객체의 ID.

상태

객체의 상태. 값에는 보류 중, 진행 중, 완료됨, 실패가 포함됩니다. 상태가 실패인 경우 데이터를 확인하고 다시 시도하세요.

생성 시간

Amazon ML에서 이 객체 생성을 완료한 날짜와 시간.

완료 시간

Amazon ML에서 이 객체를 생성하는 데 걸린 시간. 모델 완료 시간을 사용하여 새 모델의 학습 시간을 추정할 수 있습니다.

데이터 소스 ID

모델 및 평가와 같이 데이터 소스를 사용하여 만든 객체의 경우 데이터 소스의 ID. 데이터 소스를 삭제하면 해당 데이터 소스로 만든 ML 모델을 사용하여 더 이상 예측을 생성할 수 없습니다.

열 헤더 옆에 있는 이중 삼각형 아이콘을 선택하면 열을 기준으로 정렬할 수 있습니다.

객체 나열(API)

ML API에서는 다음 작업을 사용하여 유형별로 객체를 나열할 수 있습니다.

- `DescribeDataSources`
- `DescribeMLModels`
- `DescribeEvaluations`
- `DescribeBatchPredictions`

각 작업에는 긴 객체 목록을 통해 필터링, 정렬 및 페이지 매기기를 위한 파라미터가 포함되어 있습니다. API를 통해 액세스할 수 있는 객체 수에는 제한이 없습니다. 목록의 크기를 제한하려면 `Limit` 파라미터를 사용합니다. 이 파라미터는 최대값이 100이 될 수 있습니다.

`Describe*` 명령에 대한 API 응답에는 해당하는 경우 페이지 매김 토큰(`nextPageToken`)과 각 객체에 대한 간략한 설명이 포함되어 있습니다. 객체 설명에는 객체 유형별 세부 정보를 포함하여 콘솔에 표시되는 각 객체 유형에 대한 동일한 정보가 포함되어 있습니다.

> **Note**
>
> 지정된 한도보다 적은 수의 객체가 응답에 포함되더라도 더 많은 결과를 사용할 수 있다는 `nextPageToken` 메시지가 포함될 수 있습니다. 항목이 0개인 응답에도 `nextPageToken`가 포함될 수 있습니다.

자세한 내용은 ML API 참조 단원을 참조하세요.

객체 설명 검색

콘솔이나 API를 통해 모든 객체에 대한 세부 설명을 볼 수 있습니다.

콘솔의 세부 설명

콘솔에서 설명을 보려면 특정 유형의 객체(데이터 소스, ML 모델, 평가 또는 배치 예측)의 목록으로 이동합니다. 그런 다음 목록을 탐색하거나 이름 또는 ID를 검색하여 표에서 해당 객체에 해당하는 행을 찾습니다.

API의 세부 설명

객체 유형마다 Amazon ML 객체의 전체 세부 정보를 검색하는 다음과 같은 작업이 있습니다.

- GetDataSource
- GetMLModel
- GetEvaluation
- GetBatchPrediction

각 작업에는 정확히 두 개의 파라미터, 즉 객체 ID와 Verbose라는 부울 플래그가 사용됩니다. Verbose가 true로 설정된 직접 호출에는 객체에 대한 추가 세부 정보가 포함되므로 지연 시간이 길어지고 응답 시간이 길어집니다. Verbose 플래그를 설정하면 어떤 필드가 포함되는지 알아보려면 ML API 참조 단원을 참조하세요.

객체 업데이트

각 객체 유형에는 ML 객체의 세부 정보를 업데이트하는 작업이 있습니다(ML API 참조 단원 참조).

- UpdateDataSource
- UpdateMLModel
- UpdateEvaluation
- UpdateBatchPrediction

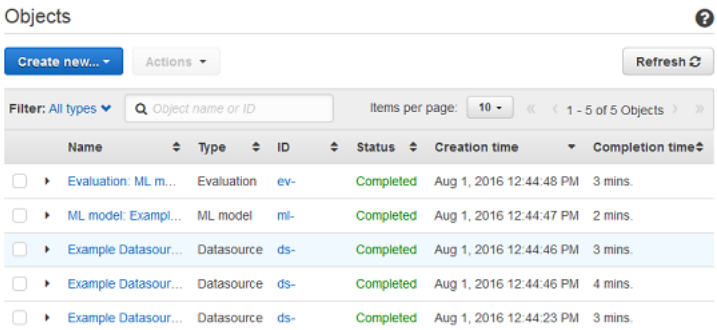
각 작업에는 업데이트되는 객체를 지정하는 객체 ID가 필요합니다. 모든 객체의 이름을 업데이트할 수 있습니다. 데이터 소스, 평가, 배치 예측에 대한 객체의 다른 속성은 업데이트할 수 없습니다. ML 모델

의 경우 ScoreThreshold 필드를 업데이트할 수 있습니다. 단, ML 모델에 연결된 실시간 예측 엔드포인트가 없어야 합니다.

객체 삭제

데이터 소스, ML 모델, 평가 및 배치 예측이 더 이상 필요하지 않으면 해당 데이터를 삭제해도 됩니다. 사용 완료 후 배치 예측 외에 Amazon ML 객체를 보관하는 데 드는 추가 비용은 없지만, 객체를 삭제하면 작업 공간이 깔끔해지고 관리하기가 더 쉬워집니다. Amazon Machine Learning(Amazon ML) 콘솔 또는 API를 사용하면 단일 또는 여러 객체를 삭제할 수 있습니다.

> **Warning**
>
> Amazon ML 객체를 삭제하면 그 효과는 즉각적이고 영구적이며 되돌릴 수 없습니다.



객체 삭제(콘솔)

Amazon ML 콘솔을 사용하면 모델을 포함하여 객체를 삭제할 수 있습니다. 모델을 삭제하는 데 사용하는 절차는 모델을 사용하여 실시간 예측을 생성하는지 여부에 따라 달라집니다. 실시간 예측을 생성하는 데 사용되는 모델을 삭제하려면 먼저 실시간 엔드포인트를 삭제합니다.

Amazon ML 객체를 삭제하려면(콘솔)

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/machinelearning/ Amazon Machine Learning 콘솔을 엽니다.
2. 삭제할 Amazon ML 객체를 선택합니다. 객체를 두 개 이상 선택하려면 Shift 키를 사용합니다. 선택한 모든 객체를 선택 취소하려면

   또는

버튼을 사용합니다.

3. 작업에 대해 삭제를 선택합니다.
4. 대화 상자에서 삭제를 선택하여 모델을 삭제합니다.

실시간 엔드포인트가 있는 Amazon ML 모델을 삭제하려면(콘솔)

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/machinelearning/ Amazon Machine Learning 콘솔을 엽니다.
2. 삭제할 모델을 선택합니다.
3. 작업에 대해 실시간 엔드포인트 삭제를 선택합니다.
4. 삭제를 선택하여 엔드포인트를 삭제합니다.
5. 모델을 다시 선택합니다.
6. 작업에 대해 삭제를 선택합니다.
7. 삭제를 선택하여 모델을 삭제합니다.

객체 삭제(API)

다음 API 직접 호출을 사용하여 Amazon ML 객체를 삭제할 수 있습니다.

- `DeleteDataSource` - 파라미터 `DataSourceId`를 사용합니다.
- `DeleteMLModel` - 파라미터 `MLModelId`를 사용합니다.
- `DeleteEvaluation` - 파라미터 `EvaluationId`를 사용합니다.
- `DeleteBatchPrediction` - 파라미터 `BatchPredictionId`를 사용합니다.

자세한 내용은 머신 러닝 API 참조 단원을 참조하세요.

Amazon CloudWatch 지표를 사용한 Amazon ML 모니터링

Amazon ML은 사용자가 ML 모델에 대한 사용 통계를 수집하고 분석할 수 있도록 Amazon CloudWatch로 지표를 자동으로 전송합니다. 예를 들어, 배치 및 실시간 예측을 추적하기 위해 RequestMode 차원에 따라 PredictCount 지표를 모니터링할 수 있습니다. 지표는 5분마다 자동 수집되어 Amazon CloudWatch로 전송됩니다. Amazon CloudWatch 콘솔, AWS CLI 또는 AWS SDK를 사용하여 이러한 지표를 모니터링할 수 있습니다.

CloudWatch를 통해 보고되는 Amazon ML 지표에는 요금이 부과되지 않습니다. 지표에 대한 경보를 설정하면 표준 CloudWatch 요금이 청구됩니다.

자세한 내용은 CloudWatch 개발자 안내서의 CloudWatch 네임스페이스, 차원 및 지표 참조의 ML 지표 목록을 참조하세요.

를 사용하여 Amazon ML API 호출 로깅 AWS CloudTrail

Amazon Machine Learning(Amazon ML)은 Amazon ML에서 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스와 통합됩니다. CloudTrail은 Amazon ML에 대한 모든 API 직접 호출을 이벤트로 캡처합니다. Amazon ML 콘솔로부터의 호출과 Amazon ML API 작업에 대한 코드 호출이 호출됩니다. 추적을 생성하면 Amazon ML 이벤트를 포함한 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 전송할 수 있습니다. 트레일을 구성하지 않은 경우에도 CloudTrail 콘솔의 이벤트 기록에서 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 Amazon ML에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

구성 및 사용 방법을 포함하여 CloudTrail에 대한 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하세요.

CloudTrail의 Amazon ML 정보

AWS 계정을 생성할 때 계정에서 CloudTrail이 활성화됩니다. 지원되는 이벤트 활동이 ML에서 발생하면, 해당 활동이 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최근 이벤트를 보고 검색하고 다운로드할 수 있습니다. 자세한 정보는 CloudTrail 이벤트 기록을 사용하여 이벤트 보기를 참조하세요.

Amazon ML에 대한 이벤트를 포함하여 AWS 계정의 이벤트를 지속적으로 기록하려면 추적을 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 추적을 생성하면 기본적으로 모든 리전에 추적이 적용됩니다. 추적은 AWS 파티션의 모든 리전에서 이벤트를 로깅하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 조치를 취하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하세요.

- 트레일 생성 개요
- CloudTrail 지원 서비스 및 통합
- CloudTrail에서 Amazon SNS 알림 구성
- 여러 리전으로부터 CloudTrail 로그 파일 받기 및 여러 계정으로부터 CloudTrail 로그 파일 받기

Amazon ML은 CloudTrail 로그 파일의 이벤트로 다음 작업의 로깅을 지원합니다.

- AddTags

- CreateBatchPrediction
- CreateDataSourceFromRDS
- CreateDataSourceFromRedshift
- CreateDataSourceFromS3
- CreateEvaluation
- CreateMLModel
- CreateRealtimeEndpoint
- DeleteBatchPrediction
- DeleteDataSource
- DeleteEvaluation
- DeleteMLModel
- DeleteRealtimeEndpoint
- DeleteTags
- DescribeTags
- UpdateBatchPrediction
- UpdateDataSource
- UpdateEvaluation
- UpdateMLModel

다음 Amazon ML 작업은 보안 인증 정보를 포함하는 있는 요청 파라미터를 사용합니다. 이러한 요청이 CloudTrail로 전송되기 전에 보안 인증 정보는 세 개의 별표("***")로 교체됩니다.

- CreateDataSourceFromRDS
- CreateDataSourceFromRedshift

CloudTrail 콘솔을 통해 다음 CloudTrail 작업을 수행할 때 속성 `ComputeStatistics`은 CloudTrail 로그의 `RequestParameters` 구성 요소에 포함되지 않습니다.

- CreateDataSourceFromRedshift
- CreateDataSourceFromS3

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 대한 정보가 포함됩니다. 자격 증명을 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청이 루트 또는 AWS Identity and Access Management (IAM) 사용자 자격 증명으로 이루어졌는지 여부입니다.
- 역할 또는 페더레이션 사용자에 대한 임시 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청을 했는지 여부입니다.

자세한 내용은 CloudTrail userIdentity 요소 단원을 참조하세요.

예: Amazon ML 로그 파일 항목

추적이란 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 전송할 수 있게 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 모든 소스로부터의 단일 요청을 나타내며 요청 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출의 주문 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

다음은 작업을 보여주는 CloudTrail 로그 항목을 나타내는 예제입니다.

```
{
    "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 객체에 태그 지정

태그를 사용하여 객체에 메타데이터를 할당하면 Amazon Machine Learning(Amazon ML) 객체를 정리하고 관리할 수 있습니다. 태그는 객체에 대해 정의하는 키-값 쌍입니다.

태그를 사용하면 Amazon ML 객체를 정리하고 관리하는 것 외에도 범주를 분류하고 AWS 비용을 추적할 수 있습니다. ML 모델을 비롯하여 AWS 객체에 태그를 적용하면, AWS 비용 할당 보고서에 태그별로 집계된 사용 내역 및 비용이 포함됩니다. 비즈니스 범주를 나타내는 태그(예: 비용 센터, 애플리케이션 이름 또는 소유자)를 적용하면 여러 서비스에 대한 비용을 정리할 수 있습니다. 자세한 내용은 AWS Billing 사용 설명서의 사용자 지정 결제 보고서에 비용 할당 태그 사용 단원을 참조하세요.

내용

태그 기본 사항

태그를 사용하여 객체를 분류하면 객체를 더 쉽게 관리할 수 있습니다. 예를 들어 용도, 소유자 또는 환경을 기준으로 객체를 분류할 수 있습니다. 그런 다음, 소유자 및 연관된 애플리케이션에 따라 모델을 추적하는 데 도움이 되는 태그 세트를 정의할 수도 있습니다. 다음은 몇 가지 예제입니다.

- 프로젝트: 프로젝트 이름
- 소유자: 이름
- 용도: 마케팅 예측
- 애플리케이션: 애플리케이션 이름
- 환경: 프로덕션

Amazon ML 콘솔 또는 API를 사용하여 다음 작업을 완료할 수 있습니다.

- 객체에 태그 추가
- 객체의 태그 보기
- 객체의 태그 편집

- 객체에서 태그 삭제

기본적으로 Amazon ML 객체에 적용된 태그는 해당 객체를 사용하여 생성된 객체에 복사됩니다. 예를 들어 Amazon Simple Storage Service(Amazon S3) 데이터 소스에 “마케팅 비용: 타겟 마케팅 캠페인” 태그가 있는 경우 해당 데이터 소스를 사용하여 만든 모델에도 모델에 대한 평가와 마찬가지로 “마케팅 비용: 타겟 마케팅 캠페인” 태그가 표시됩니다. 이렇게 하면 태그를 사용하여 마케팅 캠페인에 사용된 모든 객체와 같은 관련 객체를 추적할 수 있습니다. 태그 소스(예: “마케팅 비용: 타겟 마케팅 캠페인” 태그가 있는 모델)와 “마케팅 비용: 타겟 마케팅 고객” 태그가 있는 데이터 소스 간에 충돌이 있는 경우 Amazon ML은 모델의 태그를 적용합니다.

태그 제한

태그에 적용되는 제한은 다음과 같습니다.

기본 제한:

- 객체당 최대 태그 수는 50개입니다.
- 태그 키와 값은 대소문자를 구분합니다.
- 삭제된 객체에 대한 태그는 변경하거나 편집할 수 없습니다.

태그 키 제한:

- 각 태그 키는 고유해야 합니다. 이미 사용 중인 키를 가진 태그를 추가하면 해당 객체에 대한 기존 키-값 쌍에 새 태그가 덮어쓰기 됩니다.
- 태그 키에 `aws:` 접두사는 사용하지 마십시오. 이 접두사는 AWS용으로 예약되어 있습니다. AWS는 이 접두사로 시작되는 태그를 생성하지만, 사용자는 이를 편집 또는 삭제할 수 없습니다.
- 태그 키의 길이는 유니코드 1~128자여야 합니다.
- 태그 키의 문자로는 유니코드 문자, 숫자, 공백 그리고 `_ . / = + - @` 같은 특수 문자가 허용됩니다.

태그 값 제한:

- 태그 값의 길이는 유니코드 0~255자여야 합니다.
- 태그 값은 공백 상태로 둘 수 있습니다. 아니면 유니코드 문자, 숫자, 공백 그리고 `_ . / = + - @` 같은 특수 문자를 사용할 수 있습니다.

Amazon ML 객체에 태그 지정(콘솔)

Amazon ML 콘솔을 사용하여 태그를 보고, 추가하고, 편집하고, 삭제할 수 있습니다.

객체의 태그를 보려면(콘솔)

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/machinelearning/ Amazon Machine Learning 콘솔을 엽니다.
2. 탐색 모음에서 지역 선택기를 확장하고 지역을 선택합니다.
3. 객체 페이지에서 객체를 선택합니다.
4. 선택한 객체의 태그 섹션으로 스크롤합니다. 해당 객체의 태그는 섹션 하단에 나열됩니다.

객체에 태그를 추가하려면(콘솔)

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/machinelearning/ Amazon Machine Learning 콘솔을 엽니다.
2. 탐색 모음에서 지역 선택기를 확장하고 지역을 선택합니다.
3. 객체 페이지에서 객체를 선택합니다.
4. 선택한 객체의 태그 섹션으로 스크롤합니다. 해당 객체의 태그는 섹션 하단에 나열됩니다.
5. 태그 추가 또는 편집을 선택합니다.
6. 태그 추가 아래에서, 키 필드에서 태그 키를 지정하고, 선택적으로 값 필드에서 태그 값을 지정한 다음 변경 사항 적용을 선택합니다.

   변경 사항 적용 버튼이 활성화되지 않았다면 지정한 태그 키 또는 태그 값이 태그 제한 사항을 충족하지 않는 것입니다. 자세한 내용은 태그 제한 단원을 참조하십시오.
7. 태그 섹션의 목록에서 새 태그를 보려면 페이지를 새로 고칩니다.

태그를 편집하려면(콘솔)

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/machinelearning/ Amazon Machine Learning 콘솔을 엽니다.
2. 탐색 모음에서 지역 선택기를 확장하고 지역을 선택합니다.
3. 객체 페이지에서 객체를 선택합니다.
4. 선택한 객체의 태그 섹션으로 스크롤합니다. 해당 객체의 태그는 섹션 하단에 나열됩니다.
5. 태그 추가 또는 편집을 선택합니다.

6. 적용된 태그 아래에서 값 필드의 태그 값을 편집한 다음 변경 사항 적용을 선택합니다.

   변경 사항 적용 버튼이 활성화되지 않았다면 지정한 태그 값이 태그 제한 사항과 일치하지 않기 때문입니다. 자세한 내용은 태그 제한 단원을 참조하세요.
7. 태그 섹션의 목록에서 업데이트된 태그를 보려면 페이지를 새로 고칩니다.

객체에서 태그를 삭제하려면(콘솔)

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/machinelearning/ Amazon Machine Learning 콘솔을 엽니다.
2. 탐색 모음에서 지역 선택기를 확장하고 지역을 선택합니다.
3. 객체 페이지에서 객체를 선택합니다.
4. 선택한 객체의 태그 섹션으로 스크롤합니다. 해당 객체의 태그는 섹션 하단에 나열됩니다.
5. 태그 추가 또는 편집을 선택합니다.
6. 적용된 태그 아래에서 삭제하려는 태그를 선택한 다음 변경 사항 적용을 선택합니다.

Amazon ML 객체에 태그 지정(API)

Amazon ML API를 사용하여 태그를 추가, 나열 및 삭제할 수 있습니다. 예제는 다음 설명서를 참조하세요.

AddTags

지정된 객체의 태그를 추가 또는 편집합니다.

DescribeTags

지정된 객체의 태그를 나열합니다.

DeleteTags

지정된 객체에서 태그를 삭제합니다.

Amazon Machine Learning 참조

주제

Amazon S3에서 데이터를 읽을 수 있는 권한을 Amazon ML에 부여

Amazon S3의 입력 데이터에서 데이터 소스 객체를 생성하려면 입력 데이터가 저장되는 S3 위치에 대해 Amazon ML에 다음 권한을 부여해야 합니다.

- S3 버킷 및 접두사에 대한 GetObject 권한.
- S3 버킷에 대한 ListBucket 권한. 다른 작업과 달리 ListBucket에는 (접두사가 아닌) 버킷 전체 권한을 부여해야 합니다. 하지만 Condition 절을 사용하여 특정 접두사로 권한 범위를 지정할 수 있습니다.

Amazon ML 콘솔을 사용하여 데이터 소스를 생성하는 경우 이러한 권한을 버킷에 추가할 수 있습니다. 마법사의 단계를 완료하면 추가할 것인지 확인하는 메시지가 표시됩니다. 다음 예제 정책에서는 ML이 샘플 위치 s3://*examplebucket*/*exampleprefix*에서 데이터를 읽을 수 있는 권한을 부여하는 동시에 ListBucket 권한의 범위를 *exampleprefix* 입력 경로로만 지정하는 방법을 보여줍니다.

JSON

```
{
    "Version":"2012-10-17",
    "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:*"
                }
            }
        }
    ]
}
```

이 정책을 데이터에 적용하려면 데이터가 저장되는 S3 버킷과 연결된 정책 설명을 편집해야 합니다.

S3 버킷의 권한 정책을 편집하려면(이전 콘솔 사용)

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/s3/ Amazon S3 콘솔을 엽니다.
2. 데이터가 있는 버킷 이름을 선택합니다.
3. 속성을 선택합니다.
4. 버킷 정책 편집을 선택합니다.
5. 위에 표시된 정책을 입력하고 필요에 맞게 사용자 지정한 다음 저장을 선택합니다.
6. 저장을 선택합니다.

S3 버킷의 권한 정책을 편집하려면(새 콘솔 사용)

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/s3/ Amazon S3 콘솔을 엽니다.
2. 버킷 이름을 선택한 후 권한을 선택합니다.
3. 버킷 정책을 선택합니다.
4. 위에 표시된 정책을 입력하고 필요에 맞게 사용자 지정합니다.
5. 저장을 선택합니다.

Amazon S3에 예측을 출력할 수 있는 권한을 Amazon ML에 부여

배치 예측 작업의 결과를 Amazon S3에 출력하려면 Amazon ML에 출력 위치에 대한 다음 권한을 부여해야 합니다. 이 권한은 배치 예측 생성 작업에 대한 입력으로 제공됩니다.

- S3 버킷 및 접두사에 대한 GetObject 권한.
- S3 버킷 및 접두사에 대한 PutObject 권한.
- S3 버킷과 접두사에 대한 PutObjectAcl.
  - ML은 객체 생성 후 미리 준비된 ACL 버킷 소유자 전체 제어 권한을 계정에 부여할 수 있으려면 이 권한이 필요합니다.
- S3 버킷에 대한 ListBucket 권한. 다른 작업과 달리 ListBucket에는 (접두사가 아닌) 버킷 전체 권한을 부여해야 합니다. 하지만 Condition 절을 사용하여 특정 접두사로 권한 범위를 지정할 수 있습니다.

Amazon ML 콘솔을 사용하여 배치 예측 요청을 생성하는 경우 이러한 권한을 버킷에 추가할 수 있습니다. 마법사의 단계를 완료하면 추가 여부를 확인하라는 메시지가 표시됩니다.

다음 예제 정책에서는 ML이 샘플 위치 s3://examplebucket/exampleprefix 에 데이터를 쓸 수 있는 권한을 부여하는 동시에 ListBucket 권한의 범위를 exampleprefix 입력 경로로만 지정하고 ML이 출력 접두사에 객체 ACL을 설정할 수 있는 권한을 부여하는 방법을 보여줍니다.

JSON

```
{
    "Version":"2012-10-17",
    "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": {
                    "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:*"
                }
            }
        }
    ]
}
```

이 정책을 데이터에 적용하려면 데이터가 저장되는 S3 버킷과 관련된 정책 설명을 편집해야 합니다.

S3 버킷의 권한 정책을 편집하려면(이전 콘솔 사용)

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/s3/ Amazon S3 콘솔을 엽니다.
2. 데이터가 있는 버킷 이름을 선택합니다.
3. 속성을 선택합니다.
4. 버킷 정책 편집을 선택합니다.
5. 위에 표시된 정책을 입력하고 필요에 맞게 사용자 지정한 다음 저장을 선택합니다.
6. 저장을 선택합니다.

S3 버킷의 권한 정책을 편집하려면(새 콘솔 사용)

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/s3/ Amazon S3 콘솔을 엽니다.
2. 버킷 이름을 선택한 후 권한을 선택합니다.
3. 버킷 정책을 선택합니다.
4. 위에 표시된 정책을 입력하고 필요에 맞게 사용자 지정합니다.
5. 저장을 선택합니다.

Amazon ML 리소스에 대한 액세스 제어 - IAM 사용

AWS Identity and Access Management (IAM)를 사용하면 사용자의 AWS 서비스 및 리소스에 대한 액세스를 안전하게 제어할 수 있습니다. IAM을 사용하면 사용자, 그룹 및 역할을 생성 및 관리하고 권한을 사용하여 리소스에 대한 액세스를 허용 및 거부할 수 있습니다. IAM과 Amazon Machine Learning(Amazon ML)을 함께 사용하면 조직 내 사용자별로 특정 리소스의 사용 권한을 제어할 수 있습니다.

IAM을 사용하여 다음을 수행할 수 있습니다.

- 계정의 사용자와 그룹 생성
- 계정 사용자 각각에 고유한 보안 인증 할당
- 작업 수행 시 각 사용자의 리소스 사용 권한 제어
- 계정의 사용자 간에 리소스를 쉽게 공유
- 계정에 적용할 규칙 생성 및 이들 역할을 수임할 수 있는 사용자나 서비스를 규정하기 위해 이들 역할에 대한 권한 관리
- IAM에서 역할을 생성하고 권한을 관리하여 역할을 맡는 엔티티 또는 서비스가 수행할 수 있는 작업을 제어할 수 있습니다. 어떤 개체가 해당 역할을 맡도록 허용된 개체를 정의할 수도 있습니다.

조직에 이미 IAM 자격 증명이 있으면 이를 사용해 리소스를 사용하는 작업 권한을 부여할 수 있습니다.

IAM에 대한 자세한 내용은 IAM 사용 설명서를 참조하세요.

IAM 정책 구문

IAM 정책은 하나 이상의 구문으로 구성된 JSON 문서입니다. 각 구문의 구조는 다음과 같습니다.

```
{
    "Statement":[{
        "Effect":"effect",
        "Action":"action",
        "Resource":"arn",
        "Condition":{
            "condition operator":{
                "key":"value"
            }
        }
    }]
}
```

정책 구문은 다음 요소로 구성됩니다.

- 효과: 명령문의 뒷부분에서 지정할 리소스 및 API 작업 사용 권한을 제어합니다. 유효 값은 `Allow` 및 `Deny`입니다. 기본적으로 IAM 사용자에게는 리소스 및 API 작업을 사용할 권한이 없으므로 모든 요청이 거부됩니다. 명시적 `Allow`은 기본 설정 보다 우선합니다. 명시적 `Deny`은 어떤 `Allows` 보다 우선합니다.
- 작업: 권한을 부여하거나 거부할 특정 API 작업입니다.
- 리소스: 작업의 영향을 받는 리소스입니다. 구문에서 리소스를 지정하려면 Amazon 리소스 이름(ARN)을 사용합니다.
- 조건(선택 사항): 정책이 적용되는 시기를 제어합니다.

IAM 정책을 간단하게 생성하고 관리하려면 IAM 정책 생성기 및 IAM 정책 시뮬레이터를 사용합니다.

Amazon MLAmazon ML에 대한 IAM 정책 작업 지정

IAM 정책 구문에서 IAM을 지원하는 모든 서비스의 API 작업을 지정할 수 있습니다. ML API 작업에 대한 정책 구문을 생성할 때는 다음 예와 같이 API 작업의 이름 앞에 `machinelearning:`을 추가하세요.

- `machinelearning:CreateDataSourceFromS3`
- `machinelearning:DescribeDataSources`
- `machinelearning:DeleteDataSource`
- `machinelearning:GetDataSource`

단일 구문에서 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": ["machinelearning:action1", "machinelearning:action2"]
```

와일드카드를 사용하여 여러 작업을 지정할 수도 있습니다. 예를 들어 다음과 같이 이름이 "Get"으로 시작되는 모든 작업을 지정할 수 있습니다.

```
"Action": "machinelearning:Get*"
```

모든 Amazon ML 작업을 지정하려면 다음과 같이 * 와일드카드를 사용합니다.

```
"Action": "machinelearning:*"
```

ML API 작업의 전체 목록은 머신 러닝 API 참조 단원을 참조하세요.

IAM 정책에서 Amazon ML 리소스용 ARN 지정

IAM 정책 구문은 하나 이상의 리소스에 적용됩니다. ARN으로 정책 리소스를 지정합니다.

Amazon ML 리소스의 ARN을 지정하려면 다음 형식을 사용합니다.

최대 1GB"리소스": `arn:aws:machinelearning:region:account:resource-type/identifier`

다음 예제에서는 공통 ARN을 지정하는 방법을 보여줍니다.

데이터 소스 ID: `my-s3-datasource-id`

```
"Resource":
arn:aws:machinelearning:<region>:<your-account-id>:datasource/my-s3-datasource-id
```

ML 모델 ID: `my-ml-model-id`

```
"Resource":
arn:aws:machinelearning:<region>:<your-account-id>:mlmodel/my-ml-model-id
```

배치 예측 ID: `my-batchprediction-id`

```
"Resource":
arn:aws:machinelearning:<region>:<your-account-id>:batchprediction/my-batchprediction-
id
```

평가 ID: `my-evaluation-id`

```
"Resource": arn:aws:machinelearning:<region>:<your-account-id>:evaluation/my-
evaluation-id
```

Amazon SNS에 대한 정책 예제

예제 1: 사용자가 기계 학습 리소스의 메타데이터를 읽을 수 있도록 허용

다음 정책은 사용자나 그룹이 지정된 리소스에서 DescribeDataSources, DescribeMLModels, DescribeBatchPredictions, DescribeEvaluations, GetDataSource, GetMLModel, GetBatchPrediction 및 GetEvaluation 작업을 수행하여 데이터 소스, ML 모델, 배치 예측 및 평가의 메타데이터를 읽을 수 있도록 허용합니다. Describe * 작업 권한은 특정 리소스로 제한할 수 없습니다.

JSON

```
{ "Version":"2012-10-17",        "Statement": [ { "Effect": "Allow", "Action": [
    "machinelearning:Get*" ], "Resource": [
        "arn:aws:machinelearning:us-east-1:123456789012:datasource/S3-DS-ID1",
        "arn:aws:machinelearning:us-east-1:123456789012:datasource/REDSHIFT-DS-
ID1",
        "arn:aws:machinelearning:us-east-1:123456789012:mlmodel/ML-MODEL-ID1",
        "arn:aws:machinelearning:us-east-1:123456789012:batchprediction/BP-ID1",
        "arn:aws:machinelearning:us-east-1:123456789012:evaluation/EV-ID1"
    ] }, { "Effect": "Allow", "Action": [ "machinelearning:Describe*" ],
 "Resource": [ "*" ] } ]
    }
```

예제 2: 사용자가 기계 학습 리소스를 만들 수 있도록 허용

다음 정책은 사용자나 그룹이 `CreateDataSourceFromS3`, `CreateDataSourceFromRedshift`, `CreateDataSourceFromRDS`, `CreateMLModel`, `CreateBatchPrediction` 및 `CreateEvaluation` 작업을 수행하여 기계 학습 데이터 소스, ML 모델, 배치 예측 및 평가를 생성할 수 있도록 허용합니다. 이들 작업의 권한을 특정 리소스로 제한할 수 없습니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "machinelearning:CreateDataSourceFrom*",
                "machinelearning:CreateMLModel",
                "machinelearning:CreateBatchPrediction",
                "machinelearning:CreateEvaluation"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

예제 3: 사용자가 실시간 엔드포인트를 생성 및 삭제하고 ML 모델에서 실시간 예측을 수행하도록 허용

다음 정책은 사용자나 그룹이 실시간 엔드포인트를 생성 및 삭제하고, 해당 모델에서 `CreateRealtimeEndpoint`, `DeleteRealtimeEndpoint` 및 `Predict` 작업을 수행하여 특정 ML 모델에 대한 실시간 예측을 수행할 수 있도록 허용합니다.

JSON

```
{ "Version":"2012-10-17",        "Statement": [ { "Effect": "Allow", "Action": [
    "machinelearning:CreateRealtimeEndpoint",
 "machinelearning:DeleteRealtimeEndpoint",
    "machinelearning:Predict" ], "Resource": [
        "arn:aws:machinelearning:us-east-1:123456789012:mlmodel/ML-MODEL"
    ] } ] }
```

예제 4: 사용자가 특정 리소스를 업데이트하고 삭제하도록 허용

다음 정책은 사용자나 그룹이 계정의 해당 리소스에서 `UpdateDataSource`, `UpdateMLModel`, `UpdateBatchPrediction`, `UpdateEvaluation`, `DeleteDataSource`, `DeleteMLModel`,

DeleteBatchPrediction 및 DeleteEvaluation 작업을 수행할 권한을 부여하여 계정의 특정 리소스를 업데이트하고 삭제할 수 있도록 허용합니다.

JSON

```
{ "Version":"2012-10-17",        "Statement": [ { "Effect": "Allow", "Action": [
    "machinelearning:Update*", "machinelearning:DeleteDataSource",
 "machinelearning:DeleteMLModel",
    "machinelearning:DeleteBatchPrediction",
 "machinelearning:DeleteEvaluation" ], "Resource": [
        "arn:aws:machinelearning:us-east-1:123456789012:datasource/S3-DS-ID1",
        "arn:aws:machinelearning:us-east-1:123456789012:datasource/REDSHIFT-DS-
ID1",
        "arn:aws:machinelearning:us-east-1:123456789012:mlmodel/ML-MODEL-ID1",
        "arn:aws:machinelearning:us-east-1:123456789012:batchprediction/BP-ID1",
        "arn:aws:machinelearning:us-east-1:123456789012:evaluation/EV-ID1"
    ] } ] }
```

예제 5: 모든 ML 작업 허용

다음 정책은 사용자나 그룹이 모든 Amazon ML 작업을 사용할 수 있도록 허용합니다. 이 정책은 모든 기계 학습 리소스에 대한 전체 액세스 권한을 부여하므로 관리자로만 제한해야 합니다.

JSON

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

교차 서비스 혼동된 대리인 방지

혼동된 대리자 문제는 작업을 수행할 권한이 없는 엔터티가 권한이 더 많은 엔터티에게 작업을 수행하도록 강요할 수 있는 보안 문제입니다. 에서 AWS교차 서비스 가장은 혼동된 대리자 문제를 초래할 수 있습니다. 교차 서비스 가장은 한 서비스(직접 호출하는 서비스)가 다른 서비스(직접 호출되는 서비스)를 직접 호출할 때 발생할 수 있습니다. 직접 호출하는 서비스는 다른 고객의 리소스에 대해 액세스 권한이 없는 방식으로 작동하게 권한을 사용하도록 조작될 수 있습니다. 이를 방지하기 위해 AWS 에서는 계정의 리소스에 대한 액세스 권한이 부여된 서비스 위탁자를 사용하여 모든 서비스에 대한 데이터를 보호하는 데 도움이 되는 도구를 제공합니다.

Amazon 기계 학습이 리소스에 다른 서비스를 제공하는 권한을 제한하려면 리소스 정책에서 `aws:SourceArn` 및 `aws:SourceAccount` 전역 조건 컨텍스트 키를 사용하는 것이 좋습니다. 만약 `aws:SourceArn` 값에 S3 버킷 ARN과 같은 계정 ID가 포함되어 있지 않은 경우, 권한을 제한하려면 두 전역 조건 컨텍스트 키를 모두 사용해야 합니다. 두 전역 조건 컨텍스트 키와 계정을 포함한 `aws:SourceArn` 값을 모두 사용하는 경우, `aws:SourceAccount` 값 및 `aws:SourceArn` 값의 계정은 동일한 정책 명령문에서 사용할 경우 반드시 동일한 계정 ID를 사용해야 합니다. 하나의 리소스만 교차 서비스 액세스와 연결되도록 허용하려는 경우 `aws:SourceArn`을 사용하세요. 해당 계정의 모든 리소스가 교차 서비스 사용과 연결되도록 허용하려는 경우 `aws:SourceAccount`을(를) 사용합니다.

혼동된 대리자 문제로부터 보호하는 가장 효과적인 방법은 리소스의 전체 ARN이 포함된 `aws:SourceArn` 전역 조건 컨텍스트 키를 사용하는 것입니다. 리소스의 전체 ARN을 모를 경우 또는 여러 리소스를 지정하는 경우, ARN의 알 수 없는 부분에 대해 와일드카드(*)를 포함한 `aws:SourceArn`전역 조건 컨텍스트 키를 사용합니다. 예제: `arn:aws:servicename:*:123456789012:*`.

다음 예에서는 S3 버킷에서 데이터를 읽을 때 ML에서 `aws:SourceArn` 및 `aws:SourceAccount` 전역 조건 컨텍스트 키를 사용하여 혼동된 대리자 문제를 방지하는 방법을 보여줍니다.

JSON

```
{
    "Version":"2012-10-17",
    "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:*"
                }
            }
        }
    ]
}
```

비동기 작업의 종속성 관리

Amazon ML의 배치 작업은 성공적으로 완료되기 위해 다른 작업에 의존합니다. 이러한 종속성을 관리하기 위해 Amazon ML은 종속성이 있는 요청을 식별하고 작업이 완료되었는지 확인합니다. 작업이 완료되지 않은 경우 Amazon ML은 해당 요청이 의존하는 작업이 완료될 때까지 초기 요청을 따로 보관합니다.

배치 작업 간에는 몇 가지 종속성이 있습니다. 예를 들어 ML 모델을 만들려면 먼저 ML 모델을 학습시킬 수 있는 데이터 소스를 만들어야 합니다. Amazon ML은 사용 가능한 데이터 소스가 없는 경우 ML 모델을 학습시킬 수 없습니다.

하지만 Amazon ML은 비동기 작업에 대한 종속성 관리를 지원합니다. 예를 들어, 데이터 소스에서 ML 모델을 학습시키라는 요청을 보내기 전에 데이터 통계가 계산될 때까지 기다릴 필요가 없습니다. 대신, 데이터 소스가 생성되자마자 데이터 소스를 사용하여 ML 모델을 학습시키라는 요청을 보낼 수 있습니다. Amazon ML은 데이터 소스 통계가 계산될 때까지 실제로 학습 작업을 시작하지 않습니다. CreateMLModel 요청은 통계가 계산될 때까지 대기열에 저장됩니다. 계산이 완료되면 Amazon ML은 즉시 CreateMLModel 작업을 실행하려고 시도합니다. 마찬가지로, 학습이 완료되지 않은 ML 모델에 대한 배치 예측 및 평가 요청을 전송할 수 있습니다.

다음 표에는 다양한 Amazon ML 작업을 진행하는 데 필요한 요구 사항이 나와 있습니다.

이를 위해서는...	...이(가) 있어야 합니다
ML 모델 생성(createMLModel)	계산된 데이터 통계가 포함된 데이터 소스
배치 예측 생성(createBatchPrediction)	데이터 소스 ML 모델
배치 평가 생성(createBatchEvaluation)	데이터 소스 ML 모델

요청 상태 확인

요청을 제출하면 Amazon Machine Learning(Amazon ML) API를 사용하여 요청 상태를 확인할 수 있습니다. 예를 들어 `createMLModel` 요청을 제출하면 `describeMLModel` 직접 호출을 사용하여 상태를 확인할 수 있습니다. Amazon ML은 다음 상태 중 하나로 응답합니다.

상태	정의
PENDING	Amazon ML에서 요청을 검증하는 중입니다. 또는

상태	정의
	Amazon ML이 요청을 실행하기 전에 계산 리소스를 사용 가능한 상태가 될 때까지 기다립니다. 이는 계정이 동시 실행 배치 작업 요청의 최대 수를 초과한 경우 발생할 수 있습니다. 이 경우 실행 중인 다른 요청이 완료되거나 취소되면 상태가 진행 중으로 전환됩니다. 또는 Amazon ML이 요청이 필요한 배치 작업이 완료되기를 기다립니다.
진행 중	요청이 여전히 실행 중입니다.
완료됨	요청이 완료되었으며 객체를 사용하거나(ML 모델 및 데이터 소스) 조회할(배치 예측 및 평가) 준비가 되었습니다.
FAILED	제공한 데이터에 문제가 있거나 작업을 취소했습니다. 예를 들어 완료에 실패한 데이터 소스의 데이터 통계를 계산하려고 하면 유효하지 않음 또는 실패 상태 메시지가 표시될 수 있습니다. 오류 메시지가 작업이 성공적으로 완료되지 않은 이유를 설명합니다.
삭제됨	개체가 이미 삭제되었습니다.

Amazon ML은 Amazon ML에서 해당 객체 생성을 완료한 시기와 같은 객체에 대한 정보도 제공합니다. 자세한 내용은 객체 나열 단원을 참조하세요.

시스템 제한

강력하고 안정적인 서비스를 제공하기 위해 Amazon ML은 시스템에 보내는 요청에 특정 제한을 적용합니다. 대부분의 ML 문제는 이러한 제약 조건에 쉽게 들어 맞습니다. 하지만 이러한 제한으로 인해 ML 사용이 제한되는 경우 고객 서비스에 문의하여 한도 상향을 요청할 수 있습니다. 예를 들어, 동시에 실행할 수 있는 작업 수가 5개로 제한될 수 있습니다. 이 제한 때문에 리소스를 기다리는 작업이 대기열에 있는 경우가 많다면 계정에 대한 한도를 높이는 것이 합리적일 수 있습니다.

다음 표에는 Amazon ML의 기본 계정별 한도가 나와 있습니다. 고객 서비스에서 이러한 제한을 모두 높일 수 있는 것은 아닙니다.

제한 유형	시스템 제한
각 관측치의 크기	100KB
학습 데이터 크기 *	100GB
배치 예측 입력 크기	1TB
배치 예측 입력 크기(레코드 수)	1억
데이터 파일에 있는 변수의 수(스키마)	1,000
레시피 복잡성(처리되는 출력 변수의 수)	10,000
각 실시간 예측 엔드포인트의 TPS	200
모든 실시간 예측 엔드포인트의 총 RAM	10,000
모든 실시간 예측 엔드포인트의 총 TPS	10GB
동시 작업 수	25
특정 작업의 최장 실행 시간	7일
멀티클래스 ML 모델의 클래스 수	100
ML 모델 크기	최소 1MB, 최대 2GB
객체당 태그 수	50

- 작업이 시기적절하게 완료될 수 있도록 데이터 파일의 크기가 제한됩니다. 7일 이상 실행 중인 작업은 자동으로 종료되어 실패 상태가 됩니다.

모든 객체의 이름 및 ID

Amazon ML의 모든 객체에는 식별자 또는 ID가 있어야 합니다. Amazon ML 콘솔은 ID 값을 자동으로 생성하지만, API를 사용하는 경우 직접 생성해야 합니다. 각 ID는 계정에서 동일한 유형의 모든 Amazon ML 객체 간에 고유해야 합니다. 즉, 동일한 ID로 평가를 두 번 수행할 수 없습니다. 권장되지는 않지만 동일한 ID를 가진 평가와 데이터 소스가 있을 수 있습니다.

무작위로 생성된 객체 식별자를 사용하고, 객체 유형을 식별하기 위해 접두어에 짧은 문자열을 접두사로 붙이는 것이 좋습니다. 예를 들어 Amazon ML 콘솔이 데이터 소스를 생성할 때 해당 데이터 소스에 "ds-zScWIuWiOxF"와 같은 임의의 고유 ID를 할당합니다. 이 ID는 단일 사용자의 충돌을 피할 수 있을 만큼 충분히 무작위적이며 간결하고 가독성도 뛰어납니다. "ds-" 접두사는 편의성과 명확성을 위한 것이지만 필수는 아닙니다. ID 문자열에 무엇을 사용해야 할지 잘 모르겠으면 모든 최신 프로그래밍 환경에서 쉽게 사용할 수 있는 16진수 UUID 값(예: 28b1e915-57e5-4e6c-a7bd-6fb4e729cb23)을 사용하는 것이 좋습니다.

ID 문자열은 ASCII 문자, 숫자, 하이픈 및 밑줄을 포함할 수 있으며 최대 64자까지 가능합니다. 메타데이터를 ID 문자열로 인코딩하는 것이 가능하고 편리할 수도 있습니다. 하지만 객체를 만든 후에는 해당 ID를 변경할 수 없으므로 사용하지 않는 것이 좋습니다.

객체 이름을 사용하면 사용자에게 친숙한 메타데이터를 각 객체와 쉽게 연결할 수 있습니다. 객체를 만든 후에 이름을 업데이트할 수 있습니다. 이렇게 하면 객체 이름이 ML 워크플로의 일부 측면을 반영할 수 있습니다. 예를 들어 처음에는 ML 모델 이름을 "실험 #3"으로 지정했다가 나중에 모델 이름을 "최종 생산 모델"로 바꿀 수 있습니다. 이름은 최대 1,024자까지 원하는 문자열이 될 수 있습니다.

객체 수명

Amazon ML로 생성한 모든 데이터 소스, ML 모델, 평가 또는 배치 예측 객체는 생성 후 최소 2년 동안 사용할 수 있습니다. Amazon ML은 2년 이상 액세스하지 않았거나 사용하지 않은 객체를 자동으로 제거할 수 있습니다.

리소스

다음의 관련 리소스는 이 서비스 사용 시 도움이 될 수 있습니다.

- ML 제품 정보 — ML에 대한 모든 관련 제품 정보를 중앙 위치에서 캡처합니다.
- ML FAQs – 이 제품에 대해 개발자들이 가장 많이 질문한 내용을 소개합니다.
- ML 샘플 코드 — ML을 사용하는 샘플 애플리케이션. 샘플 코드를 시작점으로 사용하면 자신만의 ML 애플리케이션을 만들 수 있습니다.
- ML API 참조 — ML의 모든 API 작업을 자세히 설명합니다. 또한 지원되는 웹 서비스 프로토콜에 대한 샘플 요청 및 응답도 제공합니다.
- AWS 개발자 리소스 센터 – 혁신적인 애플리케이션을 빌드하는 데 도움이 될 수 있는 문서, 코드 샘플, 출시 정보 및 기타 정보를 찾을 수 있는 출발점을 제공합니다.
- AWS 교육 및 과정 – 역할 기반 및 특수 과정과 자습형 실습과 연결하여 AWS 기술을 연마하고 실용적인 경험을 얻는 데 도움을 드립니다.
- AWS 개발자 도구 – AWS로 혁신적인 애플리케이션을 빌드하는 데 도움이 될 수 있는 문서, 코드 샘플, 출시 정보 및 기타 정보를 제공하는 개발자 도구 및 리소스 링크입니다.
- AWS Support 센터 – AWS 지원 사례를 생성 및 관리하는 허브입니다. 또한 포럼, 기술 FAQ, 서비스 상태 및 AWS Trusted Advisor 등의 기타 유용한 자료에 대한 링크도 포함되어 있습니다.
- AWS Support – 클라우드에서 일대일로 애플리케이션을 빌드 및 실행하도록 지원하는 빠른 응답 지원 채널인 AWS Support에 대한 정보가 포함된 기본 웹 페이지입니다.
- 문의처 – AWS 결제, 계정, 이벤트, 침해 및 기타 문제에 대해 문의할 수 있는 중앙 연락 창구입니다.
- AWS 사이트 약관 – 저작권 및 상표, 사용자 계정, 라이선스 및 사이트 액세스와 기타 주제에 대한 상세한 정보입니다.

문서 기록

다음 표에서는 이번 릴리스의 Amazon Machine Learning(Amazon ML)에 대한 설명서에서 변경된 중요 사항에 대해 설명합니다.

- API 버전: 2015-04-09
- 마지막 설명서 업데이트: 2016년 8월 2일

변경 사항	설명	변경 날짜
지수 추가	이번 Amazon ML 릴리스에는 Amazon ML 객체에 대한 새로운 지수가 추가되었습니다. 자세한 내용은 객체 나열 단원을 참조하세요.	2016년 8월 2일
여러 객체 삭제	이번 Amazon ML 릴리스에는 여러 Amazon ML 객체를 삭제하는 기능이 추가되었습니다. 자세한 내용은 객체 삭제 단원을 참조하세요.	2016년 7월 20일
태깅 추가	이번 Amazon ML 릴리스에는 Amazon ML 객체에 태그를 적용하는 기능이 추가되었습니다. 자세한 내용은 Amazon ML 객체에 태그 지정 단원을 참조하세요.	2016년 6월 23일
Amazon Redshift 데이터 소스 복사	이번 Amazon ML 릴리스에는 Amazon Redshift 데이터 소스 설정을 새 Amazon Redshift 데이터 소스에 복사하는 기능이 추가되었습니다. Redshift 데이터 소스 설정 복사에 대한 자세한 내용은 데이터 소스 복사(콘솔) 단원을 참조하세요.	2016년 4월 11일
셔플링 추가	이번 Amazon ML 릴리스에는 입력 데이터를 셔플링하는 기능이 추가되었습니다. 셔플 유형 파라미터 사용에 대한 자세한 내용은 학습 데이터의 셔플 유형 단원을 참조하세요.	2016년 4월 5일

변경 사항	설명	변경 날짜
Amazon Redshift를 통해 데이터 소스 생성 기능 개선	이번 Amazon ML 릴리스에는 콘솔에서 Amazon ML 데이터 소스를 생성할 때 Amazon Redshift 설정을 테스트하여 연결이 작동하는지 확인할 수 있는 기능이 추가되었습니다. 자세한 내용은 Amazon Redshift 데이터로 데이터 소스 생성(콘솔) 단원을 참조하세요.	2016년 3월 21일
Amazon Redshift 데이터 스키마 변환 기능 개선	이번 아마존 ML 릴리스에서는 Amazon Redshift (Amazon Redshift) 데이터 스키마를 Amazon ML 데이터 스키마로 변환하는 기능이 개선되었습니다. Redshift에서 엔드포인트 사용에 대한 자세한 내용은 Amazon Redshift의 데이터에서 Amazon ML 데이터 소스 생성 단원을 참조하세요.	2016년 2월 9일
CloudTrail 로깅 추가	Amazon ML의 이번 릴리스에서는 AWS CloudTrail (CloudTrail)을 사용하여 요청을 로깅하는 기능이 추가되었습니다. CloudTrail 사용에 관한 자세한 내용은 를 사용하여 Amazon ML API 호출 로깅 AWS CloudTrail 단원을 참조하세요.	2015년 12월 10일
추가 DataRearrangement 옵션 추가	이번 Amazon ML 릴리스에는 입력 데이터를 무작위로 분할하고 보완적인 데이터 소스를 생성하는 기능이 추가되었습니다. `DataRearrangement` 파라미터에 대한 자세한 내용은 데이터 재배열 단원을 참조하세요. 새로운 교차 검증 옵션을 사용하는 방법에 대한 자세한 내용은 교차 검증 단원을 참조하세요.	2015년 12월 3일
실시간 예측 시도	이번 Amazon ML 릴리스에는 서비스 콘솔에서 실시간 예측을 시도할 수 있는 기능이 추가되었습니다. 실시간 예측 시도에 대한 자세한 내용은 머신 러닝 개발자 안내서의 실시간 예측 요청 단원을 참조하세요.	2015년 11월 19일

변경 사항	설명	변경 날짜
새로운 지역	이번 Amazon ML 릴리스에는 EU(아일랜드) 지역에 대한 지원이 추가되었습니다. EU(아일랜드) 지역의 ML에 대한 자세한 내용은 머신 러닝 개발자 안내서의 지역 및 엔드포인트 단원을 참조하세요.	2015년 8월 20일
최초 릴리스	이 설명서는 ML 개발자 안내서의 최초 릴리스입니다.	2015년 4월 9일