Chapter 15

Amazon Machine Learning

WHAT'S IN THIS CHAPTER

- Introduction to the Amazon Machine Learning service
- Learn to create datasources from Amazon S3 buckets
- Learn to create and evaluate ML models
- Learn to create batch predictions
- Learn to create real-time predictions

NOTE Amazon Machine Learning is not included in AWS free-tier accounts. You can find more information on the pricing model of AWS services at `https://aws.amazon.com/pricing/`.

Amazon Machine Learning is a fully managed web service that allows you to create and deploy simple machine learning models without any programming.

Amazon Machine Learning provides a wizard-like interface that allows you to define the location of the input data, define the target attribute, and build an ML model to predict the target attribute. When you are happy with the quality of your ML model, you can deploy it on AWS managed infrastructure and access the model using an API.

Amazon Machine Learning is easy to use, but offers limited feature-engineering and model-building capabilities. Nevertheless, it is a popular choice for building linear regression and logistic regression models. In this chapter you will learn how to build, evaluate, and deploy machine learning solutions on the AWS cloud with Amazon Machine Learning.

NOTE To follow along with this chapter, ensure you have created the S3 buckets listed in Appendix B.

You can download the code files for this chapter from `www.wiley.com/go/machinelearningawscloud` or from GitHub using the following URL:

```
https://github.com/asmtechnology/awsmlbook-chapter15.git
```

Key Concepts

In this section, you learn some of the key concepts you will encounter while working with Amazon Machine Learning.

Machine Learning in the AWS Cloud: Add Intelligence to Applications with Amazon SageMaker and Amazon Rekognition, First Edition. Abhishek Mishra.

Datasources

A *datasource* is an object that stores a reference to an Amazon S3 bucket that stores your input data, along with meta information that describes the characteristics of the input data. When you create a new datasource, Amazon Machine Learning analyzes your input data to create a schema and computes descriptive statistics for the attributes in the schema. The statistics, along with other information such as the test-train split, are also stored as part of the datasource. A datasource is used to train an ML model, evaluate the model, and create batch predictions.

The statistics for each attribute can be viewed as a graph in the Amazon Machine Learning management console and are also used during the model-building process to improve the quality of the resulting ML model.

When dealing with datasources, you are likely to encounter the following terms:

- *Input data:* The input data refers to all the observations referred to by a datasource.
- *Location:* This location generally refers to a file in an Amazon S3 bucket that contains the input data. Besides Amazon S3, you can store data in Amazon RedShift databases, or MySQL databases within Amazon RDS.
- *Schema:* The schema refers to the structure of the input data, typically a list of attribute names along with their data types.
- *Observation:* An observation refers to a single unit of input data. In the case of tabular input data, an observation would correspond to an entire row in the table.
- *Attribute:* An attribute is a unique property of the input data, shared across all observations. In the case of tabular input data, an attribute would correspond to a column in the table. The data type of an attribute can be Binary, Numeric, Categorical, or Text.
- *Target attribute:* When training an ML model, the target attribute identifies the name of an attribute in the input data that contains the correct answers. When evaluating and predicting using an ML model, the target attribute represents the name of the attribute whose value you want to predict.

Amazon Machine Learning requires you to use separate datasource objects for model building and batch predictions. The datasource that is used while making batch predictions does not have a target attribute.

ML Model

A machine learning model is a mathematical model that is capable of learning patterns in data and making predictions based on those patterns. The machine learning models generated by Amazon Machine Learning are based on either linear or logistic regression and can be used for the following tasks:

- *Regression:* Predicting a continuous numeric value.
- *Binary classification:* Predicting values that have only one of two possible states.
- *Multiclass classification:* Predicting values that belong to a fixed, predefined set of possible values.

The process of building a model with Amazon Machine Learning usually involves the following steps:

1. Upload your input data to Amazon S3 and create a datasource.
2. Choose the test-train spilt. The Amazon Machine Learning default behavior is to reserve 30% of the data for testing and the remaining 70% for training.
3. Shuffle the data. Amazon Machine Learning automatically shuffles data for you.
4. Select the features and target variables; optionally apply some feature processing.
5. Select model-training parameters.
6. Create the ML model.

Building an ML model that matches your needs usually involves iterating through this ML process and evaluating a few variations in the input features, feature processing, and training parameters.

Once you have arrived at a satisfactory ML model, you will use it to make predictions. It is a good idea to store a copy of the incoming data and periodically evaluate the performance of the model on this new data. ML models will only predict accurately if the data that the model was trained on has a similar distribution to the data on which it is making predictions.

If you detect the performance of the model has degraded on the new test dataset, you would need to create a new model based on a training set that includes some of the new observations.

Regularization

Regularization is a technique that prevents models from overfitting training data by penalizing extreme weights. Overfitting occurs when your model is able to perform well on the training data but performs poorly on data it has not encountered in the past—in effect, the model has memorized the training data instead of generalizing from the data.

Amazon Machine Learning allows you to choose from two types of regularization while building models:

- *L1 regularization:* This form or regularization will push small weights to zero. In a linear model, a feature with zero weight does not contribute to the prediction—therefore, in effect, this form of regularization is reducing the number of features being used by the model.
- *L2 regularization:* This form of regularization penalizes large weights and results in smaller overall weights. It is the default type of regularization selected by Amazon Machine Learning when you build an ML model.

Training Parameters

A training parameter (also known as a hyperparameter) is a setting that controls the manner in which the ML model is built and consequently the effectiveness of the model. Amazon Machine Learning allows you to control the following training parameters:

- *Maximum model size:* This is the total size in bytes of the patterns generated by Amazon Machine Learning during the training of the model. The default value of this parameter is

100 MB. Choosing a model size lets you choose a trade-off between predictive accuracy and the cost you will pay to use the model to make predictions. Using a smaller model size could result in Amazon Machine Learning discarding some patterns to fit within the size limit. Larger models, on the other hand, cost more to query when making real-time predictions.

- *Maximum number of training passes:* This parameter controls the number of passes over the input data that Amazon Machine Learning can make to discover patterns in your data. Increasing the number of passes will result in an increase in training time and the cost of training the model. The default value of this parameter is 10, but you can increase it up to 100. If your training dataset is small, it is likely to contain fewer samples that are similar to each other, and therefore you will need more passes to obtain higher model quality.
- *Shuffle type:* This parameter allows you to specify whether you want Amazon Machine Learning to shuffle the input data before it is split into the training and evaluation datasources. The default option used by Amazon Machine Learning when you create a model is to use a pseudorandom shuffling algorithm. If you have already shuffled the data prior to creating the input datasource, you can set the shuffle type to None. It is worth noting that the shuffling is performed at the point Amazon Machine Learning splits the input datasource into the training and evaluation datasources. Subsequent passes of the model-building process do not shuffle the data.
- *Regularization type:* Regularization is a technique that prevents ML models from overfitting by penalizing large weight values. You have two forms of regularization to choose from when you build an ML model. L1 regularization pushes small weight values toward zero, in effect canceling out the effect of the associated input attribute. L2 regularization prevents very large weight values, and is the default used by Amazon Machine Learning. You can also choose to apply no regularization by setting the regularization type to None during the model-building process.
- *Regularization amount:* This parameter lets you control how much regularization to apply during the model-building process. Amazon Machine Learning provides three options: Mild, Moderate, and High.

Descriptive Statistics

When you create a datasource, Amazon Machine Learning computes statistical information on your data that you can use to understand your data. These statistics can be accessed from the Amazon Machine Learning console and are computed on each attribute. For numeric attributes, Amazon Machine Learning computes the following statistics:

- Minimum, maximum, median, and mean values
- Histogram
- Number of missing/invalid values

For binary and categorical attributes, Amazon Machine Learning computes the following statistics:

- Count of distinct values per category
- Histogram

- Percentage of true values (Binary data only)
- Most common values

For text attributes, Amazon Machine Learning computes the following statistics:

- Total number of words
- Number of unique words
- Range of number of words per observation
- Range of word lengths
- Most prominent words

Pricing and Availability

Amazon Machine Learning is available on a pay-per-use model and is not included in the AWS free tier. You will be charged a flat hourly fee based on the amount of compute resources consumed to create data sources, models, and evaluations. You will also be charged for making predictions with the ML models you create. Charges for services like Amazon S3 used for building datasources are billed separately. You can get more details on the pricing model at `https://aws.amazon.com/sagemaker/`.

The ability to create datasources, ML models, and evaluations and to make batch predictions is available in all AWS regions. However, the ability to make real-time predictions with the Amazon Machine Learning API is only available in the US East (N. Virginia) and EU (Ireland) regions. You can find more information on service availability at `https://docs.aws.amazon.com/machine-learning/latest/dg/regions-and-endpoints.html`.

Creating Datasources

In this section you will upload the Titanic dataset to Amazon S3 and use the Amazon Machine Learning management console to create two sources. These datasources will be used in subsequent sections of this chapter when we build an ML model to predict which passengers were more likely to survive the Titanic disaster. The first datasource will be used for model building, and the second will be used for creating batch predictions.

The Titanic dataset is a very popular dataset that contains information on the demographic and ticket information of 1309 passengers on board the Titanic, with the goal being to predict which of the passengers were more likely to survive. The full dataset is available from the Department of BioStatistics at Vanderbilt University (`http://biostat.mc.vanderbilt.edu/wiki/Main/DataSets`). Over time, researchers at Vanderbilt University have produced various versions of the dataset, with the most recent version (titanic3) being created by Thomas Cason. Thomas Cason's version is sorted by passenger name, and you can access notes on the titanic3 dataset at `http://biostat.mc.vanderbilt.edu/wiki/pub/Main/DataSets/titanic3info.txt`.

Versions of the titanic3 dataset are also available from several other sources, including a popular Kaggle competition titled Titanic: Machine Learning From Disaster (`https://www.kaggle.com/c/titanic`). The Kaggle version is included with the resources that accompany this chapter, and has the benefit of being shuffled and pre-split into a training and validation set.

The dataset consists of two files: `train.csv` and `test.csv`. Table 15.1 lists the first five rows of the `train.csv` file.

Let's briefly examine the attributes of the dataset:

- *PassengerId:* A text variable that acts as a row identifier.
- *Survived:* A Boolean variable that indicates if the person survived the disaster. 0 = No, 1 = Yes.
- *Pclass:* A categorical variable that indicates the ticket class. 1 = 1st class, 2 = 2nd class, 3 = 3rd class.
- *Name:* The name of the passenger.
- *Sex:* A categorical variable that indicates the sex of the passenger.
- *Age:* A numeric variable that indicates the age of the passenger.
- *SibSp:* A numeric variable that indicates the number of siblings/spouses traveling together.
- *Parch:* A numeric variable that indicates the number of parents and children traveling together.
- *Ticket:* A text variable containing the ticket number.
- *Fare:* A numeric variable that indicates the fare paid in pre-1970 British pounds.
- *Cabin:* A textual variable that indicates the cabin number.
- *Embarked:* A categorical variable that indicates the port of embarkation. C = Cherbourg, Q = Queenstown, S = Southampton.

When you create a datasource in Amazon Machine Learning, you have the option to split the data into two sets. If you choose this option, Amazon Machine Learning will create two new datasources out of the original during the model-building process. We will use this feature and have Amazon Machine Learning use 70% of the `train.csv` file for model building and 30% for model evaluation. We will then create another datasource for batch predictions using the `test.csv` file and not have Amazon Machine Learning split this datasource.

You will create two datasources in this section:

- `Titanic_TrainingDataSource`
- `Titanic_TestDataSource`

Amazon Machine Learning will split the `Titanic_TrainingDataSource` datasource into two when you build an ML model in the next section:

- `Titanic_TrainingDataSource_[percentBegin=0, percentEnd=70, strategy=sequential]`
- `Titanic_TrainingDataSource_[percentBegin=70, percentEnd=100, strategy=sequential]`

Table 15.1: The First Five Rows of the Titanic Dataset

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss Laina	female	26	0	0	STON/ O2. 3101282	7.925		S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S

Effectively, you will end up with three datasources at the end of the model-building process and use the `Titanic_TestDataSource` datasource in a subsequent section of this chapter to make batch predictions.

Before you can create the two Amazon Machine Learning datasources, you need to upload the `test.csv` and `train.csv` files into an Amazon S3 bucket. The bucket name used in this section is `awsml-amazonml-source`. Since bucket names are unique, you will need to substitute references to this bucket with your own bucket name.

Creating the Training Datasource

Log in to the AWS management console using the dedicated sign-in link for your development IAM user account. Use the region selector to select a region where the Amazon Machine Learning service is available. The screenshots in this section assume that the console is connected to the EU (Ireland) region. Click the Services menu and access the Amazon S3 service home page.

Click the `awsml-amazonml-source` bucket in the S3 management console and upload the `test.csv` and `train.csv` files to the bucket, accepting the default options in the Amazon S3 file upload dialog (Figure 15.1).

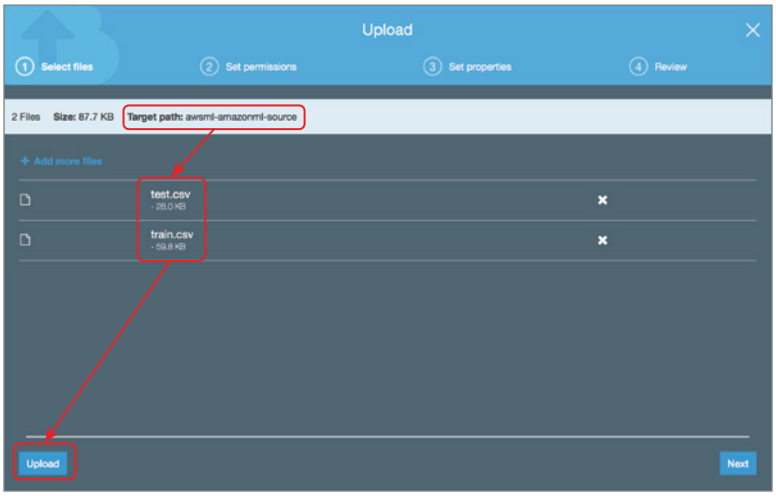


FIGURE 15.1
Uploading the Titanic dataset to an Amazon S3 bucket

After the files have been uploaded to the bucket, click the Services menu and access the Amazon Machine Learning service home page (Figure 15.2).

Click the Get Started button on the home page to proceed (Figure 15.3).

Amazon Machine Learning provides a convenient wizard-like interface for individual tasks such as creating datasources, models, evaluations, and batch predictions. To access these wizards, click the View Dashboard button (Figure 15.4).

The Amazon Machine Learning dashboard lets you view all your datasources, models, and evaluations from one screen. Click the Create New button and select Datasource from the drop-down menu (Figure 15.5).

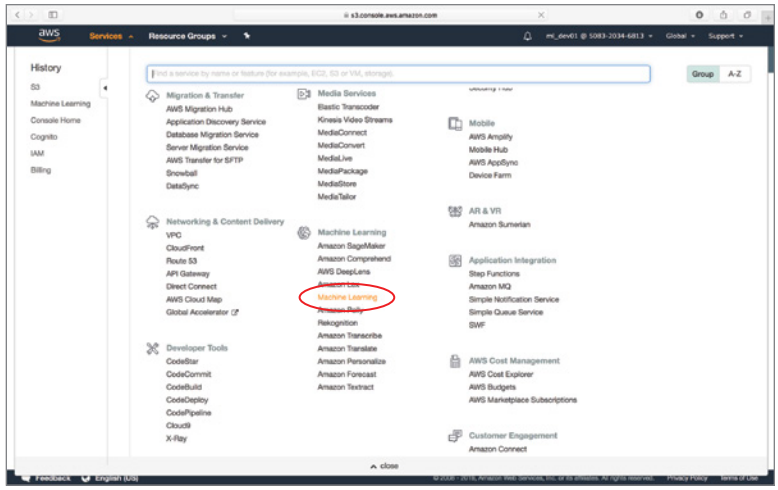


FIGURE 15.2
Accessing the Amazon Machine Learning service home page

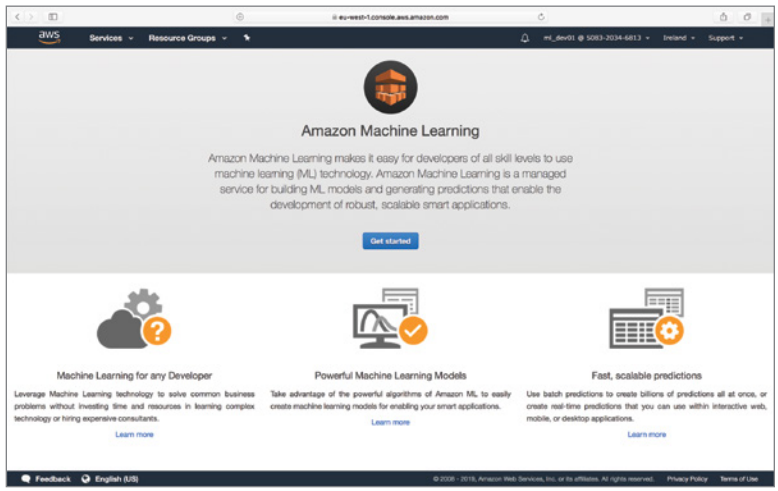


FIGURE 15.3
The Amazon Machine Learning service home page

Ensure you select Amazon S3 as the source and type the name of the bucket followed by the `train.csv` file in the S3 Location field. For example, if your bucket is called `awsml-amazonml-source`, type **`awsml-amazonml-source/train.csv`** in the S3 Location field. Name the datasource `Titanic_TrainingDataSource` and click the Verify button to proceed to the next step (Figure 15.6).

During the verification process, Amazon Machine Learning will prompt you to allow access to the Amazon S3 bucket. Click the Yes button when prompted (Figure 15.7).

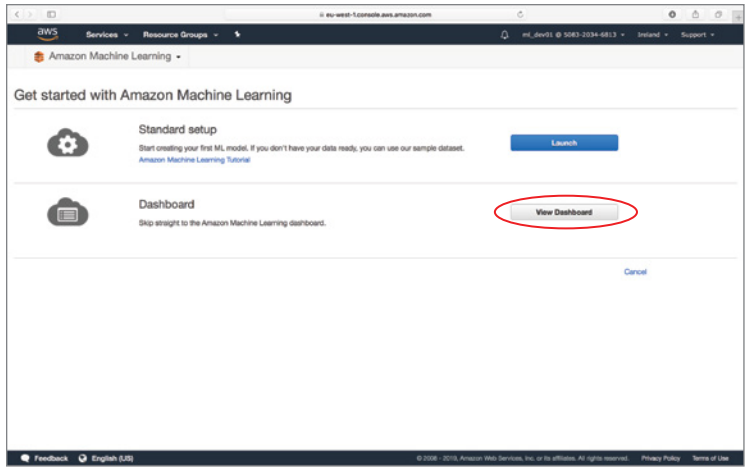


FIGURE 15.4
Accessing the Amazon Machine Learning dashboard

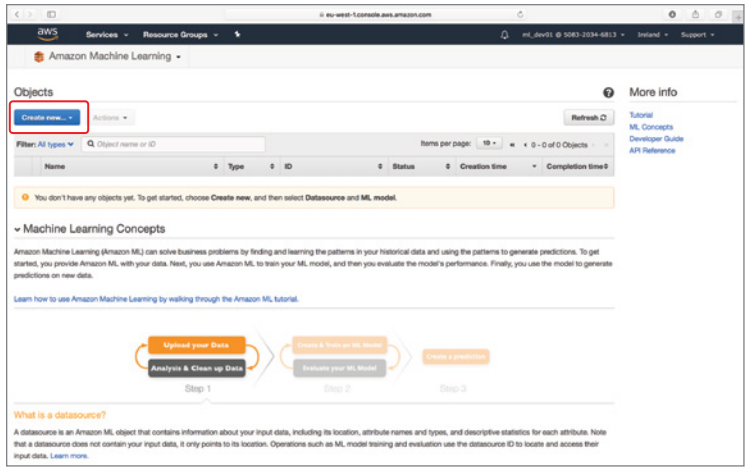


FIGURE 15.5
Accessing the Create Datasource option from the Amazon Machine Learning dashboard

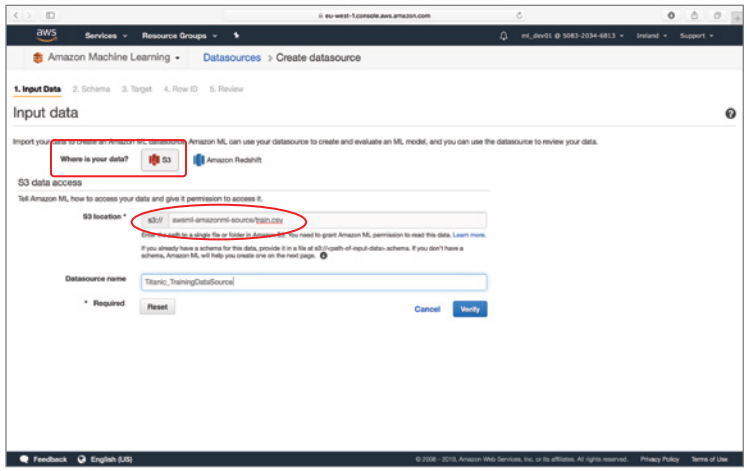


FIGURE 15.6
Specifying the location of the input file

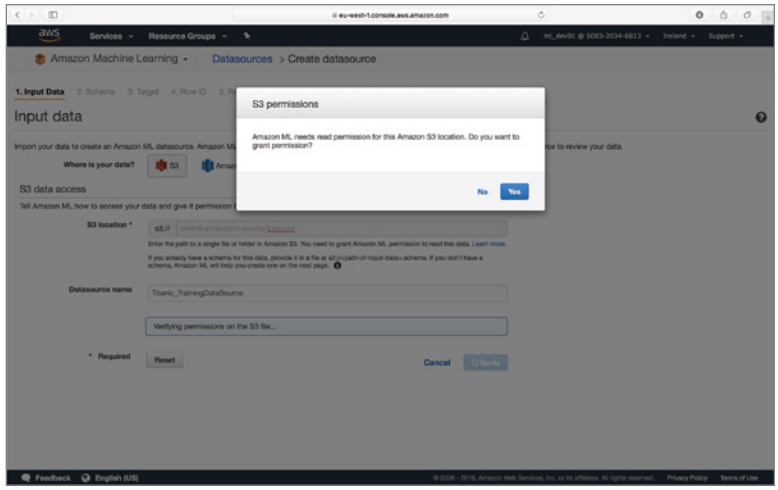


FIGURE 15.7
Granting Amazon Machine Learning access to your Amazon S3 bucket

Once the verification is successful, click the Continue button to move to the Schema section of the wizard. Amazon Machine Learning will examine your data and attempt to define the schema for your data. In most cases, you will need to tweak the default schema created by Amazon Machine Learning. Since the first row of the `train.csv` file contains column names, ensure the Does The First Line In Your CSV Contain The Column Names? option is set to Yes (Figure 15.8).

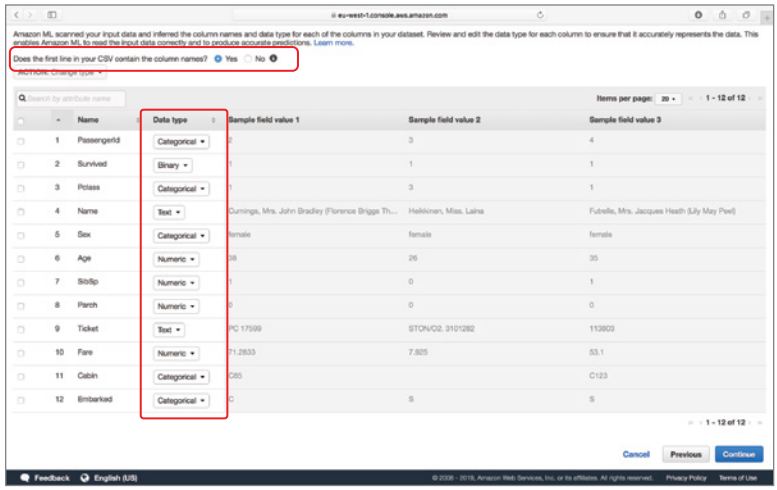


FIGURE 15.8
Modifying the default schema generated by Amazon Machine Learning

You will also need to change some of the data types inferred by Amazon Machine Learning while defining the schema. Ensure the data types for the column names match the following information and click Next:

- *PassengerId:* Categorical
- *Survived:* Binary
- *Pclass:* Categorical

- *Name:* Text
- *Sex:* Categorical
- *Age:* Numeric
- *SibSp:* Numeric
- *Parch:* Numeric
- *Ticket:* Text
- *Fare:* Numeric
- *Cabin:* Categorical
- *Embarked:* Categorical

On the next screen, you will be asked if you intend to use this datasource for training or evaluation. A datasource that is used for model building or evaluation must have known values of the target attribute. Ensure you select Yes in the Do You Plan To Use This Dataset To Create Or Evaluate An ML Model? option (Figure 15.9).

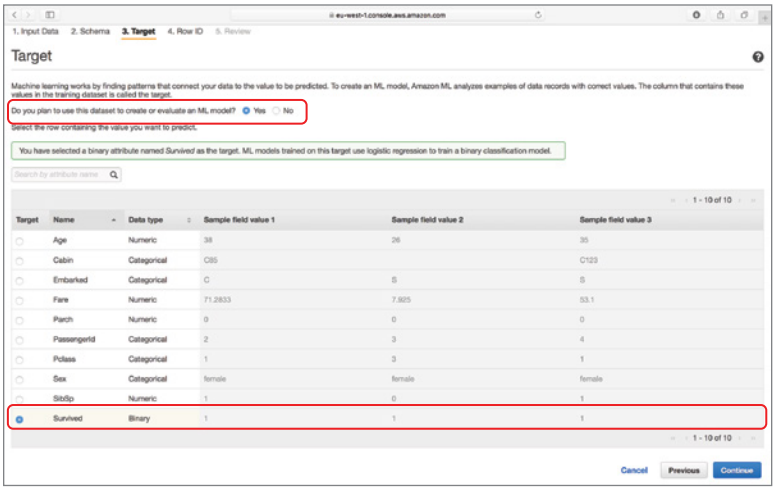


FIGURE 15.9
Specifying the target attribute

Select the `Survived` attribute as the target. Amazon Machine Learning will choose the ML model type based on the data type of the target attribute, according to the following rules:

- *Numeric:* If the target attribute is numeric, Amazon Machine Learning will generate a linear regression model.
- *Binary:* If the target attribute is binary, Amazon Machine Learning will generate a logistic regression model.
- *Categorical:* If the target attribute is categorical, Amazon Machine Learning will generate a multinomial regression model.

Amazon Machine Learning does not allow you to select a text attribute to be the target. Text attributes, therefore, are not listed in the target selection list. Click Continue to proceed.

The next step of the wizard is optional, and allows you to select an attribute that is to be used as a row identifier. Not all datasets have such an attribute. If you are using the dataset included with this lesson, answer Yes to the question Does Your Data Contain An Identifier?, and select the `PassengerId` attribute to act as the row identifier (Figure 15.10).

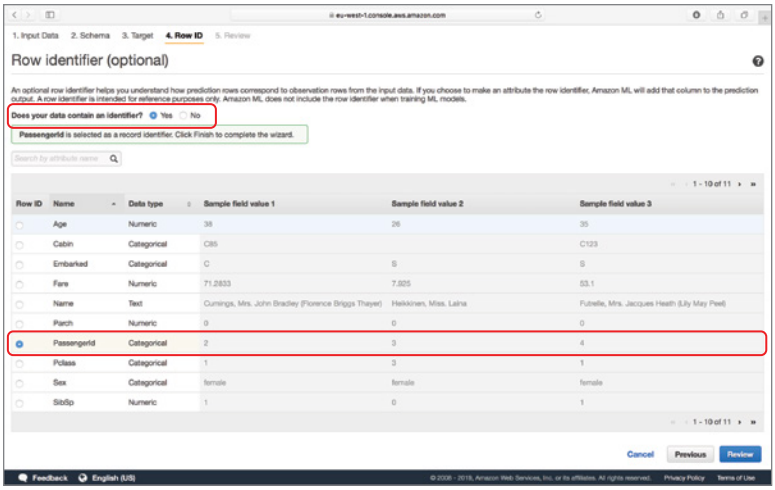


FIGURE 15.10
Specifying a row identifier attribute

A row identifier attribute is not used during the ML model-building or evaluation process. The value of the row identifier will be included with the prediction output. The row identifier attribute must be categorical. Click Review to proceed.

The final screen of the wizard allows you to review the settings for your new datasource (Figure 15.11). Review the settings on the screen and click the Create button at the bottom of the page to create the datasource.

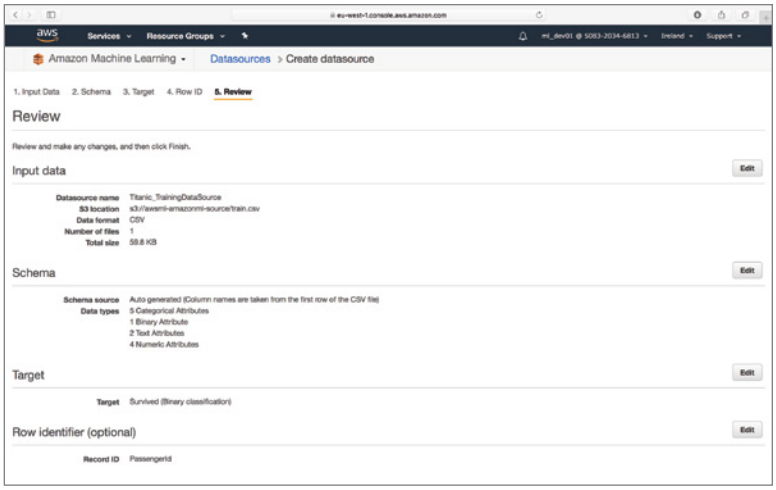


FIGURE 15.11
Datasource Review screen

Creating the datasource can take several minutes. While the datasource is being created, its status will show as Pending in the Amazon Machine Learning dashboard. You can access a list of datasources using the Amazon Machine Learning dashboard, and the dashboard allows you to filter the items that are displayed using a drop-down menu (Figure 15.12).

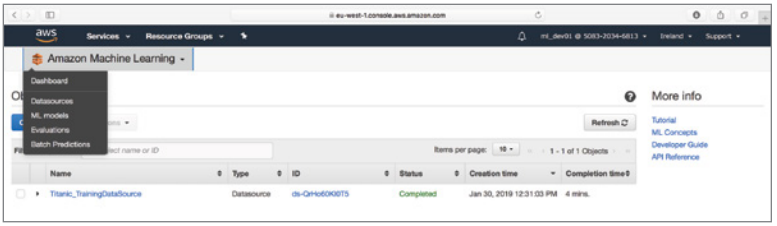


FIGURE 15.12
Filtering the items displayed in the Amazon Machine Learning dashboard

Creating the Test Datasource

Navigate to the Amazon Machine Learning dashboard, and create a new datasource. Ensure you select Amazon S3 as the source and type the name of the bucket followed by the `test.csv` file in the S3 Location field. For example, if your bucket is called `awsml-amazonml-source`, type **`awsml-amazonml-source/test.csv`** in the S3 Location field (Figure 15.13).

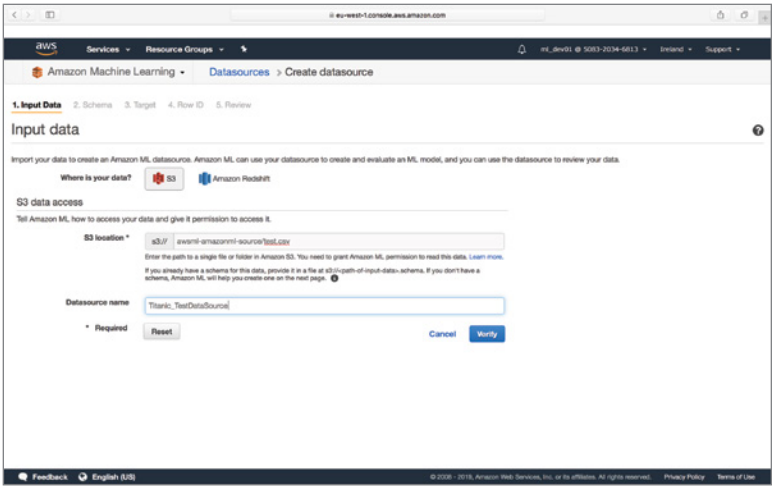


FIGURE 15.13
Specifying the location of the data for the new datasource

Name the datasource `Titanic_TestDataSource` and proceed to the Schema section of the process.

Since the first row of the `test.csv` file also contains column names, ensure the Does The First Line In Your CSV Contain The Column Names? option is set to Yes on the schema definition screen (Figure 15.14).

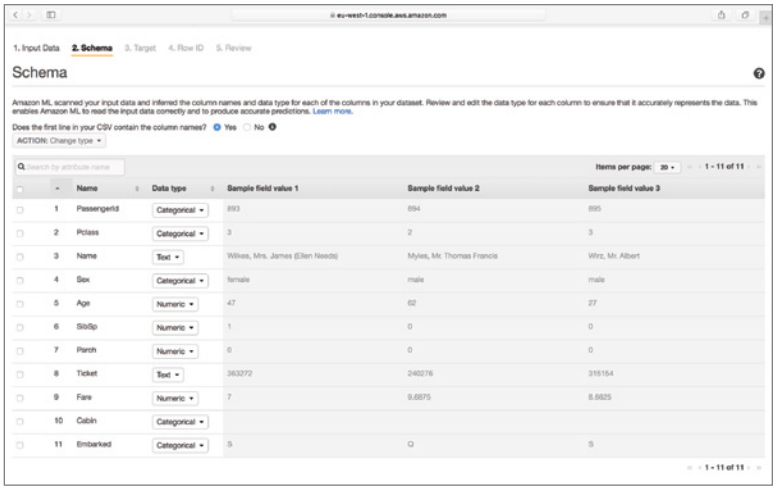


FIGURE 15.14
Setting up the schema for the new datasource

Ensure the data types for the column names match the following information:

- *PassengerId:* Categorical
- *Pclass:* Categorical
- *Name:* Text
- *Sex:* Categorical
- *Age:* Numeric
- *SibSp:* Numeric
- *Parch:* Numeric
- *Ticket:* Text
- *Fare:* Numeric
- *Cabin:* Categorical
- *Embarked:* Categorical

On the next screen, you will be asked if you intend to use this datasource for training or evaluation. Ensure you answer No to this question, as this datasource will be used for batch predictions, and not model building (Figure 15.15).

Select the `Survived` attribute as the target. Amazon Machine Learning will choose the ML model type based on the data type of the target attribute, according to the following rules:

- *For binary classification:* Logistic Regression
- *For multi-class classification:* Multinomial Logistic Regression
- *For regression:* Linear Regression

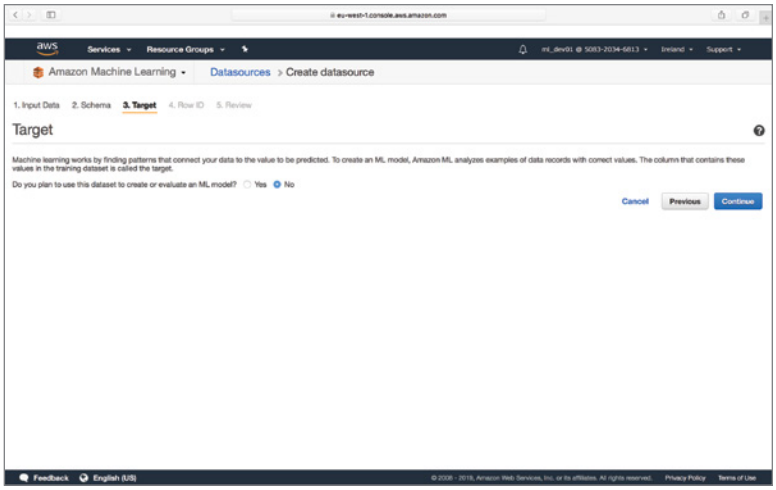


FIGURE 15.15
The new datasource does not have a target attribute.

When asked if your data contains a row identifier, answer Yes and select the `PassengerId` attribute to act as the row identifier (Figure 15.16).

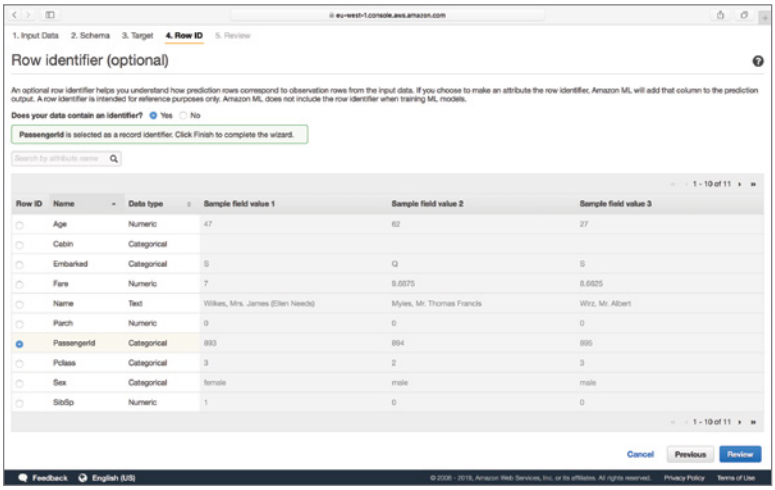


FIGURE 15.16
Specifying a row identifier attribute

Proceed to the Review screen, and click the Create button at the bottom of the page to create the datasource. After a few minutes you will see your new datasource listed in the Amazon Machine Learning dashboard.

Viewing Data Insights

After Amazon Machine Learning has finished creating the datasource, you can access datasource statistics by clicking the name of the datasource in the dashboard (Figure 15.17).

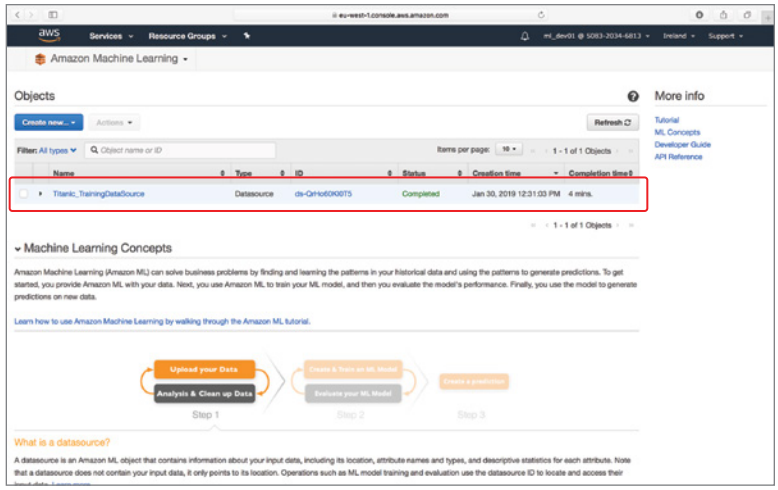


FIGURE 15.17
Selecting the datasource from the dashboard

You will be taken to the datasource summary page. Click the Target Distributions link in the menu on the left side of the page to view a histogram that depicts the distribution of the target variable—in this case, the number who survived and the number who did not (Figure 15.18).

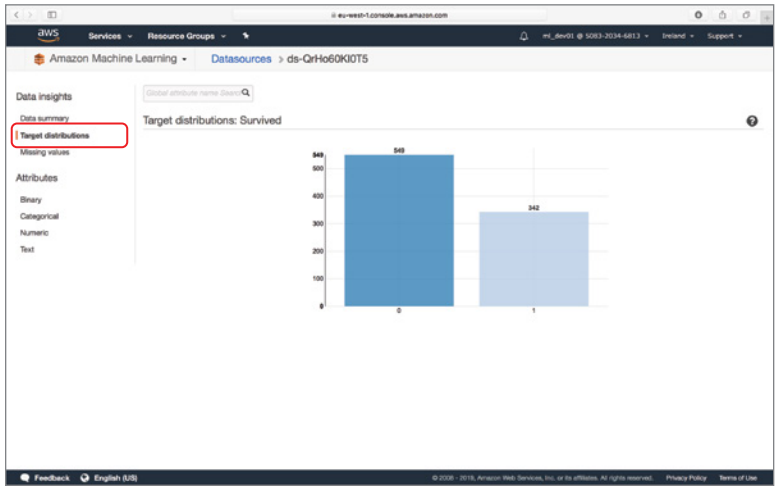


FIGURE 15.18
Histogram of the target attribute

It is quite clear from the histogram that our dataset contains data for more people who died than survived. This suggests that there is a bias in our data, but in this case we know from historical records that this bias is not artificially introduced by our sampling techniques, and the simple fact is that more people died on the Titanic than survived. You can access summary statistics for each attribute in the datasource. Click the Categorical link in the menu on the left side of the page to view summary statistics for all categorical values (Figure 15.19).

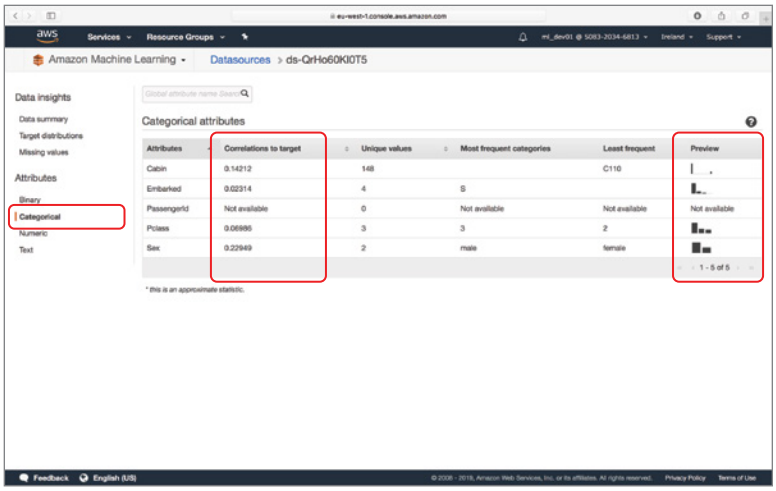


FIGURE 15.19
Summary statistics for categorical values

Looking at the correlation column of Figure 15.19, it is quite clear that the `Sex` attribute has the strongest correlation to the target attribute, and the `Embarked` attribute has the weakest. If you click the little histogram icon in the Preview column, you can see a detailed distribution of the values. Figure 15.20 depicts the distribution of values of the `Embarked` attribute.

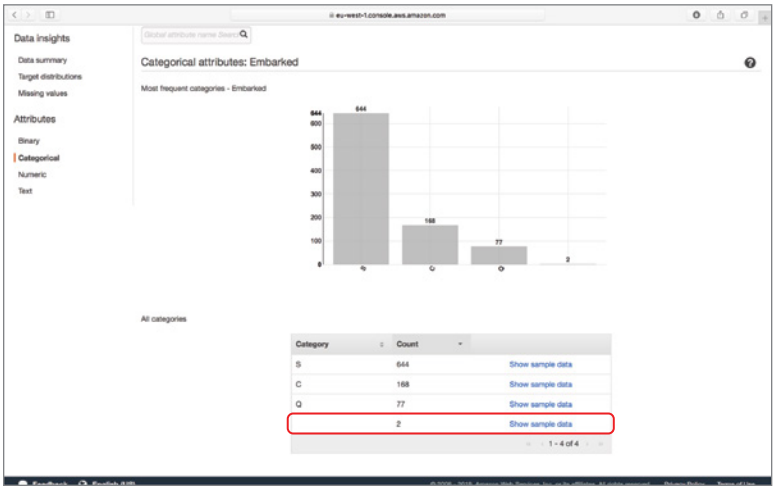
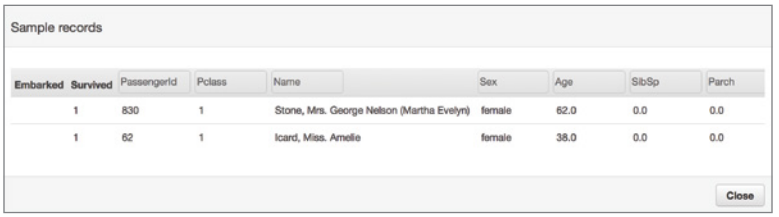


FIGURE 15.20
Distribution of values of the Embarked attribute

The distribution indicates that a vast majority of the people in our training dataset have embarked from Southampton, and there are only two rows of data where the port of embarkation was unspecified. Clicking the Show Sample Data link will present the two rows that do not have values for the `Embarked` attribute (Figure 15.21).

FIGURE 15.21
Rows that do not have a value for the Embarked attribute



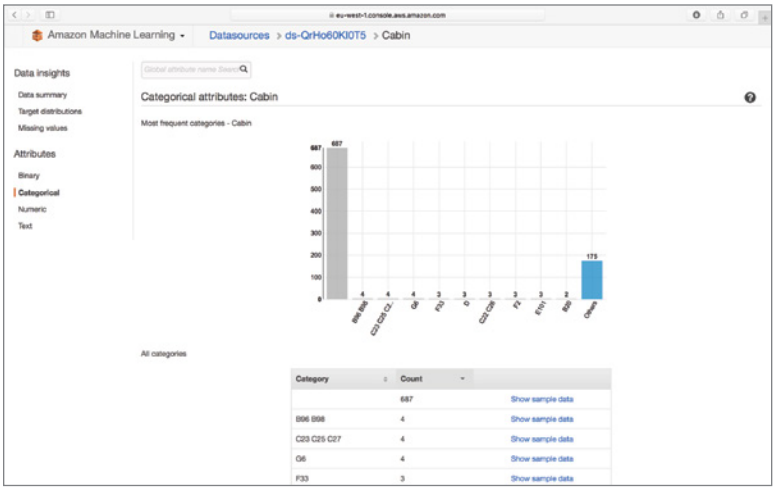
You may be wondering why Amazon Machine Learning did not report these values as missing. The reason is that these are categorical attributes, and Amazon Machine Learning has assumed that missing data is just another category.

We could deal with the missing values in two ways: either guess a value, or delete the rows from the dataset. It is not possible to guess if making these changes will result in a better model at this stage. Building a good model is an iterative process; it starts with building an initial model and then, if needed, making changes to the features or model parameters to create different versions of the model. It is possible that the first model you create is good enough for your purposes. It all depends on what you want to achieve and how much time you have.

Amazon Machine Learning provides you options to control the model-building process using model parameters. You can also create different versions of datasources with better features and build models on these. In this chapter we will accept the default model parameters suggested by Amazon Machine Learning and use the ML model. Although the `Embarkation` feature contains missing values, the proportion of values that are missing is low, and the correlation between this attribute and the target variable is also low. Therefore, the impact of leaving these values as they are may not be significant.

Another interesting categorical attribute is `Cabin`. At face value, this attribute is also very poorly correlated with the target. However, looking at the distribution of values for this attribute clearly indicates that the vast majority of passengers in our dataset do not have cabin information (Figure 15.22).

FIGURE 15.22
Distribution of the Cabin attribute



Perhaps it may be sensible to create a Boolean attribute out of this data that has 0 for all rows that do not have cabin information and 1 for the rows that do. Amazon Machine Learning has very limited feature-engineering capabilities and cannot be used for this type of feature engineering. If you want to take this on as an exercise, you can create a new CSV file using Python or Microsoft Excel and build a new datasource.

NOTE If you use Microsoft Excel on a Mac for feature engineering, ensure you save the CSV file as a Windows CSV file. There are subtle differences in how newlines are represented on Mac and Windows. Amazon Machine Learning can only build datasources out of CSV files that are saved using Microsoft Windows newline characters.

Click the Numeric link on the left-hand side of the page to access summary statistics for numeric attributes (Figure 15.23).

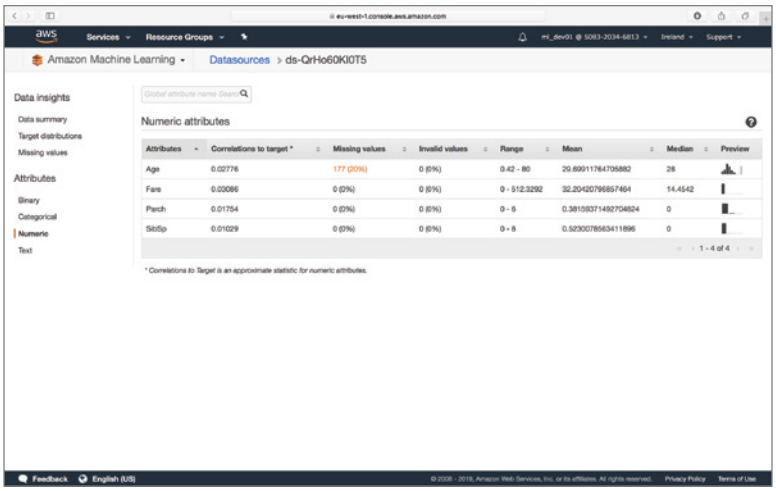


FIGURE 15.23
Summary statistics for numeric attributes

It is quite clear from the statistics that there are a significant number of missing values for the `Age` attribute. The histogram of age values indicates that they are evenly distributed around 30, which also ties in nicely with the fact that the mean and median are also close to 30 (Figure 15.24).

You could, in this case, use the median value for all the rows that are missing a value for the `Age` attribute, and create an additional binary value that captures the fact that the age is not known. This would again require creating a new CSV file with the changes and a new datasource. Once again, it is not possible to guess if these changes will make a better model than the one that Amazon Machine Learning will generate with default settings.

Amazon Machine Learning's default behavior when it encounters a numeric attribute with missing values is to create a new binary variable to capture the missing attribute values and use this in the model-building process. To learn more about how Amazon Machine Learning handles missing and invalid values, visit `https://docs.aws.amazon.com/machine-learning/latest/dg/data-insights.html`.

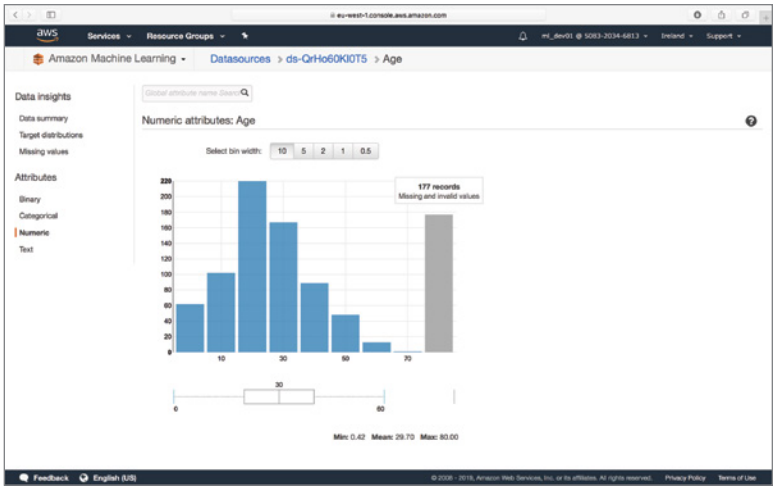


FIGURE 15.24
Distribution of values for the Age attribute

Some other statistics that stand out include that the distribution for the `Fare` attribute is extremely skewed. The range of values is between 0 and 512.3292, with the median value being only 14.4542. Also, the `Parch` and `Sibsp` attributes seem to take on a small number of values, and may perhaps have been better treated as categorical data instead of numeric. Another option could be to combine the `Parch` and `Sibsp` attributes into a new categorical attribute called `FamilySize`.

Now that we have inspected the insights provided by Amazon Machine Learning on our data, we will use this datasource as is to create an ML model.

Creating an ML Model

To create an ML model using the datasource we created earlier in this chapter, navigate to the Amazon Machine Learning dashboard and select the Create New ML Model option (Figure 15.25).

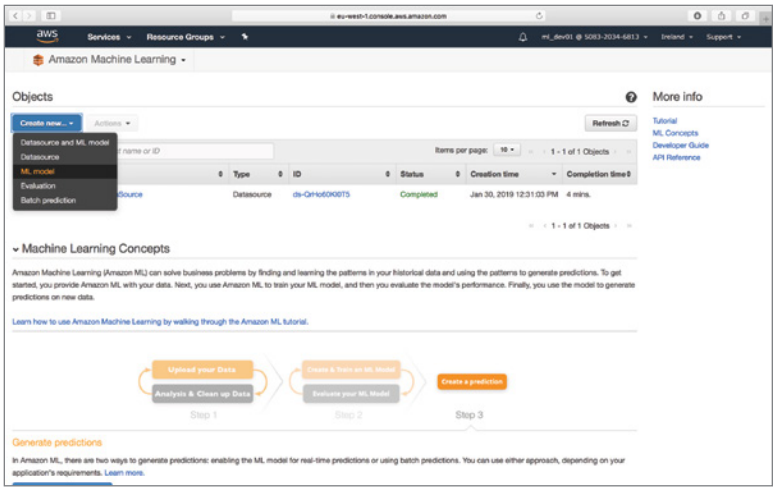


FIGURE 15.25
Creating an ML model

You will be asked to select a datasource. Locate the datasource that corresponds to the `train.csv` file and select it from the list of available datasources (Figure 15.26).

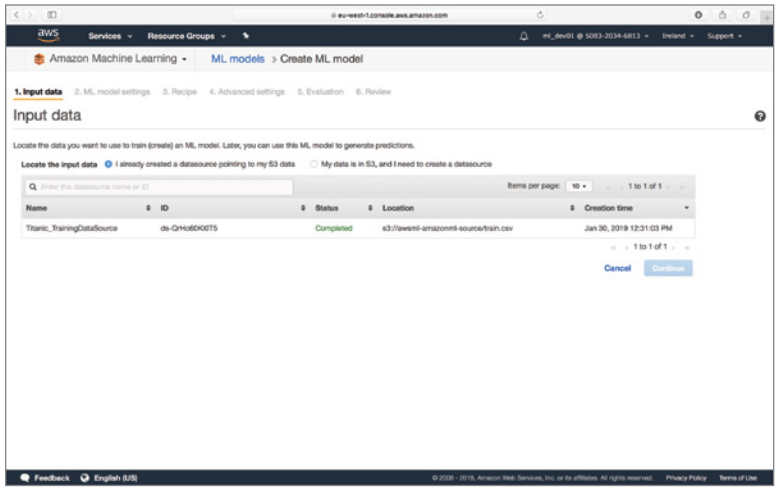


FIGURE 15.26
Selecting a datasource

A summary of the datasource will be presented. Click Continue to proceed to the next step. On the ML Model Settings page, provide a name for the model and select the Default training and evaluation option (Figure 15.27).

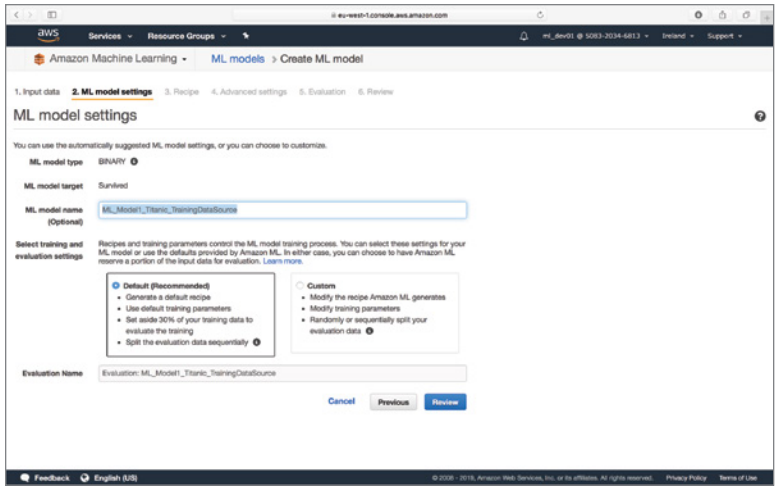


FIGURE 15.27
Specifying ML model settings

When you choose the default option, Amazon Machine Learning will set aside 30% of the data in the datasource for evaluation, and the remaining 70% for training. Amazon Machine Learning will also use default values for a number of model parameters.

Accepting the default settings is okay for creating a baseline model. You can then create additional models with different custom settings and compare the performance of these models against the baseline.

Click the Review button to proceed to the Review page. Scroll down to the bottom of the page and click the Create Model button to create the model.

It can take a few minutes to create a model. Before creating the new model, Amazon Machine Learning will create two new datasources from your datasource, with one datasource containing the 70% training set, and the other containing the 30% evaluation set.

After creating the model, Amazon Machine Learning will create an evaluation using the evaluation set. You can access the model, new data sources, and evaluation from the dashboard (Figure 15.28).

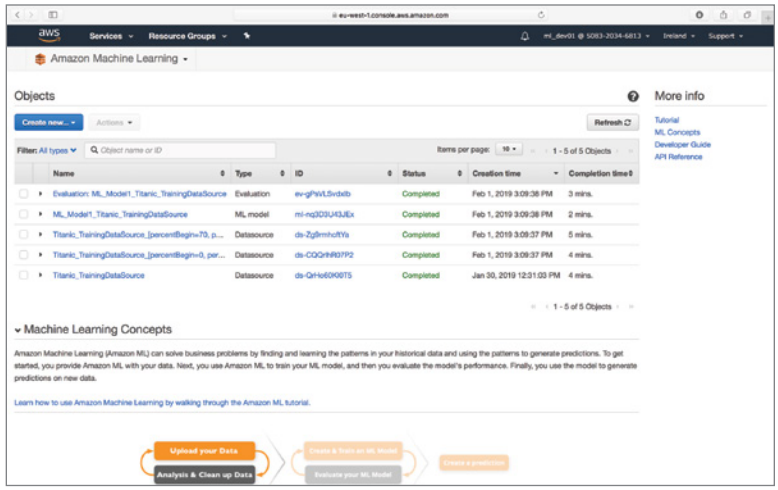


FIGURE 15.28
Amazon Machine Learning dashboard showing new data sources, the ML model, and the evaluation

Let's now examine the characteristics of the ML model created by Amazon Machine Learning using the default settings. Click the model in the dashboard to access the Summary page for the model (Figure 15.29).

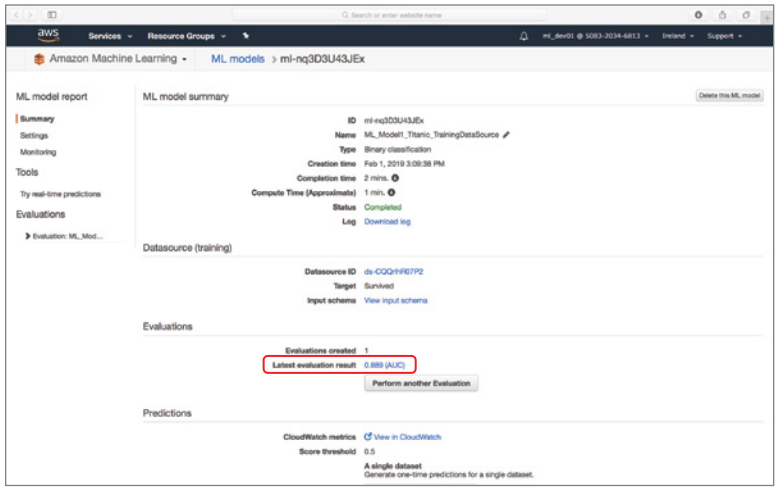


FIGURE 15.29
ML model summary

The results of the latest evaluation are presented in the Summary page. The AUC metric is listed as 0.889. Click the AUC metric for more details (Figure 15.30).

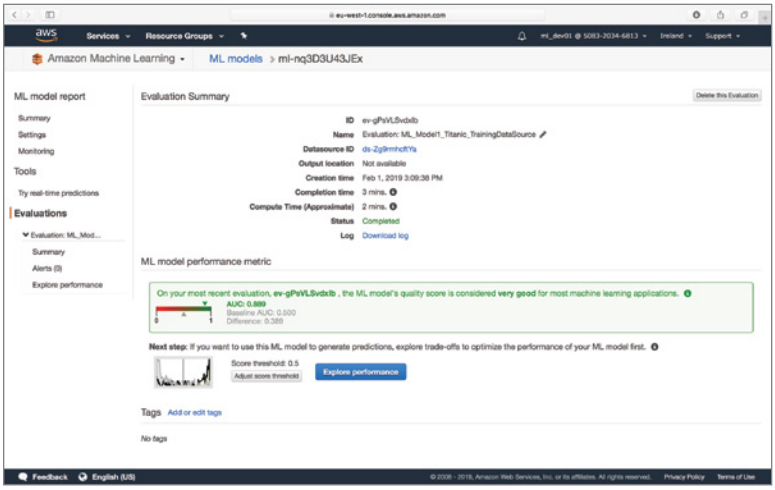


FIGURE 15.30
ML model evaluation

An AUC of 0.889 is quite good, considering an AUC score of 0.5 implies the model is randomly guessing. Click the Explore Performance button to access additional model statistics.

To tune the performance of the model, you can change the score threshold for binary classification by dragging the slider horizontally on the graph (Figure 15.31).

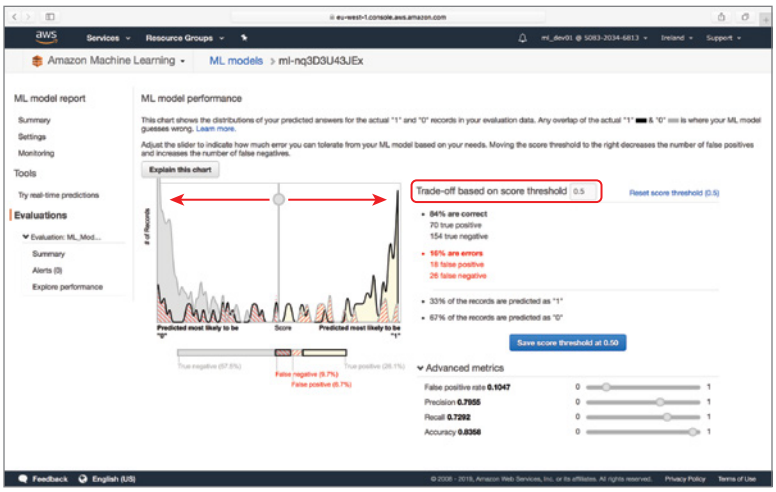


FIGURE 15.31
Advanced ML model statistics

Changing the score threshold will have an impact on the accuracy, precision, and recall of the model, but not the AUC score. At the moment, with the default score of 0.5, the key performance indicators of the model are:

- *AUC:* 0.889
- *Precision:* 0.7955

- *Recall:* 0.7292
- *Accuracy:* 0.8358

False positive rate, precision, recall, and accuracy also have their own sliders, and you can use them to change one of these variables and watch the score threshold adjust to compensate.

What is the optimum score threshold to use? This is a subject of active machine learning research. It would depend on what you intend to use the model for. A model that predicts the incidence of a disease should have a very low false positive rate. In this example, we will go for a higher accuracy, and by moving the accuracy slider to the right, we can see that a score of 0.37 results in an accuracy of 0.8507, with a slightly lower precision of 0.7917 and increased recall of 0.7919 (Figure 15.32).

FIGURE 15.32
A score of 0.37 results in a model accuracy of 0.8507 (85.07%).

Click the Save Score Threshold button to set the score threshold for the ML model to 0.37.

Making Batch Predictions

Now that you have created an ML model, it is time to use the ML model to make predictions. Amazon Machine Learning allows you to use the console to make batch as well as single predictions. In this section you will use the `Titanic_TestDataSource` datasource to create a batch prediction.

A batch prediction involves generating a CSV file with a set of observations, and submitting the observations to Amazon Machine Learning. The results of the prediction will be stored in another CSV file in an S3 bucket. This option is suitable for situations where you do not need predictions in real time.

To create a batch prediction, navigate to the Amazon Machine Learning dashboard and select the Batch Prediction option under the Create New drop-down menu (Figure 15.33).

You will be presented with a list of ML models (Figure 15.34). Click the ML model generated in the previous section.

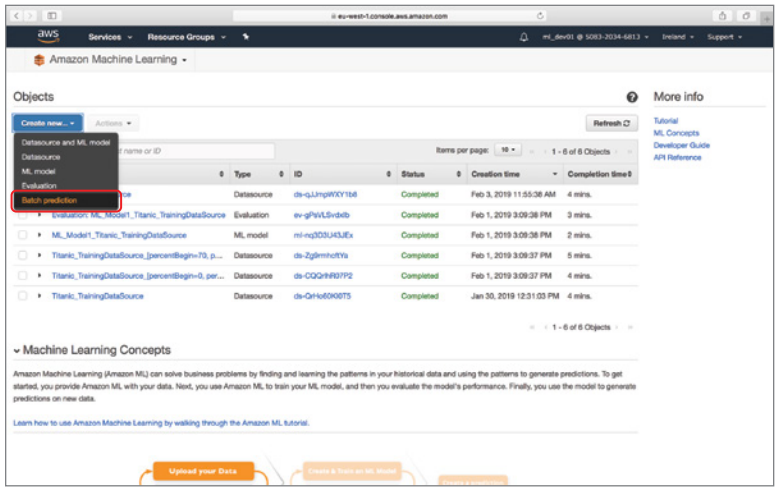


FIGURE 15.33
Accessing the option to create a new batch prediction from the Amazon Machine Learning dashboard

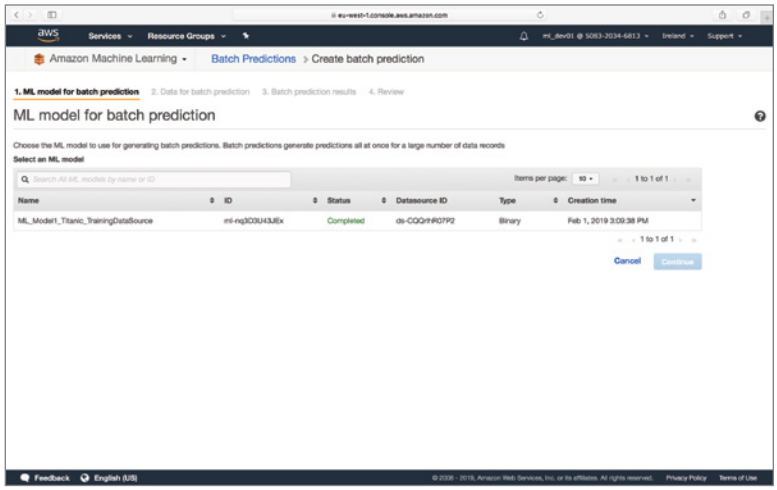


FIGURE 15.34
Selecting an ML model for batch predictions

Amazon Machine Learning will present a brief summary of the ML model, including the AUC score from the most recent evaluation created with the model. Click the Continue button to proceed.

Next, you will be asked to select a datasource for batch predictions (Figure 15.35). Select the `Titanic_TestDataSource` entry from the list.

You will be presented with a summary of the datasource. Click the Continue button to proceed. If you have selected an incorrect datasource, click the Change Datasource button to select a different datasource.

Next, you will be asked to specify an existing Amazon S3 bucket where the results of the batch prediction are to be stored, and provide a name for the batch prediction operation (Figure 15.36).

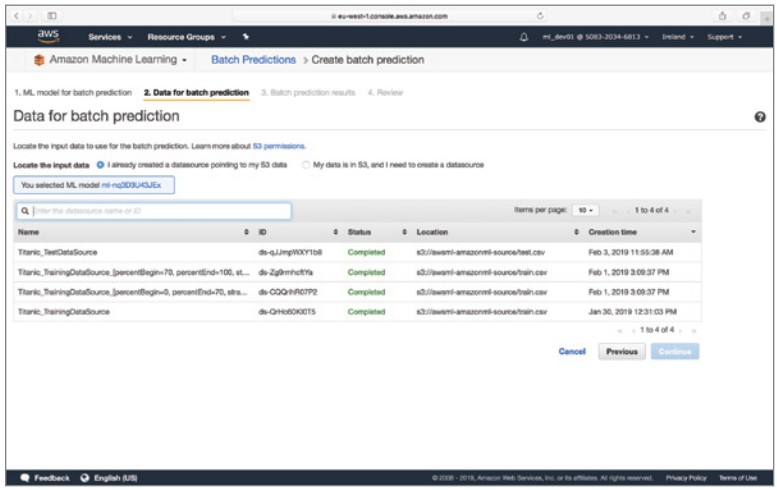


FIGURE 15.35
Selecting a datasource for batch predictions

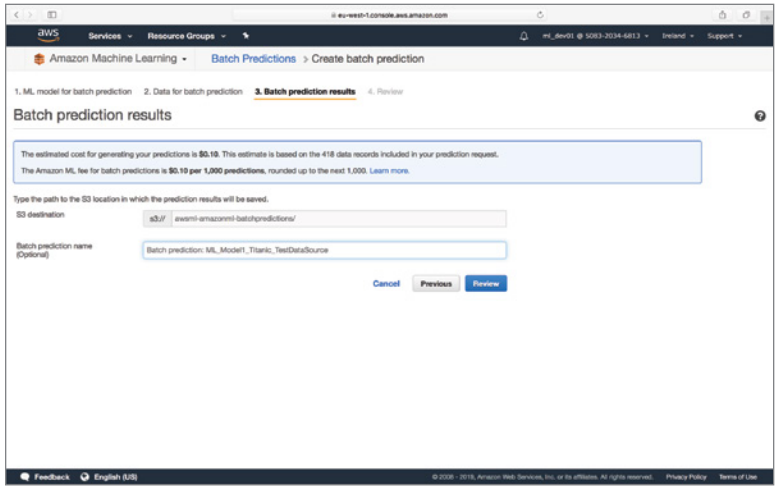


FIGURE 15.36
Specifying an Amazon S3 bucket where the results of the batch prediction are to be stored

At the top of this screen, you will be shown the cost of the batch prediction, which in Figure 15.36 is listed as $0.10 for a CSV file with 418 rows.

In this section, the batch predictions are to be stored in a bucket called `awsml-amazonml-batchpredictions` in the EU (Ireland) region. Choose an existing bucket from your account, in the same region that your Amazon Machine Learning resources are located. Provide a name for the batch prediction that will help you identify the prediction in the dashboard. In this section, the name of the prediction is `Batch Prediction: ML_Model1_Titanic_TestDataSource`. Click the Review button to proceed to the next screen.

You will be taken to the Review screen where you can review the settings for the batch prediction as well as the cost (Figure 15.37). Click the Create button at the bottom of the page to begin the batch prediction process.

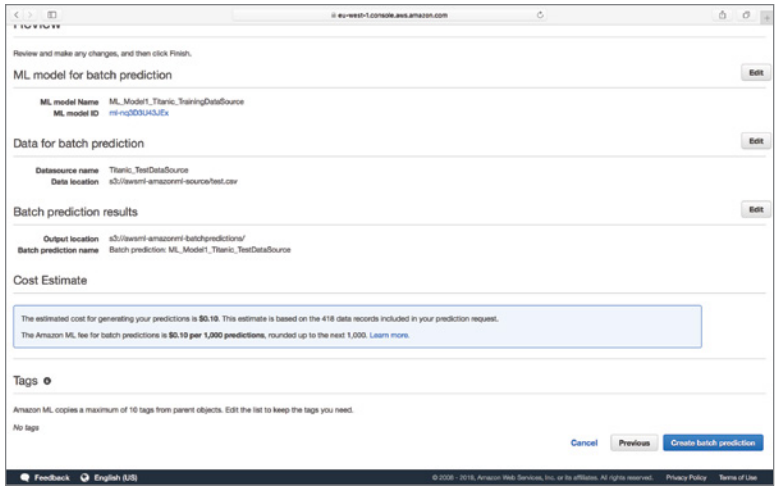


FIGURE 15.37
Batch Prediction Review screen

Navigate to the Amazon Machine Learning dashboard to access the batch prediction. It will take a few minutes for the predictions to be created. When the batch prediction process is complete, the corresponding row in the dashboard will list the prediction status as Complete (Figure 15.38).

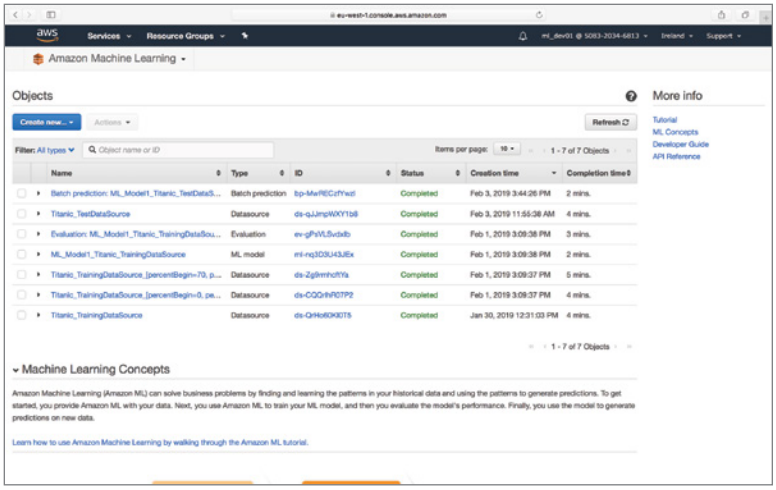


FIGURE 15.38
Amazon Machine Learning dashboard showing a completed batch prediction operation

To access the results of the batch prediction operation, navigate to the Amazon S3 bucket you specified while creating the prediction operation. You will find a `.gz` file containing the results of the prediction (Figure 15.39).

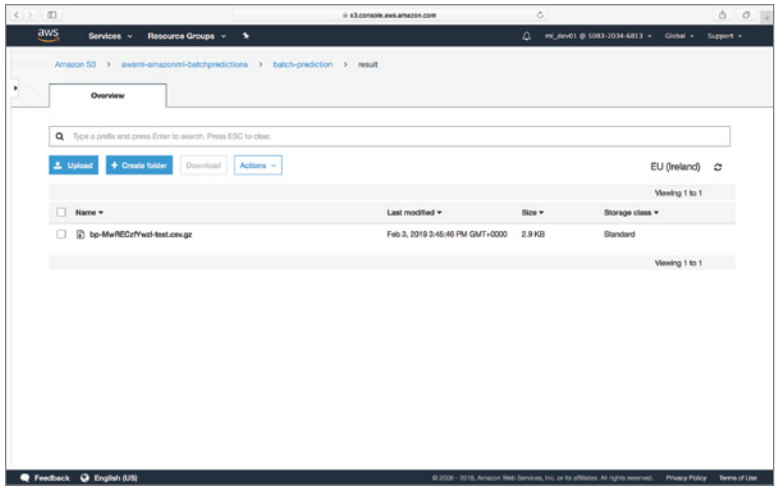


FIGURE 15.39
Amazon S3 Bucket with the results of the batch prediction

Download the file onto your computer and expand it using a suitable program. Open the resulting CSV file with the results of the prediction. Table 15.2 lists the first 10 rows from the file.

TABLE 15.2: The First Ten Rows of the Batch Prediction Result

TAG	BESTANSWER	SCORE
892	0	0.04036516
893	0	0.148552
894	0	0.007585382
895	0	0.01673749
896	1	0.953698
897	0	0.001477312
898	1	0.7763251
899	0	0.2489982
900	1	0.9017848
901	0	0.0003708922

The `tag` column contains the value of the row ID attribute, which was set up as `PassengerId` when the model was created. The `bestAnswer` column contains the prediction from the model obtained after applying the threshold to the output of the model. The `score` column contains the actual output of the model.

Creating a Real-Time Prediction Endpoint for Your Machine Learning Model

In addition to batch predictions, you can use your ML models for real-time predictions. Real-time predictions are synchronous and can only work with one observation at a time. You will typically use real-time predictions when you want to use the ML model generated by Amazon Machine Learning with an application that requires fast real-time predictions, such as a loan processing or credit scoring application. Your application can access the ML model as a stateless microservice, hosted on AWS infrastructure, through a RESTful interface.

To use real-time predictions with your applications, you will need to generate a real-time prediction endpoint. You will be billed for each hour the endpoint is active as well as for each prediction request you submit to the endpoint. Real-time prediction endpoints are only supported in the US East (N. Virginia) and EU (Ireland) regions.

To generate a real-time prediction endpoint, navigate to the Amazon Machine Learning dashboard, and select the check box at the start of the row that contains your ML model. Then click the Actions button and select the Create Real-Time Endpoint option from the drop-down menu (Figure 15.40).

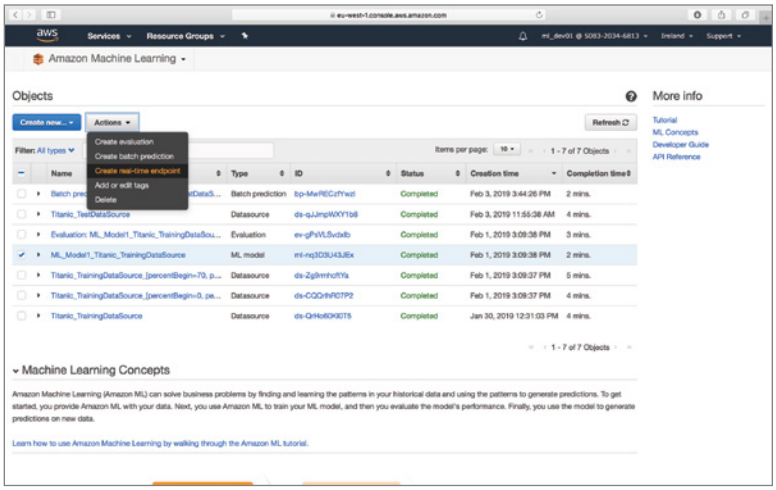


FIGURE 15.40
Creating a real-time prediction endpoint for an Amazon Machine Learning model

You will be informed of the cost of creating the endpoint and the cost of using the endpoint for predictions (Figure 15.41). Click the Create button to create the prediction endpoint.

You can access the endpoint by clicking the name of the ML model in the AWS Machine Learning dashboard and scrolling down to the bottom of the summary page to the section titled Predictions (Figure 15.42).

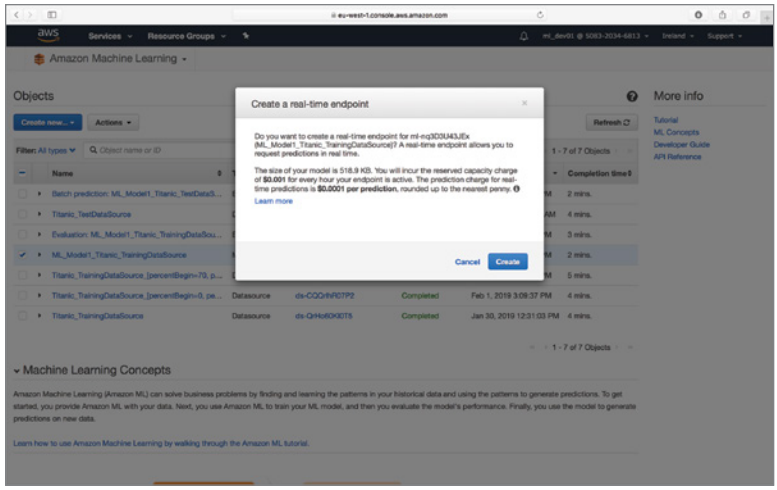


FIGURE 15.41
Costs of maintaining a real-time prediction endpoint

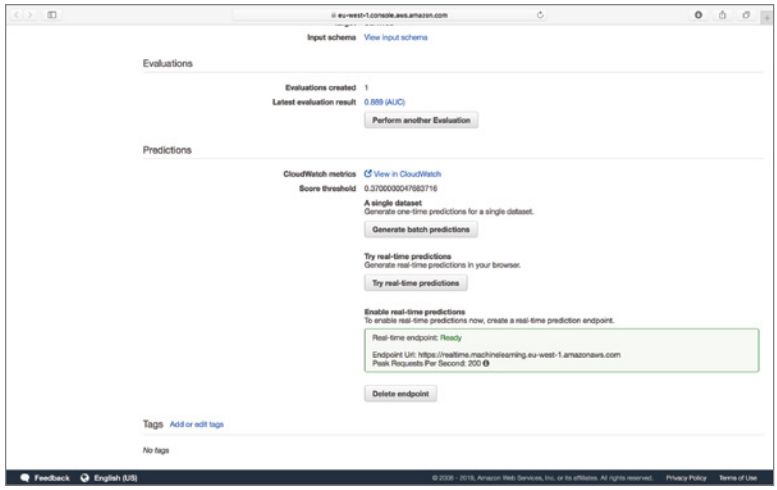


FIGURE 15.42
Accessing the real-time prediction endpoint

Making Predictions Using the AWS CLI

You can use the AWS CLI to access the underlying Amazon Machine Learning APIs and create datasources, build models, create evaluations, and make real-time predictions over the command line. Making real-time predictions with the AWS CLI requires setting up a real-time prediction endpoint.

In this section you will use the AWS CLI to make real-time predictions using the real-time prediction endpoint you created in the previous section. You can get more information on using other aspects of Amazon Machine Learning with the AWS CLI at `https://docs.aws.amazon.com/cli/latest/reference/machinelearning/index.html`.

This section assumes that you have installed and configured the AWS CLI to use your development IAM credentials, and to use the same region in which your Amazon S3 and Amazon Machine Learning resources are located.

To get started, launch a Terminal window on your Mac or a Command Prompt window on Windows, type the following command to retrieve a list of ML models, and press Enter:

```
$ aws machinelearning describe-ml-models
```

The output in your console window should resemble the following:

```
{
    "Results": [
        {
            "MLModelId": "ml-nq3D3U43JEx",
            "TrainingDataSourceId": "ds-CQQrlhR07P2",
            "CreatedByIamUser": "arn:aws:iam::508320346813:user/ml_dev01",
            "CreatedAt": 1549033778.022,
            "LastUpdatedAt": 1549034154.839,
            "Name": "ML_Model1_Titanic_TrainingDataSource",
            "Status": "COMPLETED",
            "SizeInBytes": 531399,
            "EndpointInfo": {
                "PeakRequestsPerSecond": 200,
                "CreatedAt": 1549229683.701,
                "EndpointUrl": "https://realtime.machinelearning.eu-west-1.
amazonaws.com",
                "EndpointStatus": "READY"
            },
            "TrainingParameters": {
                "algorithm": "sgd",
                "sgd.l1RegularizationAmount": "0.0",
                "sgd.l2RegularizationAmount": "1e-6",
                "sgd.maxMLModelSizeInBytes": "104857600",
                "sgd.maxPasses": "10",
                "sgd.shuffleType": "auto"
            },
            "InputDataLocationS3": "s3://awsml-amazonml-source/train.csv",
            "Algorithm": "sgd",
            "MLModelType": "BINARY",
            "ScoreThreshold": 0.3700000047683716,
            "ScoreThresholdLastUpdatedAt": 1549205043.711,
            "ComputeTime": 54000,
            "FinishedAt": 1549034154.839,
            "StartedAt": 1549034025.362
        }
    ]
}
```

The output is a list of all the ML models in your account, along with information on the model. Make a note of the `MLModelId` and the `EndpointUrl` attribute as you will need these to use the model over the command line.

If you get an `AccessDeniedException`, the IAM user does not have the relevant policy that allows that user to access your Amazon Machine Learning resources. In such a case, you need to ensure your IAM development user has the `AmazonMachineLearningFullAccess` policy.

To test the real-time prediction endpoint, type the following command to predict the likelihood of survival of a fictional 18-year-old male passenger, traveling alone, who boarded the Titanic at Southampton. Press Enter on your keyboard after typing the command:

```
$ aws machinelearning predict --ml-model-id ml-nq3D3U43JEx --record
'{"PassengerId":"3","Pclass":"3","Name":"Andrew
Fletcher","Sex":"male","Age":"18","SibSp":"0","Parch":"0","Embarked":"S"}' --
predict-endpoint https://realtime.machinelearning.eu-west-1.amazonaws.com
```

The results in your terminal window should resemble the following:

```
{
    "Prediction": {
        "predictedLabel": "0",
        "predictedScores": {
            "0": 0.0445670448243618
        },
        "details": {
            "Algorithm": "SGD",
            "PredictiveModelType": "BINARY"
        }
    }
}
```

The `predictedLabel` attribute contains the prediction after the score threshold has been applied. The `predictedScores` array contains a single item that contains the output of the ML model.

Using Real-Time Prediction Endpoints with Your Applications

Even though the real-time prediction endpoints generated by Amazon Machine Learning provide API-based access to the ML model, this API does not accept OAuth2 access tokens. The API accepts AWS credentials.

When you access the real-time prediction endpoint using the AWS CLI, the CLI tool uses the credentials your provided as part of the configuration process.

If you intend to use the real-time prediction endpoint from a server-side application that is hosted on AWS infrastructure (such as EC2), you can use policy-based access to ensure that the service running on the EC2 instance is able to access the real-time prediction endpoint.

If you intend to use the real-time prediction endpoint from a client application (such as a mobile app or a web app), you have two choices:

- Embed the credentials (`AccessKeyId`, `SecretAccessKey`) of an IAM user into the client. While this could be a quick solution during development, for most real-world production situations this option is not recommended. The only scenario where you may consider using this option in production is if you have a mechanism in place to safely transport and secure these credentials on the mobile app at runtime.
- Use Amazon Cognito identity pools to get a temporary set of credentials and use these credentials to access the real-time prediction endpoint.

If you would like to learn more about using the real-time prediction API, see `https://docs.aws.amazon.com/machine-learning/latest/dg/requesting-real-time-predictions.html`.

NOTE You can download the code files for this chapter from `www.wiley.com/go/machinelearningawscloud` or from GitHub using the following URL:

```
https://github.com/asmtechnology/awsmlbook-chapter15.git
```

Summary

- Amazon Machine Learning is a fully managed web service that allows you to create and deploy simple machine learning models without any programming.
- Amazon Machine Learning is not included in the AWS free tier.
- Amazon Machine learning is easy to use, but offers limited feature-engineering and model-building capabilities.
- A datasource is an object that stores a reference to an Amazon S3 bucket that stores your input data, along with meta information that describes the characteristics of the input data.
- The schema refers to the structure of the input data, typically a list of attribute names along with their data types.
- Amazon Machine Learning can be used for regression, binary classification, and multi-class classification problems.
- Amazon Machine Learning provides a number of training parameters that you can use to control the quality of the model.
- Amazon Machine Learning provides descriptive statistics that help you analyze your input data.
- A batch prediction involves generating a CSV file with a set of observations, and submitting the observations to Amazon Machine Learning. The results of the prediction will be stored in another CSV file in an S3 bucket.
- In addition to batch predictions, you can use your ML models for real-time predictions. Real-time predictions are synchronous and can only work with one observation at a time.

- To use real-time predictions with your applications, you will need to generate a real-time prediction endpoint. You will be billed for each hour the endpoint is active as well as for each prediction request you submit to the endpoint.
- Real-time prediction endpoints are only supported in the US East (N. Virginia) and EU (Ireland) regions.
- You can use the real-time prediction endpoint with the AWS CLI and language-specific AWS SDKs.