Amazon Machine Learning

API Reference

API Version 2014-12-12

Amazon Machine Learning: API Reference

Table of Contents

Welcome to Amazon Machine Learning

Welcome to the *Amazon Machine Learning API Reference*. Amazon Machine Learning (Amazon ML) is a robust Amazon Web Services (AWS) machine learning platform that allows software developers to build and train predictive applications and host those applications in a scalable AWS cloud solution.

The *Amazon Machine Learning API Reference* explains the Amazon ML application programming interface and describes various API operations, related request and response structures, and error codes.

The current version of the Amazon ML API is 2014-12-12.

Actions

The following actions are supported:

AddTags

Adds one or more tags to an object, up to a limit of 10. Each tag consists of a key and an optional value. If you add a tag using a key that is already associated with the ML object, `AddTags` updates the tag's value.

Request Syntax

```
{
    "ResourceId": "string",
    "ResourceType": "string",
    "Tags": [
        {
            "Key": "string",
            "Value": "string"
        }
    ]
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

ResourceId

The ID of the ML object to tag. For example, `exampleModelId`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

ResourceType

The type of the ML object to tag.

Type: String

Valid Values: `BatchPrediction | DataSource | Evaluation | MLModel`

Required: Yes

Tags

The key-value pairs to use to create tags. If you specify a key without specifying a value, Amazon ML creates a tag with the specified key and a value of null.

Type: array of Tag (p. 136) objects

Required: Yes

Response Syntax

```
{
    "ResourceId": "string",
    "ResourceType": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ResourceId
The ID of the ML object that was tagged.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

ResourceType
The type of the ML object that was tagged.

Type: String

Valid Values: `BatchPrediction | DataSource | Evaluation | MLModel`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

InvalidTagException
HTTP Status Code: 400

ResourceNotFoundException
A specified resource cannot be located.

HTTP Status Code: 400

TagLimitExceededException
HTTP Status Code: 400

Examples

The following is an example of a request and response for the AddTags operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.AddTags
{
  "ResourceId": "exampleModelId",
  "ResourceType": "MLModel",
  "Tags": {
      "Key":"exampleKey",
      "Value":"exampleKeyValue"
  }
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "ResourceId": "exampleModelId",
  "ResourceType": "MLModel"
}
```

CreateBatchPrediction

Generates predictions for a group of observations. The observations to process exist in one or more data files referenced by a `DataSource`. This operation creates a new `BatchPrediction`, and uses an `MLModel` and the data files referenced by the `DataSource` as information sources.

`CreateBatchPrediction` is an asynchronous operation. In response to `CreateBatchPrediction`, Amazon Machine Learning (Amazon ML) immediately returns and sets the `BatchPrediction` status to `PENDING`. After the `BatchPrediction` completes, Amazon ML sets the status to `COMPLETED`.

You can poll for status updates by using the GetBatchPrediction (p. 76) operation and checking the `Status` parameter of the result. After the `COMPLETED` status appears, the results are available in the location specified by the `OutputUri` parameter.

Request Syntax

```
{
    "BatchPredictionDataSourceId": "string",
    "BatchPredictionId": "string",
    "BatchPredictionName": "string",
    "MLModelId": "string",
    "OutputUri": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

BatchPredictionDataSourceId
The ID of the `DataSource` that points to the group of observations to predict.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

BatchPredictionId
A user-supplied ID that uniquely identifies the `BatchPrediction`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

BatchPredictionName
A user-supplied name or description of the `BatchPrediction`. `BatchPredictionName` can only use the UTF-8 character set.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

MLModelId

The ID of the `MLModel` that will generate predictions for the group of observations.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

OutputUri

The location of an Amazon Simple Storage Service (Amazon S3) bucket or directory to store the batch prediction results. The following substrings are not allowed in the `s3 key` portion of the `outputURI` field: ':', '//', '/./', '/../'.

Amazon ML needs permissions to store and retrieve the logs on your behalf. For information about how to set permissions, see the Amazon Machine Learning Developer Guide.

Type: String

Length constraints: Minimum length of 0. Maximum length of 2048.

Pattern: `s3://([^/]+)(/.*)?`

Required: Yes

Response Syntax

```
{
    "BatchPredictionId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

BatchPredictionId

A user-supplied ID that uniquely identifies the `BatchPrediction`. This value is identical to the value of the `BatchPredictionId` in the request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

IdempotentParameterMismatchException
A second request to use or change an object was not allowed. This can result from retrying a request using a parameter that was not present in the original request.

HTTP Status Code: 400

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

Examples

The following is a sample request and response of the BatchPrediction operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.CreateBatchPrediction
{
  "BatchPredictionId": "EXAMPLE-bp-2014-09-12-15-14-04-156",
  "BatchPredictionName": "EXAMPLE",
  "MLModelId": "EXAMPLE-pr-2014-09-12-15-14-04-924",
  "BatchPredictionDataSourceId": "EXAMPLE-tr-ds-2014-09-12-15-14-04-989",
  "OutputUri": "s3://eml-test-EXAMPLE/test-outputs/EXAMPLE-
bp-2014-09-12-15-14-04-156/results"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
```

```
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{"BatchPredictionId":"EXAMPLE-bp-2014-09-12-15-14-04-156"}
```

CreateDataSourceFromRDS

Creates a `DataSource` object from an Amazon Relational Database Service (Amazon RDS). A `DataSource` references data that can be used to perform `CreateMLModel`, `CreateEvaluation`, or `CreateBatchPrediction` operations.

`CreateDataSourceFromRDS` is an asynchronous operation. In response to `CreateDataSourceFromRDS`, Amazon Machine Learning (Amazon ML) immediately returns and sets the `DataSource` status to `PENDING`. After the `DataSource` is created and ready for use, Amazon ML sets the `Status` parameter to `COMPLETED`. `DataSource` in the `COMPLETED` or `PENDING` state can be used only to perform `>CreateMLModel>`, `CreateEvaluation`, or `CreateBatchPrediction` operations.

If Amazon ML cannot accept the input source, it sets the `Status` parameter to `FAILED` and includes an error message in the `Message` attribute of the `GetDataSource` operation response.

Request Syntax

```
{
    "ComputeStatistics": boolean,
    "DataSourceId": "string",
    "DataSourceName": "string",
    "RDSData": {
        "DataRearrangement": "string",
        "DataSchema": "string",
        "DataSchemaUri": "string",
        "DatabaseCredentials": {
            "Password": "string",
            "Username": "string"
        },
        "DatabaseInformation": {
            "DatabaseName": "string",
            "InstanceIdentifier": "string"
        },
        "ResourceRole": "string",
        "S3StagingLocation": "string",
        "SecurityGroupIds": [
            "string"
        ],
        "SelectSqlQuery": "string",
        "ServiceRole": "string",
        "SubnetId": "string"
    },
    "RoleARN": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

ComputeStatistics

The compute statistics for a `DataSource`. The statistics are generated from the observation data referenced by a `DataSource`. Amazon ML uses the statistics internally during `MLModel` training. This parameter must be set to `true` if the DataSource needs to be used for `MLModel` training.

Type: Boolean

Required: No

DataSourceId

A user-supplied ID that uniquely identifies the `DataSource`. Typically, an Amazon Resource Number (ARN) becomes the ID for a `DataSource`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

DataSourceName

A user-supplied name or description of the `DataSource`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

RDSData

The data specification of an Amazon RDS `DataSource`:

- DatabaseInformation -
  - `DatabaseName` - The name of the Amazon RDS database.
  - `InstanceIdentifier` - A unique identifier for the Amazon RDS database instance.
- DatabaseCredentials - AWS Identity and Access Management (IAM) credentials that are used to connect to the Amazon RDS database.
- ResourceRole - A role (DataPipelineDefaultResourceRole) assumed by an EC2 instance to carry out the copy task from Amazon RDS to Amazon Simple Storage Service (Amazon S3). For more information, see Role templates for data pipelines.
- ServiceRole - A role (DataPipelineDefaultRole) assumed by the AWS Data Pipeline service to monitor the progress of the copy task from Amazon RDS to Amazon S3. For more information, see Role templates for data pipelines.
- SecurityInfo - The security information to use to access an RDS DB instance. You need to set up appropriate ingress rules for the security entity IDs provided to allow access to the Amazon RDS instance. Specify a [`SubnetId`, `SecurityGroupIds`] pair for a VPC-based RDS DB instance.
- SelectSqlQuery - A query that is used to retrieve the observation data for the `Datasource`.
- S3StagingLocation - The Amazon S3 location for staging Amazon RDS data. The data retrieved from Amazon RDS using `SelectSqlQuery` is stored in this location.
- DataSchemaUri - The Amazon S3 location of the `DataSchema`.
- DataSchema - A JSON string representing the schema. This is not required if `DataSchemaUri` is specified.
- DataRearrangement - A JSON string that represents the splitting and rearrangement requirements for the `Datasource`.

  Sample - `"{\"splitting\":{\"percentBegin\":10,\"percentEnd\":60}}"`

Type: RDSDataSpec (p. 123) object

Required: Yes

RoleARN
The role that Amazon ML assumes on behalf of the user to create and activate a data pipeline in the user's account and copy data using the `SelectSqlQuery` query from Amazon RDS to Amazon S3.

Type: String

Length constraints: Minimum length of 1. Maximum length of 110.

Required: Yes

Response Syntax

```
{
    "DataSourceId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

DataSourceId
A user-supplied ID that uniquely identifies the datasource. This value should be identical to the value of the `DataSourceID` in the request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

IdempotentParameterMismatchException
A second request to use or change an object was not allowed. This can result from retrying a request using a parameter that was not present in the original request.

HTTP Status Code: 400

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

Examples

The following is a sample HTTP request and response of the CreateDataSourceFromRDS operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.CreateDataSourceFromRDS
{
  "DataSourceId": "ml-rds-data-source-demo",
  "DataSourceName": "ml-rds-data-source-demo",
  "RDSData":
  {
    "DatabaseInformation":
    {
      "InstanceIdentifier": "demo",
      "DatabaseName": "demo"
    },
    "SelectSqlQuery": "select feature1, feature2, feature3, ...., featureN
 from RDS_DEMO_TABLE;",
    "DatabaseCredentials":
    {
      "Username": "demo_user",
      "Password": "demo_password"
    },
    "S3StagingLocation": "s3://mldemo/data/",
    "DataSchemaUri": "s3://mldemo/schema/mldemo.csv.schema",
    "ResourceRole": "DataPipelineDefaultResourceRole",
    "ServiceRole": "DataPipelineDefaultRole",
    "SubnetId": "subnet-XXXX",
    "SecurityGroupIds":
      ["sg-XXXXXX", "sg-XXXXXX"]
  },
  "RoleARN": "arn:aws:iam::<awsAccountId>:role/<roleToAssume>"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
```

```
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "DataSourceId":"ml-rds-data-source-demo"
}
```

CreateDataSourceFromRedshift

Creates a `DataSource` from a database hosted on an Amazon Redshift cluster. A `DataSource` references data that can be used to perform either `CreateMLModel`, `CreateEvaluation`, or `CreateBatchPrediction` operations.

`CreateDataSourceFromRedshift` is an asynchronous operation. In response to `CreateDataSourceFromRedshift`, Amazon Machine Learning (Amazon ML) immediately returns and sets the `DataSource` status to `PENDING`. After the `DataSource` is created and ready for use, Amazon ML sets the `Status` parameter to `COMPLETED`. `DataSource` in `COMPLETED` or `PENDING` states can be used to perform only `CreateMLModel`, `CreateEvaluation`, or `CreateBatchPrediction` operations.

If Amazon ML can't accept the input source, it sets the `Status` parameter to `FAILED` and includes an error message in the `Message` attribute of the `GetDataSource` operation response.

The observations should be contained in the database hosted on an Amazon Redshift cluster and should be specified by a `SelectSqlQuery` query. Amazon ML executes an `Unload` command in Amazon Redshift to transfer the result set of the `SelectSqlQuery` query to `S3StagingLocation`.

After the `DataSource` has been created, it's ready for use in evaluations and batch predictions. If you plan to use the `DataSource` to train an `MLModel`, the `DataSource` also requires a recipe. A recipe describes how each input variable will be used in training an `MLModel`. Will the variable be included or excluded from training? Will the variable be manipulated; for example, will it be combined with another variable or will it be split apart into word combinations? The recipe provides answers to these questions.

You can't change an existing datasource, but you can copy and modify the settings from an existing Amazon Redshift datasource to create a new datasource. To do so, call `GetDataSource` for an existing datasource and copy the values to a `CreateDataSource` call. Change the settings that you want to change and make sure that all required fields have the appropriate values.

Request Syntax

```
{
    "ComputeStatistics": boolean,
    "DataSourceId": "string",
    "DataSourceName": "string",
    "DataSpec": {
        "DataRearrangement": "string",
        "DataSchema": "string",
        "DataSchemaUri": "string",
        "DatabaseCredentials": {
            "Password": "string",
            "Username": "string"
        },
        "DatabaseInformation": {
            "ClusterIdentifier": "string",
            "DatabaseName": "string"
        },
        "S3StagingLocation": "string",
        "SelectSqlQuery": "string"
    },
    "RoleARN": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

ComputeStatistics
The compute statistics for a `DataSource`. The statistics are generated from the observation data referenced by a `DataSource`. Amazon ML uses the statistics internally during `MLModel` training. This parameter must be set to `true` if the `DataSource` needs to be used for `MLModel` training.

Type: Boolean

Required: No

DataSourceId
A user-supplied ID that uniquely identifies the `DataSource`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

DataSourceName
A user-supplied name or description of the `DataSource`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

DataSpec
The data specification of an Amazon Redshift `DataSource`:

- DatabaseInformation -
  - `DatabaseName` - The name of the Amazon Redshift database.
  - `ClusterIdentifier` - The unique ID for the Amazon Redshift cluster.
- DatabaseCredentials - The AWS Identity and Access Management (IAM) credentials that are used to connect to the Amazon Redshift database.
- SelectSqlQuery - The query that is used to retrieve the observation data for the `Datasource`.
- S3StagingLocation - The Amazon Simple Storage Service (Amazon S3) location for staging Amazon Redshift data. The data retrieved from Amazon Redshift using the `SelectSqlQuery` query is stored in this location.
- DataSchemaUri - The Amazon S3 location of the `DataSchema`.
- DataSchema - A JSON string representing the schema. This is not required if `DataSchemaUri` is specified.
- DataRearrangement - A JSON string that represents the splitting and rearrangement requirements for the `DataSource`.

  Sample - `"{\"splitting\":{\"percentBegin\":10,\"percentEnd\":60}}"`

Type: RedshiftDataSpec (p. 130) object

Required: Yes

RoleARN
A fully specified role Amazon Resource Name (ARN). Amazon ML assumes the role on behalf of the user to create the following:

- A security group to allow Amazon ML to execute the `SelectSqlQuery` query on an Amazon Redshift cluster
- An Amazon S3 bucket policy to grant Amazon ML read/write permissions on the `S3StagingLocation`

Type: String

Length constraints: Minimum length of 1. Maximum length of 110.

Required: Yes

Response Syntax

```
{
    "DataSourceId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

DataSourceId
A user-supplied ID that uniquely identifies the datasource. This value should be identical to the value of the `DataSourceID` in the request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

IdempotentParameterMismatchException
A second request to use or change an object was not allowed. This can result from retrying a request using a parameter that was not present in the original request.

HTTP Status Code: 400

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

Examples

The following is a sample request and response of the CreateDataSourceFromRedshift operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.CreateDataSourceFromRedshift
{
  "DataSourceId": "ds-exampleDatasourceId",
  "DataSourceName": "exampleDatasourceName",
  "DataSpec":
  {
    "DatabaseInformation":
    {
      "DatabaseName": "dev",
      "ClusterIdentifier": "test-cluster-1234"
    },
    "SelectSqlQuery": "select * from table",
    "DatabaseCredentials":
    {
      "Username": "foo",
      "Password": "foo"
    },
   "S3StagingLocation": "s3://bucketName/",
   "DataSchemaUri": "s3://bucketName/locationToUri/example.schema.json"},
   "RoleARN": "arn:aws:iam::<awsAccountId>:role/username"
  }
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{"DataSourceId": "ds-exampleDatasourceId"}
```

CreateDataSourceFromS3

Creates a `DataSource` object. A `DataSource` references data that can be used to perform `CreateMLModel`, `CreateEvaluation`, or `CreateBatchPrediction` operations.

`CreateDataSourceFromS3` is an asynchronous operation. In response to `CreateDataSourceFromS3`, Amazon Machine Learning (Amazon ML) immediately returns and sets the `DataSource` status to `PENDING`. After the `DataSource` has been created and is ready for use, Amazon ML sets the `Status` parameter to `COMPLETED`. `DataSource` in the `COMPLETED` or `PENDING` state can be used to perform only `CreateMLModel`, `CreateEvaluation` or `CreateBatchPrediction` operations.

If Amazon ML can't accept the input source, it sets the `Status` parameter to `FAILED` and includes an error message in the `Message` attribute of the `GetDataSource` operation response.

The observation data used in a `DataSource` should be ready to use; that is, it should have a consistent structure, and missing data values should be kept to a minimum. The observation data must reside in one or more .csv files in an Amazon Simple Storage Service (Amazon S3) location, along with a schema that describes the data items by name and type. The same schema must be used for all of the data files referenced by the `DataSource`.

After the `DataSource` has been created, it's ready to use in evaluations and batch predictions. If you plan to use the `DataSource` to train an `MLModel`, the `DataSource` also needs a recipe. A recipe describes how each input variable will be used in training an `MLModel`. Will the variable be included or excluded from training? Will the variable be manipulated; for example, will it be combined with another variable or will it be split apart into word combinations? The recipe provides answers to these questions.

Request Syntax

```
{
    "ComputeStatistics": boolean,
    "DataSourceId": "string",
    "DataSourceName": "string",
    "DataSpec": {
        "DataLocationS3": "string",
        "DataRearrangement": "string",
        "DataSchema": "string",
        "DataSchemaLocationS3": "string"
    }
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

ComputeStatistics

The compute statistics for a `DataSource`. The statistics are generated from the observation data referenced by a `DataSource`. Amazon ML uses the statistics internally during `MLModel` training. This parameter must be set to `true` if the DataSource needs to be used for `MLModel` training.

Type: Boolean

Required: No

DataSourceId

A user-supplied identifier that uniquely identifies the `DataSource`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

DataSourceName

A user-supplied name or description of the `DataSource`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

DataSpec

The data specification of a `DataSource`:

- DataLocationS3 - The Amazon S3 location of the observation data.
- DataSchemaLocationS3 - The Amazon S3 location of the `DataSchema`.
- DataSchema - A JSON string representing the schema. This is not required if `DataSchemaUri` is specified.
- DataRearrangement - A JSON string that represents the splitting and rearrangement requirements for the `Datasource`.

  Sample - `"{\"splitting\":{\"percentBegin\":10,\"percentEnd\":60}}"`

Type: S3DataSpec (p. 133) object

Required: Yes

Response Syntax

```
{
    "DataSourceId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

DataSourceId

A user-supplied ID that uniquely identifies the `DataSource`. This value should be identical to the value of the `DataSourceID` in the request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

IdempotentParameterMismatchException
A second request to use or change an object was not allowed. This can result from retrying a request using a parameter that was not present in the original request.

HTTP Status Code: 400

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

Examples

The following is a sample request and response of the CreateDataSourceFromS3 operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.CreateDataSourceFromS3
{
  "DataSourceId": "exampleDataSourceId",
  "DataSourceName": "exampleDataSourceName",
  "DataSpec":
  {
    "DataLocationS3": "s3://eml-test-EXAMPLE/data.csv",
    "DataSchemaLocationS3": "s3://eml-test-EXAMPLE/data.csv.schema",
    "DataRearrangement": "{\"splitting\":{\"percentBegin\":10,\"percentEnd
\":60}}"
  }
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{"DataSourceId":"exampleDataSourceId"}
```

CreateEvaluation

Creates a new `Evaluation` of an `MLModel`. An `MLModel` is evaluated on a set of observations associated to a `DataSource`. Like a `DataSource` for an `MLModel`, the `DataSource` for an `Evaluation` contains values for the `Target Variable`. The `Evaluation` compares the predicted result for each observation to the actual outcome and provides a summary so that you know how effective the `MLModel` functions on the test data. Evaluation generates a relevant performance metric, such as BinaryAUC, RegressionRMSE or MulticlassAvgFScore based on the corresponding `MLModelType`: `BINARY`, `REGRESSION` or `MULTICLASS`.

`CreateEvaluation` is an asynchronous operation. In response to `CreateEvaluation`, Amazon Machine Learning (Amazon ML) immediately returns and sets the evaluation status to `PENDING`. After the `Evaluation` is created and ready for use, Amazon ML sets the status to `COMPLETED`.

You can use the `GetEvaluation` operation to check progress of the evaluation during the creation operation.

Request Syntax

```
{
    "EvaluationDataSourceId": "string",
    "EvaluationId": "string",
    "EvaluationName": "string",
    "MLModelId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

EvaluationDataSourceId
The ID of the `DataSource` for the evaluation. The schema of the `DataSource` must match the schema used to create the `MLModel`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

EvaluationId
A user-supplied ID that uniquely identifies the `Evaluation`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

EvaluationName
A user-supplied name or description of the `Evaluation`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

MLModelId

The ID of the `MLModel` to evaluate.

The schema used in creating the `MLModel` must match the schema of the `DataSource` used in the `Evaluation`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

Response Syntax

```
{
    "EvaluationId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

EvaluationId

The user-supplied ID that uniquely identifies the `Evaluation`. This value should be identical to the value of the `EvaluationId` in the request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

IdempotentParameterMismatchException

A second request to use or change an object was not allowed. This can result from retrying a request using a parameter that was not present in the original request.

HTTP Status Code: 400

InternalServerException

An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException

An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

Examples

The following is a sample request and response of the CreateEvaluation operation:

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.CreateEvaluation
{
  "EvaluationId": "CreateEvaluation-pr-2014-09-12-15-14-04-924",
  "EvaluationName": "EXAMPLE",
  "MLModelId": "EXAMPLE-pr-2014-09-12-15-14-04-924",
  "EvaluationDataSourceId": "EXAMPLE-ev-ds-2014-09-12-15-14-04-411",
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{"EvaluationId":"CreateEvaluation-pr-2014-09-12-15-14-04-924"}
```

CreateMLModel

Creates a new `MLModel` using the `DataSource` and the recipe as information sources.

An `MLModel` is nearly immutable. Users can update only the `MLModelName` and the `ScoreThreshold` in an `MLModel` without creating a new `MLModel`.

`CreateMLModel` is an asynchronous operation. In response to `CreateMLModel`, Amazon Machine Learning (Amazon ML) immediately returns and sets the `MLModel` status to `PENDING`. After the `MLModel` has been created and ready is for use, Amazon ML sets the status to `COMPLETED`.

You can use the `GetMLModel` operation to check the progress of the `MLModel` during the creation operation.

`CreateMLModel` requires a `DataSource` with computed statistics, which can be created by setting `ComputeStatistics` to `true` in `CreateDataSourceFromRDS`, `CreateDataSourceFromS3`, or `CreateDataSourceFromRedshift` operations.

Request Syntax

```
{
    "MLModelId": "string",
    "MLModelName": "string",
    "MLModelType": "string",
    "Parameters":
        {
            "string" :
                "string"
        },
    "Recipe": "string",
    "RecipeUri": "string",
    "TrainingDataSourceId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

MLModelId
A user-supplied ID that uniquely identifies the `MLModel`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

MLModelName
A user-supplied name or description of the `MLModel`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

MLModelType

The category of supervised learning that this `MLModel` will address. Choose from the following types:

- Choose `REGRESSION` if the `MLModel` will be used to predict a numeric value.
- Choose `BINARY` if the `MLModel` result has two possible values.
- Choose `MULTICLASS` if the `MLModel` result has a limited number of values.

For more information, see the Amazon Machine Learning Developer Guide.

Type: String

Valid Values: `REGRESSION | BINARY | MULTICLASS`

Required: Yes

Parameters

A list of the training parameters in the `MLModel`. The list is implemented as a map of key-value pairs.

The following is the current set of training parameters:

- `sgd.maxMLModelSizeInBytes` - The maximum allowed size of the model. Depending on the input data, the size of the model might affect its performance.

  The value is an integer that ranges from `100000` to `2147483648`. The default value is `33554432`.
- `sgd.maxPasses` - The number of times that the training process traverses the observations to build the `MLModel`. The value is an integer that ranges from `1` to `10000`. The default value is `10`.
- `sgd.shuffleType` - Whether Amazon ML shuffles the training data. Shuffling the data improves a model's ability to find the optimal solution for a variety of data types. The valid values are `auto` and `none`. The default value is `none`. We strongly recommend that you shuffle your data.
- `sgd.l1RegularizationAmount` - The coefficient regularization L1 norm. It controls overfitting the data by penalizing large coefficients. This tends to drive coefficients to zero, resulting in a sparse feature set. If you use this parameter, start by specifying a small value, such as `1.0E-08`.

  The value is a double that ranges from `0` to `MAX_DOUBLE`. The default is to not use L1 normalization. This parameter can't be used when `L2` is specified. Use this parameter sparingly.
- `sgd.l2RegularizationAmount` - The coefficient regularization L2 norm. It controls overfitting the data by penalizing large coefficients. This tends to drive coefficients to small, nonzero values. If you use this parameter, start by specifying a small value, such as `1.0E-08`.

  The value is a double that ranges from `0` to `MAX_DOUBLE`. The default is to not use L2 normalization. This parameter can't be used when `L1` is specified. Use this parameter sparingly.

Type: String to String map

Required: No

Recipe

The data recipe for creating the `MLModel`. You must specify either the recipe or its URI. If you don't specify a recipe or its URI, Amazon ML creates a default.

Type: String

Length constraints: Minimum length of 0. Maximum length of 131071.

Required: No

RecipeUri

The Amazon Simple Storage Service (Amazon S3) location and file name that contains the `MLModel` recipe. You must specify either the recipe or its URI. If you don't specify a recipe or its URI, Amazon ML creates a default.

Type: String

Length constraints: Minimum length of 0. Maximum length of 2048.

Pattern: `s3://([^/]+)(/.*)?`

Required: No

TrainingDataSourceId

The `DataSource` that points to the training data.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

Response Syntax

```
{
    "MLModelId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

MLModelId

A user-supplied ID that uniquely identifies the `MLModel`. This value should be identical to the value of the `MLModelId` in the request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

IdempotentParameterMismatchException

A second request to use or change an object was not allowed. This can result from retrying a request using a parameter that was not present in the original request.

HTTP Status Code: 400

InternalServerException

An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException

An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

Examples

The following is a sample request and response of the CreateMLModel operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.CreateMLModel
{
  "MLModelId": "exampleModelId",
  "MLModelName": "EXAMPLE",
  "MLModelType": "BINARY",
  "TrainingDataSourceId": "17SdAv6WC6r5vACAxF7U",
  "RecipeUri": "s3://eml-test-EXAMPLE/data.recipe.json"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{"MLModelId":"exampleModelId"}
```

CreateRealtimeEndpoint

Creates a real-time endpoint for the `MLModel`. The endpoint contains the URI of the `MLModel`; that is, the location to send real-time prediction requests for the specified `MLModel`.

Request Syntax

```
{
    "MLModelId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

MLModelId
The ID assigned to the `MLModel` during creation.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

Response Syntax

```
{
    "MLModelId": "string",
    "RealtimeEndpointInfo": {
        "CreatedAt": number,
        "EndpointStatus": "string",
        "EndpointUrl": "string",
        "PeakRequestsPerSecond": number
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

MLModelId
A user-supplied ID that uniquely identifies the `MLModel`. This value should be identical to the value of the `MLModelId` in the request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

RealtimeEndpointInfo
The endpoint information of the `MLModel`

Type: RealtimeEndpointInfo (p. 128) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

ResourceNotFoundException
A specified resource cannot be located.

HTTP Status Code: 400

Examples

The following is a sample request and response of the CreateRealtimeEndpoint operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.CreateRealtimeEndpoint
{
  "MLModelId": "ml-ModelExampleId",
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "MLModelId": "ml-ModelExampleId",
  "EndpointInfo":
  {
    "CreatedAt": 1422488124.71,
    "EndpointUrl": "<realtime endpoint from Amazon Machine Learning for ml-
ModelExampleId>",
    "EndpointStatus": "READY",
    "PeakRequestsPerSecond": 200
  }
}
```

DeleteBatchPrediction

Assigns the DELETED status to a `BatchPrediction`, rendering it unusable.

After using the `DeleteBatchPrediction` operation, you can use the GetBatchPrediction (p. 76) operation to verify that the status of the `BatchPrediction` changed to DELETED.

Caution: The result of the `DeleteBatchPrediction` operation is irreversible.

Request Syntax

```
{
    "BatchPredictionId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

BatchPredictionId
A user-supplied ID that uniquely identifies the `BatchPrediction`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

Response Syntax

```
{
    "BatchPredictionId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

BatchPredictionId
A user-supplied ID that uniquely identifies the `BatchPrediction`. This value should be identical to the value of the `BatchPredictionID` in the request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

ResourceNotFoundException
A specified resource cannot be located.

HTTP Status Code: 400

Examples

The following is a sample request and response of the DeleteBatchPrediction operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.DeleteBatchPrediction
{"BatchPredictionId": "exampleBatchPredictionId"}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
```

```
{"BatchPredictionId":"exampleBatchPredictionId"}
```

DeleteDataSource

Assigns the DELETED status to a `DataSource`, rendering it unusable.

After using the `DeleteDataSource` operation, you can use the GetDataSource (p. 80) operation to verify that the status of the `DataSource` changed to DELETED.

Caution: The results of the `DeleteDataSource` operation are irreversible.

Request Syntax

```
{
    "DataSourceId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

DataSourceId
A user-supplied ID that uniquely identifies the `DataSource`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

Response Syntax

```
{
    "DataSourceId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

DataSourceId
A user-supplied ID that uniquely identifies the `DataSource`. This value should be identical to the value of the `DataSourceID` in the request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

ResourceNotFoundException
A specified resource cannot be located.

HTTP Status Code: 400

Examples

The following is a sample request and response of the DeleteDataSource operation:

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.DeleteDataSource
{"DataSourceId": "exampleDataSourceId"}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
```

```
{"DataSourceId":"exampleDataSourceId"}
```

DeleteEvaluation

Assigns the `DELETED` status to an `Evaluation`, rendering it unusable.

After invoking the `DeleteEvaluation` operation, you can use the `GetEvaluation` operation to verify that the status of the `Evaluation` changed to `DELETED`.

Request Syntax

```
{
    "EvaluationId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

EvaluationId
A user-supplied ID that uniquely identifies the `Evaluation` to delete.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

Response Syntax

```
{
    "EvaluationId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

EvaluationId
A user-supplied ID that uniquely identifies the `Evaluation`. This value should be identical to the value of the `EvaluationId` in the request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

ResourceNotFoundException
A specified resource cannot be located.

HTTP Status Code: 400

Examples

The following is a sample request and response of the DeleteEvaluation operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.DeleteEvaluation
{"EvaluationId": "exampleEvaluationId"}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{"EvaluationId":"exampleEvaluationId"}
```

DeleteMLModel

Assigns the `DELETED` status to an `MLModel`, rendering it unusable.

After using the `DeleteMLModel` operation, you can use the `GetMLModel` operation to verify that the status of the `MLModel` changed to DELETED.

Caution: The result of the `DeleteMLModel` operation is irreversible.

Request Syntax

```
{
    "MLModelId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

MLModelId
A user-supplied ID that uniquely identifies the `MLModel`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

Response Syntax

```
{
    "MLModelId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

MLModelId
A user-supplied ID that uniquely identifies the `MLModel`. This value should be identical to the value of the `MLModelID` in the request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

ResourceNotFoundException
A specified resource cannot be located.

HTTP Status Code: 400

Examples

The following is a sample request and response of the DeleteMLModel operation:

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.DeleteMLModel
{"MLModelId": "exampleMLModelId"}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
```

```
{"MLModelId": "exampleMLModelId"}
```

DeleteRealtimeEndpoint

Deletes a real time endpoint of an `MLModel`.

Request Syntax

```
{
    "MLModelId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

MLModelId
The ID assigned to the `MLModel` during creation.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

Response Syntax

```
{
    "MLModelId": "string",
    "RealtimeEndpointInfo": {
        "CreatedAt": number,
        "EndpointStatus": "string",
        "EndpointUrl": "string",
        "PeakRequestsPerSecond": number
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

MLModelId
A user-supplied ID that uniquely identifies the `MLModel`. This value should be identical to the value of the `MLModelId` in the request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

RealtimeEndpointInfo
The endpoint information of the `MLModel`

Type: RealtimeEndpointInfo (p. 128) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

ResourceNotFoundException
A specified resource cannot be located.

HTTP Status Code: 400

Examples

The following is a sample request and response of the DeleteRealtimeEndpoint operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.DeleteRealtimeEndpoint
{
  "MLModelId": "ml-ModelExampleId",
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "MLModelId": "ml-ModelExampleId",
  "EndpointInfo":
  {
    "EndpointStatus": "NONE",
    "PeakRequestsPerSecond": 0
  }
}
```

DeleteTags

Deletes the specified tags associated with an ML object. After this operation is complete, you can't recover deleted tags.

If you specify a tag that doesn't exist, Amazon ML ignores it.

Request Syntax

```
{
    "ResourceId": "string",
    "ResourceType": "string",
    "TagKeys": [
        "string"
    ]
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

ResourceId
The ID of the tagged ML object. For example, `exampleModelId`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

ResourceType
The type of the tagged ML object.

Type: String

Valid Values: `BatchPrediction | DataSource | Evaluation | MLModel`

Required: Yes

TagKeys
One or more tags to delete.

Type: array of Strings

Required: Yes

Response Syntax

```
{
    "ResourceId": "string",
    "ResourceType": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ResourceId
The ID of the ML object from which tags were deleted.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

ResourceType
The type of the ML object from which tags were deleted.

Type: String

Valid Values: `BatchPrediction | DataSource | Evaluation | MLModel`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

InvalidTagException
HTTP Status Code: 400

ResourceNotFoundException
A specified resource cannot be located.

HTTP Status Code: 400

Examples

The following are an example request and response for the DeleteTags operation.

Sample Request

```
POST / HTTP/1.1
```

```
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.DeleteTags
{
  "ResourceId": "exampleModelId",
  "ResourceType": "MLModel",
  "Tags": [
      "exampleKey"
  ]
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "ResourceId": "exampleModelId",
  "ResourceType": "MLModel"
}
```

DescribeBatchPredictions

Returns a list of `BatchPrediction` operations that match the search criteria in the request.

Request Syntax

```
{
    "EQ": "string",
    "FilterVariable": "string",
    "GE": "string",
    "GT": "string",
    "LE": "string",
    "Limit": number,
    "LT": "string",
    "NE": "string",
    "NextToken": "string",
    "Prefix": "string",
    "SortOrder": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

EQ

The equal to operator. The `BatchPrediction` results will have `FilterVariable` values that exactly match the value specified with `EQ`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

FilterVariable

Use one of the following variables to filter a list of `BatchPrediction`:

- `CreatedAt` - Sets the search criteria to the `BatchPrediction` creation date.
- `Status` - Sets the search criteria to the `BatchPrediction` status.
- `Name` - Sets the search criteria to the contents of the `BatchPrediction` `Name`.
- `IAMUser` - Sets the search criteria to the user account that invoked the `BatchPrediction` creation.
- `MLModelId` - Sets the search criteria to the `MLModel` used in the `BatchPrediction`.
- `DataSourceId` - Sets the search criteria to the `DataSource` used in the `BatchPrediction`.
- `DataURI` - Sets the search criteria to the data file(s) used in the `BatchPrediction`. The URL can identify either a file or an Amazon Simple Storage Solution (Amazon S3) bucket or directory.

Type: String

Valid Values: `CreatedAt | LastUpdatedAt | Status | Name | IAMUser | MLModelId | DataSourceId | DataURI`

Required: No

GE

The greater than or equal to operator. The `BatchPrediction` results will have `FilterVariable` values that are greater than or equal to the value specified with `GE`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

GT

The greater than operator. The `BatchPrediction` results will have `FilterVariable` values that are greater than the value specified with `GT`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

LE

The less than or equal to operator. The `BatchPrediction` results will have `FilterVariable` values that are less than or equal to the value specified with `LE`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

Limit

The number of pages of information to include in the result. The range of acceptable values is `1` through `100`. The default value is `100`.

Type: Number

Valid range: Minimum value of 1. Maximum value of 100.

Required: No

LT

The less than operator. The `BatchPrediction` results will have `FilterVariable` values that are less than the value specified with `LT`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

NE

The not equal to operator. The `BatchPrediction` results will have `FilterVariable` values not equal to the value specified with `NE`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

NextToken

An ID of the page in the paginated results.

Type: String

Required: No

Prefix

A string that is found at the beginning of a variable, such as `Name` or `Id`.

For example, a `Batch Prediction` operation could have the `Name` `2014-09-09-HolidayGiftMailer`. To search for this `BatchPrediction`, select `Name` for the `FilterVariable` and any of the following strings for the `Prefix`:

- 2014-09
- 2014-09-09
- 2014-09-09-Holiday

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

SortOrder

A two-value parameter that determines the sequence of the resulting list of `MLModels`.

- `asc` - Arranges the list in ascending order (A-Z, 0-9).
- `dsc` - Arranges the list in descending order (Z-A, 9-0).

Results are sorted by `FilterVariable`.

Type: String

Valid Values: `asc | dsc`

Required: No

Response Syntax

```
{
    "NextToken": "string",
    "Results": [
        {
            "BatchPredictionDataSourceId": "string",
            "BatchPredictionId": "string",
            "CreatedAt": number,
            "CreatedByIamUser": "string",
            "InputDataLocationS3": "string",
            "LastUpdatedAt": number,
            "MLModelId": "string",
            "Message": "string",
```

```
            "Name": "string",
            "OutputUri": "string",
            "Status": "string"
        }
    ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken
The ID of the next page in the paginated results that indicates at least one more page follows.

Type: String

Results
A list of `BatchPrediction` objects that meet the search criteria.

Type: array of BatchPrediction (p. 111) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

Examples

The following is a sample request and response of the DescribeBatchPredictions operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.DescribeBatchPredictions
```

```
{
  "FilterVariable": "Name",
  "Prefix": "bp-",
  "SortOrder": "asc",
  "Limit": 1
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "Results": [
    {
      "BatchPredictionDataSourceId": "ds-exampleDataSourceId",
      "BatchPredictionId": "bp-exampleBatchPredictionId",
      "CreatedAt": 1422057670.697,
      "CreatedByIamUser": "arn:aws:iam::<awsAccountId>:user/username",
      "InputDataLocationS3": "s3://bucket/locationToInput/example-
data.testing.csv",
      "LastUpdatedAt": 1422057811.431,
      "MLModelId": "pr-exampleModelId",
      "Name": "bp-exampleBatchPredictionName",
      "OutputUri": "s3://bucket/locationToLogs/",
      "Status": "COMPLETED"
    }
  ]
}
```

DescribeDataSources

Returns a list of `DataSource` that match the search criteria in the request.

Request Syntax

```
{
    "EQ": "string",
    "FilterVariable": "string",
    "GE": "string",
    "GT": "string",
    "LE": "string",
    "Limit": number,
    "LT": "string",
    "NE": "string",
    "NextToken": "string",
    "Prefix": "string",
    "SortOrder": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

EQ

The equal to operator. The `DataSource` results will have `FilterVariable` values that exactly match the value specified with `EQ`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

FilterVariable

Use one of the following variables to filter a list of `DataSource`:

- `CreatedAt` - Sets the search criteria to `DataSource` creation dates.
- `Status` - Sets the search criteria to `DataSource` statuses.
- `Name` - Sets the search criteria to the contents of `DataSource` `Name`.
- `DataUri` - Sets the search criteria to the URI of data files used to create the `DataSource`. The URI can identify either a file or an Amazon Simple Storage Service (Amazon S3) bucket or directory.
- `IAMUser` - Sets the search criteria to the user account that invoked the `DataSource` creation.

Type: String

Valid Values: `CreatedAt | LastUpdatedAt | Status | Name | DataLocationS3 | IAMUser`

Required: No

GE

The greater than or equal to operator. The `DataSource` results will have `FilterVariable` values that are greater than or equal to the value specified with `GE`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

GT

The greater than operator. The `DataSource` results will have `FilterVariable` values that are greater than the value specified with `GT`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

LE

The less than or equal to operator. The `DataSource` results will have `FilterVariable` values that are less than or equal to the value specified with `LE`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

Limit

The maximum number of `DataSource` to include in the result.

Type: Number

Valid range: Minimum value of 1. Maximum value of 100.

Required: No

LT

The less than operator. The `DataSource` results will have `FilterVariable` values that are less than the value specified with `LT`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

NE

The not equal to operator. The `DataSource` results will have `FilterVariable` values not equal to the value specified with `NE`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

NextToken

The ID of the page in the paginated results.

Type: String

Required: No

Prefix

A string that is found at the beginning of a variable, such as `Name` or `Id`.

For example, a `DataSource` could have the `Name` `2014-09-09-HolidayGiftMailer`. To search for this `DataSource`, select `Name` for the `FilterVariable` and any of the following strings for the `Prefix`:

- 2014-09
- 2014-09-09
- 2014-09-09-Holiday

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

SortOrder

A two-value parameter that determines the sequence of the resulting list of `DataSource`.

- `asc` - Arranges the list in ascending order (A-Z, 0-9).
- `dsc` - Arranges the list in descending order (Z-A, 9-0).

Results are sorted by `FilterVariable`.

Type: String

Valid Values: `asc | dsc`

Required: No

Response Syntax

```
{
    "NextToken": "string",
    "Results": [
        {
            "ComputeStatistics": boolean,
            "CreatedAt": number,
            "CreatedByIamUser": "string",
            "DataLocationS3": "string",
            "DataRearrangement": "string",
            "DataSizeInBytes": number,
            "DataSourceId": "string",
            "LastUpdatedAt": number,
            "Message": "string",
            "Name": "string",
            "NumberOfFiles": number,
            "RDSMetadata": {
```

```
                "DataPipelineId": "string",
                "Database": {
                    "DatabaseName": "string",
                    "InstanceIdentifier": "string"
                },
                "DatabaseUserName": "string",
                "ResourceRole": "string",
                "SelectSqlQuery": "string",
                "ServiceRole": "string"
            },
            "RedshiftMetadata": {
                "DatabaseUserName": "string",
                "RedshiftDatabase": {
                    "ClusterIdentifier": "string",
                    "DatabaseName": "string"
                },
                "SelectSqlQuery": "string"
            },
            "RoleARN": "string",
            "Status": "string"
        }
    ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken
An ID of the next page in the paginated results that indicates at least one more page follows.

Type: String

Results
A list of `DataSource` that meet the search criteria.

Type: array of DataSource (p. 113) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

Examples

The following is a sample request and response of the DescribeDataSources operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.DescribeDataSources
{
  "FilterVariable": "Name",
  "Prefix": "bp-",
  "SortOrder": "asc",
  "Limit": 1
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "NextToken": "{\"DataSourceId\":\"ds-exampleDataSource2\"}",
  "Results": [
    {
      "ComputeStatistics": true,
      "CreatedAt": 1428008286.077,
      "CreatedByIamUser": "arn:aws:iam::<awsAccountId>:user/username",
      "DataLocationS3": "s3://bucket/locationToInput/example-
data.testing.csv",
      "DataSourceId": "ds-exampleDatasourceId",
      "LastUpdatedAt": 1428008286.077,
      "Name": "exampleDatasource",
      "Status": "PENDING"
    }
  ]
}
```

DescribeEvaluations

Returns a list of `DescribeEvaluations` that match the search criteria in the request.

Request Syntax

```
{
    "EQ": "string",
    "FilterVariable": "string",
    "GE": "string",
    "GT": "string",
    "LE": "string",
    "Limit": number,
    "LT": "string",
    "NE": "string",
    "NextToken": "string",
    "Prefix": "string",
    "SortOrder": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

EQ

The equal to operator. The `Evaluation` results will have `FilterVariable` values that exactly match the value specified with `EQ`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

FilterVariable

Use one of the following variable to filter a list of `Evaluation` objects:

- `CreatedAt` - Sets the search criteria to the `Evaluation` creation date.
- `Status` - Sets the search criteria to the `Evaluation` status.
- `Name` - Sets the search criteria to the contents of `Evaluation` `Name`.
- `IAMUser` - Sets the search criteria to the user account that invoked an `Evaluation`.
- `MLModelId` - Sets the search criteria to the `MLModel` that was evaluated.
- `DataSourceId` - Sets the search criteria to the `DataSource` used in `Evaluation`.
- `DataUri` - Sets the search criteria to the data file(s) used in `Evaluation`. The URL can identify either a file or an Amazon Simple Storage Solution (Amazon S3) bucket or directory.

Type: String

Valid Values: `CreatedAt | LastUpdatedAt | Status | Name | IAMUser | MLModelId | DataSourceId | DataURI`

Required: No

GE

The greater than or equal to operator. The `Evaluation` results will have `FilterVariable` values that are greater than or equal to the value specified with `GE`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

GT

The greater than operator. The `Evaluation` results will have `FilterVariable` values that are greater than the value specified with `GT`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

LE

The less than or equal to operator. The `Evaluation` results will have `FilterVariable` values that are less than or equal to the value specified with `LE`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

Limit

The maximum number of `Evaluation` to include in the result.

Type: Number

Valid range: Minimum value of 1. Maximum value of 100.

Required: No

LT

The less than operator. The `Evaluation` results will have `FilterVariable` values that are less than the value specified with `LT`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

NE

The not equal to operator. The `Evaluation` results will have `FilterVariable` values not equal to the value specified with `NE`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

NextToken

The ID of the page in the paginated results.

Type: String

Required: No

Prefix

A string that is found at the beginning of a variable, such as `Name` or `Id`.

For example, an `Evaluation` could have the `Name` `2014-09-09-HolidayGiftMailer`. To search for this `Evaluation`, select `Name` for the `FilterVariable` and any of the following strings for the `Prefix`:

- 2014-09
- 2014-09-09
- 2014-09-09-Holiday

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

SortOrder

A two-value parameter that determines the sequence of the resulting list of `Evaluation`.

- `asc` - Arranges the list in ascending order (A-Z, 0-9).
- `dsc` - Arranges the list in descending order (Z-A, 9-0).

Results are sorted by `FilterVariable`.

Type: String

Valid Values: `asc | dsc`

Required: No

Response Syntax

```
{
    "NextToken": "string",
    "Results": [
        {
            "CreatedAt": number,
            "CreatedByIamUser": "string",
            "EvaluationDataSourceId": "string",
            "EvaluationId": "string",
            "InputDataLocationS3": "string",
            "LastUpdatedAt": number,
            "MLModelId": "string",
            "Message": "string",
            "Name": "string",
```

```
            "PerformanceMetrics": {
                "Properties":
                    {
                        "string" :
                            "string"
                    }
            },
            "Status": "string"
        }
    ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken
The ID of the next page in the paginated results that indicates at least one more page follows.

Type: String

Results
A list of `Evaluation` that meet the search criteria.

Type: array of Evaluation (p. 116) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

Examples

The following is a sample request and response of the DescribeEvaluations operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
```

```
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.DescribeEvaluations
{
  "FilterVariable": "Name",
  "Prefix": "ev-",
  "SortOrder": "asc",
  "Limit": 1
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "NextToken": "{\"EvaluationId\":\"ev-exampleId2\"}",
  "Results": [
    {
      "CreatedAt": 1420745248.785,
      "CreatedByIamUser": "arn:aws:iam::<awsAccountId>:user/username",
      "EvaluationDataSourceId": "ds-exampleDataSourceId",
      "EvaluationId": "ev-exampleId1",
      "InputDataLocationS3": "s3://bucket/locationToInput/example-
data.testing.csv",
      "LastUpdatedAt": 1420745524.506,
      "MLModelId": "pr-exampleModelId",
      "Name": "ev-1",
      "PerformanceMetrics":
       {
         "Properties": {"BinaryAUC": "0.9228827246570067"}
       },
      "Status": "COMPLETED"
    }
  ]
}
```

DescribeMLModels

Returns a list of `MLModel` that match the search criteria in the request.

Request Syntax

```
{
    "EQ": "string",
    "FilterVariable": "string",
    "GE": "string",
    "GT": "string",
    "LE": "string",
    "Limit": number,
    "LT": "string",
    "NE": "string",
    "NextToken": "string",
    "Prefix": "string",
    "SortOrder": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

EQ

The equal to operator. The `MLModel` results will have `FilterVariable` values that exactly match the value specified with `EQ`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

FilterVariable

Use one of the following variables to filter a list of `MLModel`:

- `CreatedAt` - Sets the search criteria to `MLModel` creation date.
- `Status` - Sets the search criteria to `MLModel` status.
- `Name` - Sets the search criteria to the contents of `MLModel` `Name`.
- `IAMUser` - Sets the search criteria to the user account that invoked the `MLModel` creation.
- `TrainingDataSourceId` - Sets the search criteria to the `DataSource` used to train one or more `MLModel`.
- `RealtimeEndpointStatus` - Sets the search criteria to the `MLModel` real-time endpoint status.
- `MLModelType` - Sets the search criteria to `MLModel` type: binary, regression, or multi-class.
- `Algorithm` - Sets the search criteria to the algorithm that the `MLModel` uses.
- `TrainingDataURI` - Sets the search criteria to the data file(s) used in training a `MLModel`. The URL can identify either a file or an Amazon Simple Storage Service (Amazon S3) bucket or directory.

Type: String

Valid Values: `CreatedAt | LastUpdatedAt | Status | Name | IAMUser | TrainingDataSourceId | RealtimeEndpointStatus | MLModelType | Algorithm | TrainingDataURI`

Required: No

GE

The greater than or equal to operator. The `MLModel` results will have `FilterVariable` values that are greater than or equal to the value specified with `GE`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

GT

The greater than operator. The `MLModel` results will have `FilterVariable` values that are greater than the value specified with `GT`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

LE

The less than or equal to operator. The `MLModel` results will have `FilterVariable` values that are less than or equal to the value specified with `LE`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

Limit

The number of pages of information to include in the result. The range of acceptable values is `1` through `100`. The default value is `100`.

Type: Number

Valid range: Minimum value of 1. Maximum value of 100.

Required: No

LT

The less than operator. The `MLModel` results will have `FilterVariable` values that are less than the value specified with `LT`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

NE

The not equal to operator. The `MLModel` results will have `FilterVariable` values not equal to the value specified with `NE`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

NextToken

The ID of the page in the paginated results.

Type: String

Required: No

Prefix

A string that is found at the beginning of a variable, such as `Name` or `Id`.

For example, an `MLModel` could have the `Name` `2014-09-09-HolidayGiftMailer`. To search for this `MLModel`, select `Name` for the `FilterVariable` and any of the following strings for the `Prefix`:

- 2014-09
- 2014-09-09
- 2014-09-09-Holiday

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

SortOrder

A two-value parameter that determines the sequence of the resulting list of `MLModel`.

- `asc` - Arranges the list in ascending order (A-Z, 0-9).
- `dsc` - Arranges the list in descending order (Z-A, 9-0).

Results are sorted by `FilterVariable`.

Type: String

Valid Values: `asc | dsc`

Required: No

Response Syntax

```
{
    "NextToken": "string",
    "Results": [
        {
```

```
            "Algorithm": "string",
            "CreatedAt": number,
            "CreatedByIamUser": "string",
            "EndpointInfo": {
                "CreatedAt": number,
                "EndpointStatus": "string",
                "EndpointUrl": "string",
                "PeakRequestsPerSecond": number
            },
            "InputDataLocationS3": "string",
            "LastUpdatedAt": number,
            "MLModelId": "string",
            "MLModelType": "string",
            "Message": "string",
            "Name": "string",
            "ScoreThreshold": number,
            "ScoreThresholdLastUpdatedAt": number,
            "SizeInBytes": number,
            "Status": "string",
            "TrainingDataSourceId": "string",
            "TrainingParameters":
                {
                    "string" :
                        "string"
                }
        }
    ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken
The ID of the next page in the paginated results that indicates at least one more page follows.

Type: String

Results
A list of `MLModel` that meet the search criteria.

Type: array of MLModel (p. 118) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

Examples

The following is a sample request and response of the DescribeMLModels operation:

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.DescribeMLModels
{
  "FilterVariable": "Name",
  "Prefix": "ml-",
  "SortOrder": "asc",
  "Limit": 1
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "NextToken": "\"PredictorId\":\"Spr-ml-model-testing\"}",
  "Results": [
    {
      "CreatedAt": 1422475435.595,
      "CreatedByIamUser": "arn:aws:iam::<awsAccountId>:user/username",
      "InputDataLocationS3": "s3://bucket/locationToInput/example-
data.testing.csv",
      "LastUpdatedAt": 1422475709.691,
      "MLModelId": "ml-model-testing",
      "MLModelType": "MULTICLASS",
      "EndpointInfo": {
        "CreatedAt": 1424378682.266,
        "EndpointStatus": "READY",
        "EndpointUrl": "<realtime endpoint from Amazon Machine Learning for
 ml-model-testing>",
        "PeakRequestsPerSecond": 200}
      "Name": "ml-model-name",
      "SizeInBytes": 352720,
```

```
      "Status": "COMPLETED",
      "TrainingDataSourceId": "exampleDataSourceId",
      "TrainingParameters":
       {
         "algorithm": "sgd",
         "sgd.l1RegularizationAmount": "0.0",
         "sgd.l2RegularizationAmount": "1E-6",
         "sgd.maxMLModelSizeInBytes": "33554432",
         "sgd.maxPasses": "10"
      }
    }
  ]
}
```

DescribeTags

Describes one or more of the tags for your Amazon ML object.

Request Syntax

```
{
    "ResourceId": "string",
    "ResourceType": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

ResourceId
The ID of the ML object. For example, `exampleModelId`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

ResourceType
The type of the ML object.

Type: String

Valid Values: `BatchPrediction | DataSource | Evaluation | MLModel`

Required: Yes

Response Syntax

```
{
    "ResourceId": "string",
    "ResourceType": "string",
    "Tags": [
        {
            "Key": "string",
            "Value": "string"
        }
    ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ResourceId
The ID of the tagged ML object.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

ResourceType
The type of the tagged ML object.

Type: String

Valid Values: `BatchPrediction | DataSource | Evaluation | MLModel`

Tags
A list of tags associated with the ML object.

Type: array of Tag (p. 136) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

ResourceNotFoundException
A specified resource cannot be located.

HTTP Status Code: 400

Examples

The following are an example request and response for the DescribeTags operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
```

```
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.DescribeTags
{
  "ResourceId": "exampleModelId",
  "ResourceType": "MLModel"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "ResourceId": "exampleModelId",
  "ResourceType": "MLModel",
  "Tags": {
      "Key":"exampleKey",
   "Value":"exampleKeyValue"
  }
}
```

GetBatchPrediction

Returns a `BatchPrediction` that includes detailed metadata, status, and data file information for a `Batch Prediction` request.

Request Syntax

```
{
    "BatchPredictionId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

BatchPredictionId
An ID assigned to the `BatchPrediction` at creation.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

Response Syntax

```
{
    "BatchPredictionDataSourceId": "string",
    "BatchPredictionId": "string",
    "CreatedAt": number,
    "CreatedByIamUser": "string",
    "InputDataLocationS3": "string",
    "LastUpdatedAt": number,
    "LogUri": "string",
    "Message": "string",
    "MLModelId": "string",
    "Name": "string",
    "OutputUri": "string",
    "Status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

BatchPredictionDataSourceId
The ID of the `DataSource` that was used to create the `BatchPrediction`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

BatchPredictionId
An ID assigned to the `BatchPrediction` at creation. This value should be identical to the value of the `BatchPredictionID` in the request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

CreatedAt
The time when the `BatchPrediction` was created. The time is expressed in epoch time.

Type: DateTime

CreatedByIamUser
The AWS user account that invoked the `BatchPrediction`. The account type can be either an AWS root account or an AWS Identity and Access Management (IAM) user account.

Type: String

Pattern: `arn:aws:iam::[0-9]+:((user/.+)|(root))`

InputDataLocationS3
The location of the data file or directory in Amazon Simple Storage Service (Amazon S3).

Type: String

Length constraints: Minimum length of 0. Maximum length of 2048.

Pattern: `s3://([^/]+)(/.*)?`

LastUpdatedAt
The time of the most recent edit to `BatchPrediction`. The time is expressed in epoch time.

Type: DateTime

LogUri
A link to the file that contains logs of the `CreateBatchPrediction` operation.

Type: String

Message
A description of the most recent details about processing the batch prediction request.

Type: String

Length constraints: Minimum length of 0. Maximum length of 10240.

MLModelId
The ID of the `MLModel` that generated predictions for the `BatchPrediction` request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Name
A user-supplied name or description of the `BatchPrediction`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

OutputUri
The location of an Amazon S3 bucket or directory to receive the operation results.

Type: String

Length constraints: Minimum length of 0. Maximum length of 2048.

Pattern: `s3://([^/]+)(/.*)?`

Status
The status of the `BatchPrediction`, which can be one of the following values:

- `PENDING` - Amazon Machine Learning (Amazon ML) submitted a request to generate batch predictions.
- `INPROGRESS` - The batch predictions are in progress.
- `FAILED` - The request to perform a batch prediction did not run to completion. It is not usable.
- `COMPLETED` - The batch prediction process completed successfully.
- `DELETED` - The `BatchPrediction` is marked as deleted. It is not usable.

Type: String

Valid Values: `PENDING | INPROGRESS | FAILED | COMPLETED | DELETED`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

ResourceNotFoundException
A specified resource cannot be located.

HTTP Status Code: 400

Examples

The following is a sample request and response of the GetBatchPrediction operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.GetBatchPrediction
{"BatchPredictionId": "EXAMPLE-bp-2014-09-12-15-14-04-156"}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "BatchPredictionDataSourceId":"EXAMPLE-tr-ds-2014-09-12-15-14-04-989",
  "BatchPredictionId":"EXAMPLE-bp-2014-09-12-15-14-04-156",
  "CreatedAt":1.410560632327E9,
  "CreatedByIamUser":"arn:aws:iam::<awsAccountId>:user/user",
  "InputDataLocationS3": "s3://eml-test-EXAMPLE/example.csv",
  "LastUpdatedAt":1.410560632327E9,
  "LogUri": "https://s3bucket/locationToLogs/logname.tar.gz",
  "Name":"EXAMPLE",
  "OutputUri":"s3://eml-test-EXAMPLE/test-outputs/EXAMPLE-
bp-2014-09-12-15-14-04-156/results",
  "MLModelId":"EXAMPLE-pr-2014-09-12-15-14-04-924",
  "Status":"COMPLETED"
}
```

GetDataSource

Returns a `DataSource` that includes metadata and data file information, as well as the current status of the `DataSource`.

`GetDataSource` provides results in normal or verbose format. The verbose format adds the schema description and the list of files pointed to by the DataSource to the normal format.

Request Syntax

```
{
    "DataSourceId": "string",
    "Verbose": boolean
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

DataSourceId
The ID assigned to the `DataSource` at creation.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

Verbose
Specifies whether the `GetDataSource` operation should return `DataSourceSchema`.

If true, `DataSourceSchema` is returned.

If false, `DataSourceSchema` is not returned.

Type: Boolean

Required: No

Response Syntax

```
{
    "ComputeStatistics": boolean,
    "CreatedAt": number,
    "CreatedByIamUser": "string",
    "DataLocationS3": "string",
    "DataRearrangement": "string",
```

```
    "DataSizeInBytes": number,
    "DataSourceId": "string",
    "DataSourceSchema": "string",
    "LastUpdatedAt": number,
    "LogUri": "string",
    "Message": "string",
    "Name": "string",
    "NumberOfFiles": number,
    "RDSMetadata": {
        "DataPipelineId": "string",
        "Database": {
            "DatabaseName": "string",
            "InstanceIdentifier": "string"
        },
        "DatabaseUserName": "string",
        "ResourceRole": "string",
        "SelectSqlQuery": "string",
        "ServiceRole": "string"
    },
    "RedshiftMetadata": {
        "DatabaseUserName": "string",
        "RedshiftDatabase": {
            "ClusterIdentifier": "string",
            "DatabaseName": "string"
        },
        "SelectSqlQuery": "string"
    },
    "RoleARN": "string",
    "Status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ComputeStatistics
The parameter is `true` if statistics need to be generated from the observation data.

Type: Boolean

CreatedAt
The time that the `DataSource` was created. The time is expressed in epoch time.

Type: DateTime

CreatedByIamUser
The AWS user account from which the `DataSource` was created. The account type can be either an AWS root account or an AWS Identity and Access Management (IAM) user account.

Type: String

Pattern: `arn:aws:iam::[0-9]+:((user/.+)|(root))`

DataLocationS3
The location of the data file or directory in Amazon Simple Storage Service (Amazon S3).

Type: String

Length constraints: Minimum length of 0. Maximum length of 2048.

Pattern: `s3://([^/]+)(/.*)?`

DataRearrangement
A JSON string that represents the splitting and rearrangement requirement used when this `DataSource` was created.

Type: String

DataSizeInBytes
The total size of observations in the data files.

Type: Long

DataSourceId
The ID assigned to the `DataSource` at creation. This value should be identical to the value of the `DataSourceId` in the request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

DataSourceSchema
The schema used by all of the data files of this `DataSource`.

> **Note**
> This parameter is provided as part of the verbose format.

Type: String

Length constraints: Minimum length of 0. Maximum length of 131071.

LastUpdatedAt
The time of the most recent edit to the `DataSource`. The time is expressed in epoch time.

Type: DateTime

LogUri
A link to the file containing logs of `CreateDataSourceFrom*` operations.

Type: String

Message
The user-supplied description of the most recent details about creating the `DataSource`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 10240.

Name
A user-supplied name or description of the `DataSource`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

NumberOfFiles
The number of data files referenced by the `DataSource`.

Type: Long

RDSMetadata
The datasource details that are specific to Amazon RDS.

Type: RDSMetadata (p. 127) object

RedshiftMetadata
Describes the `DataSource` details specific to Amazon Redshift.

Type: RedshiftMetadata (p. 133) object

RoleARN
The Amazon Resource Name (ARN) of an AWS IAM Role, such as the following: arn:aws:iam::account:role/rolename.

Type: String

Length constraints: Minimum length of 1. Maximum length of 110.

Status
The current status of the `DataSource`. This element can have one of the following values:

- `PENDING` - Amazon ML submitted a request to create a `DataSource`.
- `INPROGRESS` - The creation process is underway.
- `FAILED` - The request to create a `DataSource` did not run to completion. It is not usable.
- `COMPLETED` - The creation process completed successfully.
- `DELETED` - The `DataSource` is marked as deleted. It is not usable.

Type: String

Valid Values: `PENDING | INPROGRESS | FAILED | COMPLETED | DELETED`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

ResourceNotFoundException
A specified resource cannot be located.

HTTP Status Code: 400

Examples

The following is a sample request and response of the GetDataSource operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
```

```
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.GetDataSource
{"DataSourceId": "17SdAv6WC6r5vACAxF7U", "Verbose": true}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  {
    "CreatedAt":1.41045168275E9,
    "CreatedByIamUser":"arn:aws:iam::<awsAccountId>:user/testuser",
    "DataLocationS3":"s3://eml-test-EXAMPLE /data.csv",
    "DataRearrangement": "{\"splitting\":{\"percentBegin\":10,\"percentEnd
\":60}}",
    "DataSizeInBytes":0,"DataSourceId":"17SdAv6WC6r5vACAxF7U",
    "DataSourceSchema":"
    {
      \"version\":\"1.0\",
      \"recordAnnotationFieldName\":null,
      \"recordWeightFieldName\":\"weight\",
      \"targetFieldName\":\"label\",
      \"dataFormat\":\"CSV\",
      \"dataFileContainsHeader\":false,
      \"attributes\":
      [
        {\"fieldName\":\"obsId\",\"fieldType\":\"NUMERIC\"},
        {\"fieldName\":\"label\",\"fieldType\":\"BINARY\"},
        {\"fieldName\":\"weight\",\"fieldType\":\"NUMERIC\"},
        {\"fieldName\":\"x\",\"fieldType\":\"TEXT\"}
      ],
      \"excludedVariableNames\":[]
    }",
    "DataStatisticsStatus":"COMPLETED",
    "LastUpdatedAt":1.41045168275E9,
    "LogUri": "https://s3bucket/locationToLogs/logname.tar.gz",
    "Name":"EXAMPLE",
    "Status":"COMPLETED"
  }
}
```

GetEvaluation

Returns an `Evaluation` that includes metadata as well as the current status of the `Evaluation`.

Request Syntax

```
{
    "EvaluationId": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

EvaluationId

The ID of the `Evaluation` to retrieve. The evaluation of each `MLModel` is recorded and cataloged. The ID provides the means to access the information.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

Response Syntax

```
{
    "CreatedAt": number,
    "CreatedByIamUser": "string",
    "EvaluationDataSourceId": "string",
    "EvaluationId": "string",
    "InputDataLocationS3": "string",
    "LastUpdatedAt": number,
    "LogUri": "string",
    "Message": "string",
    "MLModelId": "string",
    "Name": "string",
    "PerformanceMetrics": {
        "Properties":
            {
                "string" :
                    "string"
            }
    },
    "Status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

CreatedAt
The time that the `Evaluation` was created. The time is expressed in epoch time.

Type: DateTime

CreatedByIamUser
The AWS user account that invoked the evaluation. The account type can be either an AWS root account or an AWS Identity and Access Management (IAM) user account.

Type: String

Pattern: `arn:aws:iam::[0-9]+:((user/.+)|(root))`

EvaluationDataSourceId
The `DataSource` used for this evaluation.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

EvaluationId
The evaluation ID which is same as the `EvaluationId` in the request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

InputDataLocationS3
The location of the data file or directory in Amazon Simple Storage Service (Amazon S3).

Type: String

Length constraints: Minimum length of 0. Maximum length of 2048.

Pattern: `s3://([^/]+)(/.*)?`

LastUpdatedAt
The time of the most recent edit to the `BatchPrediction`. The time is expressed in epoch time.

Type: DateTime

LogUri
A link to the file that contains logs of the `CreateEvaluation` operation.

Type: String

Message
A description of the most recent details about evaluating the `MLModel`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 10240.

MLModelId
The ID of the `MLModel` that was the focus of the evaluation.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Name

A user-supplied name or description of the `Evaluation`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

PerformanceMetrics

Measurements of how well the `MLModel` performed using observations referenced by the `DataSource`. One of the following metric is returned based on the type of the `MLModel`:

- BinaryAUC: A binary `MLModel` uses the Area Under the Curve (AUC) technique to measure performance.
- RegressionRMSE: A regression `MLModel` uses the Root Mean Square Error (RMSE) technique to measure performance. RMSE measures the difference between predicted and actual values for a single variable.
- MulticlassAvgFScore: A multiclass `MLModel` uses the F1 score technique to measure performance.

For more information about performance metrics, please see the Amazon Machine Learning Developer Guide.

Type: PerformanceMetrics (p. 121) object

Status

The status of the evaluation. This element can have one of the following values:

- `PENDING` - Amazon Machine Language (Amazon ML) submitted a request to evaluate an `MLModel`.
- `INPROGRESS` - The evaluation is underway.
- `FAILED` - The request to evaluate an `MLModel` did not run to completion. It is not usable.
- `COMPLETED` - The evaluation process completed successfully.
- `DELETED` - The `Evaluation` is marked as deleted. It is not usable.

Type: String

Valid Values: `PENDING | INPROGRESS | FAILED | COMPLETED | DELETED`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException

An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException

An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

ResourceNotFoundException

A specified resource cannot be located.

HTTP Status Code: 400

Examples

The following is a sample request and response of the GetEvaluation operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.GetEvaluation
{"EvaluationId": "ev-2014-09-12-15-14-04-924"}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "CreatedAt":1.410560805669E9,
  "CreatedByIamUser":"arn:aws:iam::<awsAccountId>:user/user",
  "EvaluationDataSourceId":"EXAMPLE-ev-ds-2014-09-12-15-14-04-411",
  "EvaluationId":"ev-2014-09-12-15-14-04-924",
  "InputDataLocationS3": "s3://eml-test-EXAMPLE/example.csv",
  "LastUpdatedAt":1.410560805669E9,
  "LogUri": "https://s3bucket/locationToLogs/logname.tar.gz",
  "Name":"EXAMPLE",
  "PerformanceMetrics":{"Properties":{}},
  "MLModelId":"EXAMPLE-pr-2014-09-12-15-14-04-924",
  "Status":"COMPLETED"
}
```

GetMLModel

Returns an `MLModel` that includes detailed metadata, data source information, and the current status of the `MLModel`.

`GetMLModel` provides results in normal or verbose format.

Request Syntax

```
{
    "MLModelId": "string",
    "Verbose": boolean
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

MLModelId

The ID assigned to the `MLModel` at creation.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

Verbose

Specifies whether the `GetMLModel` operation should return `Recipe`.

If true, `Recipe` is returned.

If false, `Recipe` is not returned.

Type: Boolean

Required: No

Response Syntax

```
{
    "CreatedAt": number,
    "CreatedByIamUser": "string",
    "EndpointInfo": {
        "CreatedAt": number,
        "EndpointStatus": "string",
        "EndpointUrl": "string",
        "PeakRequestsPerSecond": number
    },
```

```
    "InputDataLocationS3": "string",
    "LastUpdatedAt": number,
    "LogUri": "string",
    "Message": "string",
    "MLModelId": "string",
    "MLModelType": "string",
    "Name": "string",
    "Recipe": "string",
    "Schema": "string",
    "ScoreThreshold": number,
    "ScoreThresholdLastUpdatedAt": number,
    "SizeInBytes": number,
    "Status": "string",
    "TrainingDataSourceId": "string",
    "TrainingParameters":
        {
            "string" :
                "string"
        }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

CreatedAt
The time that the `MLModel` was created. The time is expressed in epoch time.

Type: DateTime

CreatedByIamUser
The AWS user account from which the `MLModel` was created. The account type can be either an AWS root account or an AWS Identity and Access Management (IAM) user account.

Type: String

Pattern: `arn:aws:iam::[0-9]+:((user/.+)|(root))`

EndpointInfo
The current endpoint of the `MLModel`

Type: RealtimeEndpointInfo (p. 128) object

InputDataLocationS3
The location of the data file or directory in Amazon Simple Storage Service (Amazon S3).

Type: String

Length constraints: Minimum length of 0. Maximum length of 2048.

Pattern: `s3://([^/]+)(/.*)?`

LastUpdatedAt
The time of the most recent edit to the `MLModel`. The time is expressed in epoch time.

Type: DateTime

LogUri
A link to the file that contains logs of the `CreateMLModel` operation.

Type: String

Message
A description of the most recent details about accessing the `MLModel`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 10240.

MLModelId
The MLModel ID, which is same as the `MLModelId` in the request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

MLModelType
Identifies the `MLModel` category. The following are the available types:

- REGRESSION -- Produces a numeric result. For example, "What price should a house be listed at?"
- BINARY -- Produces one of two possible results. For example, "Is this an e-commerce website?"
- MULTICLASS -- Produces one of several possible results. For example, "Is this a HIGH, LOW or MEDIUM risk trade?"

Type: String

Valid Values: `REGRESSION | BINARY | MULTICLASS`

Name
A user-supplied name or description of the `MLModel`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Recipe
The recipe to use when training the `MLModel`. The `Recipe` provides detailed information about the observation data to use during training, and manipulations to perform on the observation data during training.

> **Note**
> This parameter is provided as part of the verbose format.

Type: String

Length constraints: Minimum length of 0. Maximum length of 131071.

Schema
The schema used by all of the data files referenced by the `DataSource`.

> **Note**
> This parameter is provided as part of the verbose format.

Type: String

Length constraints: Minimum length of 0. Maximum length of 131071.

ScoreThreshold
The scoring threshold is used in binary classification `MLModel` models. It marks the boundary between a positive prediction and a negative prediction.

Output values greater than or equal to the threshold receive a positive result from the MLModel, such as `true`. Output values less than the threshold receive a negative response from the MLModel, such as `false`.

Type: Float

ScoreThresholdLastUpdatedAt

The time of the most recent edit to the `ScoreThreshold`. The time is expressed in epoch time.

Type: DateTime

SizeInBytes

Long integer type that is a 64-bit signed number.

Type: Long

Status

The current status of the `MLModel`. This element can have one of the following values:

- `PENDING` - Amazon Machine Learning (Amazon ML) submitted a request to describe a `MLModel`.
- `INPROGRESS` - The request is processing.
- `FAILED` - The request did not run to completion. The ML model isn't usable.
- `COMPLETED` - The request completed successfully.
- `DELETED` - The `MLModel` is marked as deleted. It isn't usable.

Type: String

Valid Values: `PENDING | INPROGRESS | FAILED | COMPLETED | DELETED`

TrainingDataSourceId

The ID of the training `DataSource`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

TrainingParameters

A list of the training parameters in the `MLModel`. The list is implemented as a map of key-value pairs.

The following is the current set of training parameters:

- `sgd.maxMLModelSizeInBytes` - The maximum allowed size of the model. Depending on the input data, the size of the model might affect its performance.

  The value is an integer that ranges from `100000` to `2147483648`. The default value is `33554432`.
- `sgd.maxPasses` - The number of times that the training process traverses the observations to build the `MLModel`. The value is an integer that ranges from `1` to `10000`. The default value is `10`.
- `sgd.shuffleType` - Whether Amazon ML shuffles the training data. Shuffling data improves a model's ability to find the optimal solution for a variety of data types. The valid values are `auto` and `none`. The default value is `none`. We strongly recommend that you shuffle your data.
- `sgd.l1RegularizationAmount` - The coefficient regularization `L1` norm. It controls overfitting the data by penalizing large coefficients. This tends to drive coefficients to zero, resulting in a sparse feature set. If you use this parameter, start by specifying a small value, such as `1.0E-08`.

  The value is a double that ranges from `0` to `MAX_DOUBLE`. The default is to not use `L1` normalization. This parameter can't be used when `L2` is specified. Use this parameter sparingly.
- `sgd.l2RegularizationAmount` - The coefficient regularization `L2` norm. It controls overfitting the data by penalizing large coefficients. This tends to drive coefficients to small, nonzero values. If you use this parameter, start by specifying a small value, such as `1.0E-08`.

  The value is a double that ranges from `0` to `MAX_DOUBLE`. The default is to not use `L2` normalization. This parameter can't be used when `L1` is specified. Use this parameter sparingly.

Type: String to String map

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

ResourceNotFoundException
A specified resource cannot be located.

HTTP Status Code: 400

Examples

The following is a sample request and response of the GetMLModel operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.GetMLModel
{"MLModelId": "EXAMPLE-pr-2014-09-12-15-14-04-924", "Verbose": true}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "CreatedAt":1.410560408264E9,
```

```
  "CreatedByIamUser":"arn:aws:iam::<awsAccountId>:user/user",
  "HasCalibration":false,
  "LastUpdatedAt":1.410560416338E9,
  "LogUri": "https://s3bucket/locationToLogs/logname.tar.gz",
  "Name":"Name-ml-model",
  "MLModelId":"ml-model",
  "EndpointInfo": {
    "CreatedAt": 1424378682.266,
   "EndpointStatus": "READY",
   "EndpointUrl": "<realtime endpoint from Amazon Machine Learning for ml-
model>",
   "PeakRequestsPerSecond": 200}
  "MLModelType":"BINARY",
  "Recipe":"{
    \"groups\": {},
    \n\"assignments\": {},
    \n\"dependencies\": {},
    \n\"outputs\": [\n    \"'x'\"\n]
  }\n",
  "Schema": "{
  \"version\":\"1.0\",
  \"rowId\":null,
  \"rowWeight\":null,
  \"targetAttributeName\":\"y\",
  \"dataFormat\":\"CSV\",
  \"dataFileContainsHeader\":false,
  \"attributes\":[{\"attributeName\":\"age\",
  \"attributeType\":\"NUMERIC\"},
  {\"attributeName\":\"job\",
  \"attributeType\":\"CATEGORICAL\"},
  {\"attributeName\":\"contact\",
  \"attributeType\":\"CATEGORICAL\"},
  {\"attributeName\":\"month\",
  \"attributeType\":\"CATEGORICAL\"},
  {\"attributeName\":\"day_of_week\",
  \"attributeType\":\"CATEGORICAL\"},
  {\"attributeName\":\"duration\",
  \"attributeType\":\"NUMERIC\"},
  {\"attributeName\":\"poutcome\",
  \"attributeType\":\"CATEGORICAL\"},
  {\"attributeName\":\"nr_employed\",
  \"attributeType\":\"NUMERIC\"},
  {\"attributeName\":\"y\",
  \"attributeType\":\"BINARY\"}],
  \"excludedAttributeNames\":[]}"
  "SizeInBytes": 400374,
  "Status":"COMPLETED",
  "TrainingDataSourceId":"EXAMPLE-tr-ds-2014-09-12-15-14-04-989",
  "TrainingParameters":{
      "algorithm":"sgd",
      "sgd.l1":"0.0",
      "sgd.l2":"0.0",
      "sgd.likelihood":"logreg",
      "sgd.passes":"1"
  }
}
```

Predict

Generates a prediction for the observation using the specified `ML Model`.

Note
Not all response parameters will be populated. Whether a response parameter is populated depends on the type of model requested.

Request Syntax

```
{
    "MLModelId": "string",
    "PredictEndpoint": "string",
    "Record":
        {
            "string" :
                "string"
        }
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

MLModelId
A unique identifier of the `MLModel`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

PredictEndpoint
Type: String

Length constraints: Minimum length of 0. Maximum length of 2048.

Pattern: `https://[a-zA-Z0-9-.]*\.amazon(aws)?\.com[/]?`

Required: Yes

Record
A map of variable name-value pairs that represent an observation.

Type: String to String map

Required: Yes

Response Syntax

```
{
    "Prediction": {
        "Details":
            {
                "string" :
                    "string"
            },
        "PredictedLabel": "string",
        "PredictedScores":
            {
                "string" :
                    number
            },
        "PredictedValue": number
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Prediction
The output from a `Predict` operation:

- `Details` - Contains the following attributes: `DetailsAttributes.PREDICTIVE_MODEL_TYPE - REGRESSION | BINARY | MULTICLASS DetailsAttributes.ALGORITHM - SGD`
- `PredictedLabel` - Present for either a `BINARY` or `MULTICLASS` `MLModel` request.
- `PredictedScores` - Contains the raw classification score corresponding to each label.
- `PredictedValue` - Present for a `REGRESSION` `MLModel` request.

Type: Prediction (p. 121) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

LimitExceededException
The subscriber exceeded the maximum number of operations. This exception can occur when listing objects such as `DataSource`.

HTTP Status Code: 400

PredictorNotMountedException
The exception is thrown when a predict request is made to an unmounted `MLModel`.

HTTP Status Code: 400

ResourceNotFoundException
A specified resource cannot be located.

HTTP Status Code: 400

Examples

The following is a sample request and response of the Predict operation.

Sample Request

```
POST / HTTP/1.1
Host: <hostname from the GetMLModel response EndpointUrl object>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.Predict
{"MLModelId" : "exampleMLModelId",
 "Record" : {
   "ExampleData" : "exampleValue"
 },
 "PredictEndpoint" : "<realtime endpoint from Amazon Machine Learning for
 exampleMLModelId>"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{"PredictedLabel" : "0"
 "PredictedScores" : {
   "0" : "0.446588516"
 },
 "Details" : {
   "PredictiveModelType" : "BINARY",
   "Algorithm" : "SGD"
 }

}
```

UpdateBatchPrediction

Updates the `BatchPredictionName` of a `BatchPrediction`.

You can use the `GetBatchPrediction` operation to view the contents of the updated data element.

Request Syntax

```
{
    "BatchPredictionId": "string",
    "BatchPredictionName": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

BatchPredictionId
The ID assigned to the `BatchPrediction` during creation.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

BatchPredictionName
A new user-supplied name or description of the `BatchPrediction`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: Yes

Response Syntax

```
{
    "BatchPredictionId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

BatchPredictionId
The ID assigned to the `BatchPrediction` during creation. This value should be identical to the value of the `BatchPredictionId` in the request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

ResourceNotFoundException
A specified resource cannot be located.

HTTP Status Code: 400

Examples

The following is a sample request and response of the UpdateBatchPrediction operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.UpdateBatchPrediction
{
  "BatchPredictionId": "bp-exampleBatchPredictionId",
  "BatchPredictionName": "bp-exampleBatchPredictionName"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{"BatchPredictionId": "bp-exampleBatchPredictionId"}
```

UpdateDataSource

Updates the `DataSourceName` of a `DataSource`.

You can use the `GetDataSource` operation to view the contents of the updated data element.

Request Syntax

```
{
    "DataSourceId": "string",
    "DataSourceName": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

DataSourceId
The ID assigned to the `DataSource` during creation.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

DataSourceName
A new user-supplied name or description of the `DataSource` that will replace the current description.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: Yes

Response Syntax

```
{
    "DataSourceId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

DataSourceId
The ID assigned to the `DataSource` during creation. This value should be identical to the value of the `DataSourceID` in the request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

ResourceNotFoundException
A specified resource cannot be located.

HTTP Status Code: 400

Examples

The following is a sample request and response of the UpdateDataSource operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.UpdateDataSource
{
  "DataSourceId": "ds-exampleDataSourceId",
  "DataSourceName": "ds-exampleDataSourceName"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{"DataSourceId": "ds-exampleDataSourceId"}
```

UpdateEvaluation

Updates the `EvaluationName` of an `Evaluation`.

You can use the `GetEvaluation` operation to view the contents of the updated data element.

Request Syntax

```
{
    "EvaluationId": "string",
    "EvaluationName": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

EvaluationId
The ID assigned to the `Evaluation` during creation.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

EvaluationName
A new user-supplied name or description of the `Evaluation` that will replace the current content.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: Yes

Response Syntax

```
{
    "EvaluationId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

EvaluationId
The ID assigned to the `Evaluation` during creation. This value should be identical to the value of the `Evaluation` in the request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

ResourceNotFoundException
A specified resource cannot be located.

HTTP Status Code: 400

Examples

The following is a sample request and response of the UpdateEvaluation operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.UpdateEvaluation
{
  "EvaluationId": "ev-exampleEvaluationId",
  "EvaluationName": "ev-exampleEvaluationName"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{"EvaluationId": "ev-exampleEvaluationId"}
```

UpdateMLModel

Updates the `MLModelName` and the `ScoreThreshold` of an `MLModel`.

You can use the `GetMLModel` operation to view the contents of the updated data element.

Request Syntax

```
{
    "MLModelId": "string",
    "MLModelName": "string",
    "ScoreThreshold": number
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 137).

The request requires the following data in JSON format.

MLModelId
The ID assigned to the `MLModel` during creation.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

MLModelName
A user-supplied name or description of the `MLModel`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

ScoreThreshold
The `ScoreThreshold` used in binary classification `MLModel` that marks the boundary between a positive prediction and a negative prediction.

Output values greater than or equal to the `ScoreThreshold` receive a positive result from the `MLModel`, such as `true`. Output values less than the `ScoreThreshold` receive a negative response from the `MLModel`, such as `false`.

Type: Float

Required: No

Response Syntax

```
{
    "MLModelId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

MLModelId
The ID assigned to the `MLModel` during creation. This value should be identical to the value of the `MLModelID` in the request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 139).

InternalServerException
An error on the server occurred when trying to process a request.

HTTP Status Code: 500

InvalidInputException
An error on the client occurred. Typically, the cause is an invalid input value.

HTTP Status Code: 400

ResourceNotFoundException
A specified resource cannot be located.

HTTP Status Code: 400

Examples

The following is a sample request and response of the UpdateBatchPrediction operation.

Sample Request

```
POST / HTTP/1.1
Host: machinelearning.<region>.<domain>
x-amz-Date: <Date>
```

```
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=contenttype;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid,Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: AmazonML_20141212.UpdateMLModel
{
  "MLModelId": "ml-exampleModelId",
  "MLModelName": "ml-exampleModelName",
  "ScoreThreshold": 0.8
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{"MLModelId": "pr-exampleModelId"}
```

Data Types

The Amazon Machine Learning API contains several data types that various actions use. This section describes each data type in detail.

> **Note**
> The order of each element in the response is not guaranteed. Applications should not assume a particular order.

The following data types are supported:

BatchPrediction

Description

Represents the output of a `GetBatchPrediction` operation.

The content consists of the detailed metadata, the status, and the data file information of a `Batch Prediction`.

Contents

BatchPredictionDataSourceId
The ID of the `DataSource` that points to the group of observations to predict.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

BatchPredictionId
The ID assigned to the `BatchPrediction` at creation. This value should be identical to the value of the `BatchPredictionID` in the request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

CreatedAt
The time that the `BatchPrediction` was created. The time is expressed in epoch time.

Type: DateTime

Required: No

CreatedByIamUser
The AWS user account that invoked the `BatchPrediction`. The account type can be either an AWS root account or an AWS Identity and Access Management (IAM) user account.

Type: String

Pattern: `arn:aws:iam::[0-9]+:((user/.+)|(root))`

Required: No

InputDataLocationS3
The location of the data file or directory in Amazon Simple Storage Service (Amazon S3).

Type: String

Length constraints: Minimum length of 0. Maximum length of 2048.

Pattern: `s3://([^/]+)(/.*)?`

Required: No

LastUpdatedAt
The time of the most recent edit to the `BatchPrediction`. The time is expressed in epoch time.

Type: DateTime

Required: No

Message
A description of the most recent details about processing the batch prediction request.

Type: String

Length constraints: Minimum length of 0. Maximum length of 10240.

Required: No

MLModelId

The ID of the `MLModel` that generated predictions for the `BatchPrediction` request.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

Name

A user-supplied name or description of the `BatchPrediction`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

OutputUri

The location of an Amazon S3 bucket or directory to receive the operation results. The following substrings are not allowed in the `s3 key` portion of the `outputURI` field: ':', '//', '/./', '/../'.

Type: String

Length constraints: Minimum length of 0. Maximum length of 2048.

Pattern: `s3://([^/]+)(/.*)?`

Required: No

Status

The status of the `BatchPrediction`. This element can have one of the following values:

- `PENDING` - Amazon Machine Learning (Amazon ML) submitted a request to generate predictions for a batch of observations.
- `INPROGRESS` - The process is underway.
- `FAILED` - The request to perform a batch prediction did not run to completion. It is not usable.
- `COMPLETED` - The batch prediction process completed successfully.
- `DELETED` - The `BatchPrediction` is marked as deleted. It is not usable.

Type: String

Valid Values: `PENDING | INPROGRESS | FAILED | COMPLETED | DELETED`

Required: No

DataSource

Description

Represents the output of the `GetDataSource` operation.

The content consists of the detailed metadata and data file information and the current status of the `DataSource`.

Contents

ComputeStatistics
The parameter is `true` if statistics need to be generated from the observation data.

Type: Boolean

Required: No

CreatedAt
The time that the `DataSource` was created. The time is expressed in epoch time.

Type: DateTime

Required: No

CreatedByIamUser
The AWS user account from which the `DataSource` was created. The account type can be either an AWS root account or an AWS Identity and Access Management (IAM) user account.

Type: String

Pattern: `arn:aws:iam::[0-9]+:((user/.+)|(root))`

Required: No

DataLocationS3
The location and name of the data in Amazon Simple Storage Service (Amazon S3) that is used by a `DataSource`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 2048.

Pattern: `s3://([^/]+)(/.*)?`

Required: No

DataRearrangement
A JSON string that represents the splitting and rearrangement requirement used when this `DataSource` was created.

Type: String

Required: No

DataSizeInBytes
The total number of observations contained in the data files that the `DataSource` references.

Type: Long

Required: No

DataSourceId
The ID that is assigned to the `DataSource` during creation.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

LastUpdatedAt
The time of the most recent edit to the `BatchPrediction`. The time is expressed in epoch time.

Type: DateTime

Required: No

Message
A description of the most recent details about creating the `DataSource`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 10240.

Required: No

Name
A user-supplied name or description of the `DataSource`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

NumberOfFiles
The number of data files referenced by the `DataSource`.

Type: Long

Required: No

RDSMetadata
The datasource details that are specific to Amazon RDS.

Type: RDSMetadata (p. 127) object

Required: No

RedshiftMetadata
Describes the `DataSource` details specific to Amazon Redshift.

Type: RedshiftMetadata (p. 133) object

Required: No

RoleARN
The Amazon Resource Name (ARN) of an AWS IAM Role, such as the following: arn:aws:iam::account:role/rolename.

Type: String

Length constraints: Minimum length of 1. Maximum length of 110.

Required: No

Status
The current status of the `DataSource`. This element can have one of the following values:

- PENDING - Amazon Machine Learning (Amazon ML) submitted a request to create a `DataSource`.
- INPROGRESS - The creation process is underway.
- FAILED - The request to create a `DataSource` did not run to completion. It is not usable.

- COMPLETED - The creation process completed successfully.
- DELETED - The `DataSource` is marked as deleted. It is not usable.

Type: String

Valid Values: `PENDING | INPROGRESS | FAILED | COMPLETED | DELETED`

Required: No

Evaluation

Description

Represents the output of `GetEvaluation` operation.

The content consists of the detailed metadata and data file information and the current status of the `Evaluation`.

Contents

CreatedAt
The time that the `Evaluation` was created. The time is expressed in epoch time.

Type: DateTime

Required: No

CreatedByIamUser
The AWS user account that invoked the evaluation. The account type can be either an AWS root account or an AWS Identity and Access Management (IAM) user account.

Type: String

Pattern: `arn:aws:iam::[0-9]+:((user/.+)|(root))`

Required: No

EvaluationDataSourceId
The ID of the `DataSource` that is used to evaluate the `MLModel`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

EvaluationId
The ID that is assigned to the `Evaluation` at creation.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

InputDataLocationS3
The location and name of the data in Amazon Simple Storage Server (Amazon S3) that is used in the evaluation.

Type: String

Length constraints: Minimum length of 0. Maximum length of 2048.

Pattern: `s3://([^/]+)(/.*)?`

Required: No

LastUpdatedAt
The time of the most recent edit to the `Evaluation`. The time is expressed in epoch time.

Type: DateTime

Required: No

Message
A description of the most recent details about evaluating the `MLModel`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 10240.

Required: No

MLModelId
The ID of the `MLModel` that is the focus of the evaluation.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

Name
A user-supplied name or description of the `Evaluation`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Pattern: `.*\S.*|^$`

Required: No

PerformanceMetrics
Measurements of how well the `MLModel` performed, using observations referenced by the `DataSource`. One of the following metrics is returned, based on the type of the `MLModel`:

- BinaryAUC: A binary `MLModel` uses the Area Under the Curve (AUC) technique to measure performance.
- RegressionRMSE: A regression `MLModel` uses the Root Mean Square Error (RMSE) technique to measure performance. RMSE measures the difference between predicted and actual values for a single variable.
- MulticlassAvgFScore: A multiclass `MLModel` uses the F1 score technique to measure performance.

For more information about performance metrics, please see the Amazon Machine Learning Developer Guide.

Type: PerformanceMetrics (p. 121) object

Required: No

Status

The status of the evaluation. This element can have one of the following values:

- `PENDING` - Amazon Machine Learning (Amazon ML) submitted a request to evaluate an `MLModel`.
- `INPROGRESS` - The evaluation is underway.
- `FAILED` - The request to evaluate an `MLModel` did not run to completion. It is not usable.
- `COMPLETED` - The evaluation process completed successfully.
- `DELETED` - The `Evaluation` is marked as deleted. It is not usable.

Type: String

Valid Values: `PENDING | INPROGRESS | FAILED | COMPLETED | DELETED`

Required: No

MLModel

Description

Represents the output of a `GetMLModel` operation.

The content consists of the detailed metadata and the current status of the `MLModel`.

Contents

Algorithm

The algorithm used to train the `MLModel`. The following algorithm is supported:

- `SGD` -- Stochastic gradient descent. The goal of `SGD` is to minimize the gradient of the loss function.

Type: String

Valid Values: `sgd`

Required: No

CreatedAt

The time that the `MLModel` was created. The time is expressed in epoch time.

Type: DateTime

Required: No

CreatedByIamUser

The AWS user account from which the `MLModel` was created. The account type can be either an AWS root account or an AWS Identity and Access Management (IAM) user account.

Type: String

Pattern: `arn:aws:iam::[0-9]+:((user/.+)|(root))`

Required: No

EndpointInfo
The current endpoint of the `MLModel`.

Type: RealtimeEndpointInfo (p. 128) object

Required: No

InputDataLocationS3
The location of the data file or directory in Amazon Simple Storage Service (Amazon S3).

Type: String

Length constraints: Minimum length of 0. Maximum length of 2048.

Pattern: `s3://([^/]+)(/.*)?`

Required: No

LastUpdatedAt
The time of the most recent edit to the `MLModel`. The time is expressed in epoch time.

Type: DateTime

Required: No

Message
A description of the most recent details about accessing the `MLModel`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 10240.

Required: No

MLModelId
The ID assigned to the `MLModel` at creation.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

MLModelType
Identifies the `MLModel` category. The following are the available types:

- `REGRESSION` - Produces a numeric result. For example, "What price should a house be listed at?"
- `BINARY` - Produces one of two possible results. For example, "Is this a child-friendly web site?".
- `MULTICLASS` - Produces one of several possible results. For example, "Is this a HIGH-, LOW-, or MEDIUM-risk trade?".

Type: String

Valid Values: `REGRESSION | BINARY | MULTICLASS`

Required: No

Name
A user-supplied name or description of the `MLModel`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 1024.

Required: No

ScoreThreshold
Type: Float

Required: No

ScoreThresholdLastUpdatedAt
The time of the most recent edit to the `ScoreThreshold`. The time is expressed in epoch time.

Type: DateTime

Required: No

SizeInBytes
Long integer type that is a 64-bit signed number.

Type: Long

Required: No

Status
The current status of an `MLModel`. This element can have one of the following values:

- `PENDING` - Amazon Machine Learning (Amazon ML) submitted a request to create an `MLModel`.
- `INPROGRESS` - The creation process is underway.
- `FAILED` - The request to create an `MLModel` didn't run to completion. The model isn't usable.
- `COMPLETED` - The creation process completed successfully.
- `DELETED` - The `MLModel` is marked as deleted. It isn't usable.

Type: String

Valid Values: `PENDING | INPROGRESS | FAILED | COMPLETED | DELETED`

Required: No

TrainingDataSourceId
The ID of the training `DataSource`. The `CreateMLModel` operation uses the `TrainingDataSourceId`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

TrainingParameters
A list of the training parameters in the `MLModel`. The list is implemented as a map of key-value pairs.

The following is the current set of training parameters:

- `sgd.maxMLModelSizeInBytes` - The maximum allowed size of the model. Depending on the input data, the size of the model might affect its performance.

  The value is an integer that ranges from `100000` to `2147483648`. The default value is `33554432`.
- `sgd.maxPasses` - The number of times that the training process traverses the observations to build the `MLModel`. The value is an integer that ranges from `1` to `10000`. The default value is `10`.
- `sgd.shuffleType` - Whether Amazon ML shuffles the training data. Shuffling the data improves a model's ability to find the optimal solution for a variety of data types. The valid values are `auto` and `none`. The default value is `none`.

- `sgd.l1RegularizationAmount` - The coefficient regularization L1 norm, which controls overfitting the data by penalizing large coefficients. This parameter tends to drive coefficients to zero, resulting in sparse feature set. If you use this parameter, start by specifying a small value, such as `1.0E-08`.

  The value is a double that ranges from `0` to `MAX_DOUBLE`. The default is to not use L1 normalization. This parameter can't be used when `L2` is specified. Use this parameter sparingly.
- `sgd.l2RegularizationAmount` - The coefficient regularization L2 norm, which controls overfitting the data by penalizing large coefficients. This tends to drive coefficients to small, nonzero values. If you use this parameter, start by specifying a small value, such as `1.0E-08`.

  The value is a double that ranges from `0` to `MAX_DOUBLE`. The default is to not use L2 normalization. This parameter can't be used when `L1` is specified. Use this parameter sparingly.

Type: String to String map

Required: No

PerformanceMetrics

Description

Measurements of how well the `MLModel` performed on known observations. One of the following metrics is returned, based on the type of the `MLModel`:

- BinaryAUC: The binary `MLModel` uses the Area Under the Curve (AUC) technique to measure performance.
- RegressionRMSE: The regression `MLModel` uses the Root Mean Square Error (RMSE) technique to measure performance. RMSE measures the difference between predicted and actual values for a single variable.
- MulticlassAvgFScore: The multiclass `MLModel` uses the F1 score technique to measure performance.

For more information about performance metrics, please see the Amazon Machine Learning Developer Guide.

Contents

Properties

Type: String to String map

Required: No

Prediction

Description

The output from a `Predict` operation:

- `Details` - Contains the following attributes: `DetailsAttributes.PREDICTIVE_MODEL_TYPE - REGRESSION | BINARY | MULTICLASS DetailsAttributes.ALGORITHM - SGD`

- `PredictedLabel` - Present for either a `BINARY` or `MULTICLASS` `MLModel` request.
- `PredictedScores` - Contains the raw classification score corresponding to each label.
- `PredictedValue` - Present for a `REGRESSION` `MLModel` request.

Contents

Details
Provides any additional details regarding the prediction.

Type: String to String map

Valid Map Keys: `PredictiveModelType | Algorithm`

Required: No

PredictedLabel
The prediction label for either a `BINARY` or `MULTICLASS` `MLModel`.

Type: String

Length constraints: Minimum length of 1.

Required: No

PredictedScores
Provides the raw classification score corresponding to each label.

Type: String to Float map

Required: No

PredictedValue
The prediction value for `REGRESSIONMLModel`

.

Type: Float

Required: No

RDSDatabase

Description

The database details of an Amazon RDS database.

Contents

DatabaseName
The name of a database hosted on an RDS DB instance.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Required: Yes

InstanceIdentifier
The ID of an RDS DB instance.

Type: String

Length constraints: Minimum length of 1. Maximum length of 63.

Pattern: `[a-z0-9-]+`

Required: Yes

RDSDatabaseCredentials

Description

The database credentials to connect to a database on an RDS DB instance.

Contents

Password
The password to be used by Amazon ML to connect to a database on an RDS DB instance. The password should have sufficient permissions to execute the `RDSSelectQuery` query.

Type: String

Length constraints: Minimum length of 8. Maximum length of 128.

Required: Yes

Username
The username to be used by Amazon ML to connect to database on an Amazon RDS instance. The username should have sufficient permissions to execute an `RDSSelectSqlQuery` query.

Type: String

Length constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

RDSDataSpec

Description

The data specification of an Amazon Relational Database Service (Amazon RDS) `DataSource`.

Contents

DatabaseCredentials
The AWS Identity and Access Management (IAM) credentials that are used connect to the Amazon RDS database.

Type: RDSDatabaseCredentials (p. 123) object

Required: Yes

DatabaseInformation
Describes the `DatabaseName` and `InstanceIdentifier` of an Amazon RDS database.

Type: RDSDatabase (p. 122) object

Required: Yes

DataRearrangement
A JSON string that represents the splitting and rearrangement processing to be applied to a `DataSource`. If the `DataRearrangement` parameter is not provided, all of the input data is used to create the `Datasource`.

There are multiple parameters that control what data is used to create a datasource:

- **percentBegin**

  Use `percentBegin` to indicate the beginning of the range of the data used to create the Datasource. If you do not include `percentBegin` and `percentEnd`, Amazon ML includes all of the data when creating the datasource.

- **percentEnd**

  Use `percentEnd` to indicate the end of the range of the data used to create the Datasource. If you do not include `percentBegin` and `percentEnd`, Amazon ML includes all of the data when creating the datasource.

- **complement**

  The `complement` parameter instructs Amazon ML to use the data that is not included in the range of `percentBegin` to `percentEnd` to create a datasource. The `complement` parameter is useful if you need to create complementary datasources for training and evaluation. To create a complementary datasource, use the same values for `percentBegin` and `percentEnd`, along with the `complement` parameter.

  For example, the following two datasources do not share any data, and can be used to train and evaluate a model. The first datasource has 25 percent of the data, and the second one has 75 percent of the data.

  Datasource for evaluation: `{"splitting":{"percentBegin":0, "percentEnd":25}}`

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

- **strategy**

  To change how Amazon ML splits the data for a datasource, use the `strategy` parameter.

  The default value for the `strategy` parameter is `sequential`, meaning that Amazon ML takes all of the data records between the `percentBegin` and `percentEnd` parameters for the datasource, in the order that the records appear in the input data.

  The following two `DataRearrangement` lines are examples of sequentially ordered training and evaluation datasources:

  Datasource for evaluation: `{"splitting":{"percentBegin":70, "percentEnd":100, "strategy":"sequential"}}`

  Datasource for training: `{"splitting":{"percentBegin":70, "percentEnd":100, "strategy":"sequential", "complement":"true"}}`

  To randomly split the input data into the proportions indicated by the percentBegin and percentEnd parameters, set the `strategy` parameter to `random` and provide a string that is used as the seed value for the random data splitting (for example, you can use the S3 path to your data as the random seed string). If you choose the random split strategy, Amazon ML

assigns each row of data a pseudo-random number between 0 and 100, and then selects the rows that have an assigned number between `percentBegin` and `percentEnd`. Pseudo-random numbers are assigned using both the input seed string value and the byte offset as a seed, so changing the data results in a different split. Any existing ordering is preserved. The random splitting strategy ensures that variables in the training and evaluation data are distributed similarly. It is useful in the cases where the input data may have an implicit sort order, which would otherwise result in training and evaluation datasources containing non-similar data records.

The following two `DataRearrangement` lines are examples of non-sequentially ordered training and evaluation datasources:

Datasource for evaluation: `{"splitting":{"percentBegin":70, "percentEnd":100, "strategy":"random", "randomSeed"="s3://my_s3_path/bucket/file.csv"}}`

Datasource for training: `{"splitting":{"percentBegin":70, "percentEnd":100, "strategy":"random", "randomSeed"="s3://my_s3_path/bucket/file.csv", "complement":"true"}}`

Type: String

Required: No

DataSchema

A JSON string that represents the schema for an Amazon RDS `DataSource`. The `DataSchema` defines the structure of the observation data in the data file(s) referenced in the `DataSource`.

A `DataSchema` is not required if you specify a `DataSchemaUri`

Define your `DataSchema` as a series of key-value pairs. `attributes` and `excludedVariableNames` have an array of key-value pairs for their value. Use the following format to define your `DataSchema`.

{ "version": "1.0",

"recordAnnotationFieldName": "F1",

"recordWeightFieldName": "F2",

"targetFieldName": "F3",

"dataFormat": "CSV",

"dataFileContainsHeader": true,

"attributes": [

{ "fieldName": "F1", "fieldType": "TEXT" }, { "fieldName": "F2", "fieldType": "NUMERIC" }, { "fieldName": "F3", "fieldType": "CATEGORICAL" }, { "fieldName": "F4", "fieldType": "NUMERIC" }, { "fieldName": "F5", "fieldType": "CATEGORICAL" }, { "fieldName": "F6", "fieldType": "TEXT" }, { "fieldName": "F7", "fieldType": "WEIGHTED_INT_SEQUENCE" }, { "fieldName": "F8", "fieldType": "WEIGHTED_STRING_SEQUENCE" }],

"excludedVariableNames": ["F6"] }

Type: String

Length constraints: Minimum length of 0. Maximum length of 131071.

Required: No

DataSchemaUri

The Amazon S3 location of the `DataSchema`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 2048.

Pattern: `s3://([^/]+)(/.*)?`

Required: No

ResourceRole
The role (DataPipelineDefaultResourceRole) assumed by an Amazon Elastic Compute Cloud (Amazon EC2) instance to carry out the copy operation from Amazon RDS to an Amazon S3 task. For more information, see Role templates for data pipelines.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Required: Yes

S3StagingLocation
The Amazon S3 location for staging Amazon RDS data. The data retrieved from Amazon RDS using `SelectSqlQuery` is stored in this location.

Type: String

Length constraints: Minimum length of 0. Maximum length of 2048.

Pattern: `s3://([^/]+)(/.*)?`

Required: Yes

SecurityGroupIds
The security group IDs to be used to access a VPC-based RDS DB instance. Ensure that there are appropriate ingress rules set up to allow access to the RDS DB instance. This attribute is used by Data Pipeline to carry out the copy operation from Amazon RDS to an Amazon S3 task.

Type: array of Strings

Required: Yes

SelectSqlQuery
The query that is used to retrieve the observation data for the `DataSource`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 16777216.

Required: Yes

ServiceRole
The role (DataPipelineDefaultRole) assumed by AWS Data Pipeline service to monitor the progress of the copy task from Amazon RDS to Amazon S3. For more information, see Role templates for data pipelines.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Required: Yes

SubnetId
The subnet ID to be used to access a VPC-based RDS DB instance. This attribute is used by Data Pipeline to carry out the copy task from Amazon RDS to Amazon S3.

Type: String

Length constraints: Minimum length of 1. Maximum length of 255.

Required: Yes

RDSMetadata

Description

The datasource details that are specific to Amazon RDS.

Contents

Database
The database details required to connect to an Amazon RDS.

Type: RDSDatabase (p. 122) object

Required: No

DatabaseUserName
The username to be used by Amazon ML to connect to database on an Amazon RDS instance. The username should have sufficient permissions to execute an `RDSSelectSqlQuery` query.

Type: String

Length constraints: Minimum length of 1. Maximum length of 128.

Required: No

DataPipelineId
The ID of the Data Pipeline instance that is used to carry to copy data from Amazon RDS to Amazon S3. You can use the ID to find details about the instance in the Data Pipeline console.

Type: String

Length constraints: Minimum length of 1. Maximum length of 1024.

Required: No

ResourceRole
The role (DataPipelineDefaultResourceRole) assumed by an Amazon EC2 instance to carry out the copy task from Amazon RDS to Amazon S3. For more information, see Role templates for data pipelines.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Required: No

SelectSqlQuery
The SQL query that is supplied during CreateDataSourceFromRDS (p. 10). Returns only if `Verbose` is true in `GetDataSourceInput`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 16777216.

Required: No

ServiceRole
The role (DataPipelineDefaultRole) assumed by the Data Pipeline service to monitor the progress of the copy task from Amazon RDS to Amazon S3. For more information, see Role templates for data pipelines.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Required: No

RealtimeEndpointInfo

Description

Describes the real-time endpoint information for an `MLModel`.

Contents

CreatedAt
The time that the request to create the real-time endpoint for the `MLModel` was received. The time is expressed in epoch time.

Type: DateTime

Required: No

EndpointStatus
The current status of the real-time endpoint for the `MLModel`. This element can have one of the following values:

- `NONE` - Endpoint does not exist or was previously deleted.
- `READY` - Endpoint is ready to be used for real-time predictions.
- `UPDATING` - Updating/creating the endpoint.

Type: String

Valid Values: `NONE | READY | UPDATING | FAILED`

Required: No

EndpointUrl
The URI that specifies where to send real-time prediction requests for the `MLModel`.

> **Note**
> The application must wait until the real-time endpoint is ready before using this URI.

Type: String

Length constraints: Minimum length of 0. Maximum length of 2048.

Pattern: `https://[a-zA-Z0-9-.]*\.amazon(aws)?\.com[/]?`

Required: No

PeakRequestsPerSecond
The maximum processing rate for the real-time endpoint for `MLModel`, measured in incoming requests per second.

Type: Number

Required: No

RedshiftDatabase

Description

Describes the database details required to connect to an Amazon Redshift database.

Contents

ClusterIdentifier
The ID of an Amazon Redshift cluster.

Type: String

Length constraints: Minimum length of 1. Maximum length of 63.

Pattern: `[a-z0-9-]+`

Required: Yes

DatabaseName
The name of a database hosted on an Amazon Redshift cluster.

Type: String

Length constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-z0-9]+`

Required: Yes

RedshiftDatabaseCredentials

Description

Describes the database credentials for connecting to a database on an Amazon Redshift cluster.

Contents

Password
A password to be used by Amazon ML to connect to a database on an Amazon Redshift cluster. The password should have sufficient permissions to execute a `RedshiftSelectSqlQuery` query. The password should be valid for an Amazon Redshift USER.

Type: String

Length constraints: Minimum length of 8. Maximum length of 64.

Required: Yes

Username
A username to be used by Amazon Machine Learning (Amazon ML)to connect to a database on an Amazon Redshift cluster. The username should have sufficient permissions to execute the `RedshiftSelectSqlQuery` query. The username should be valid for an Amazon Redshift USER.

Type: String

Length constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

RedshiftDataSpec

Description

Describes the data specification of an Amazon Redshift `DataSource`.

Contents

DatabaseCredentials
Describes AWS Identity and Access Management (IAM) credentials that are used connect to the Amazon Redshift database.

Type: RedshiftDatabaseCredentials (p. 129) object

Required: Yes

DatabaseInformation
Describes the `DatabaseName` and `ClusterIdentifier` for an Amazon Redshift `DataSource`.

Type: RedshiftDatabase (p. 129) object

Required: Yes

DataRearrangement
A JSON string that represents the splitting and rearrangement processing to be applied to a `DataSource`. If the `DataRearrangement` parameter is not provided, all of the input data is used to create the `Datasource`.

There are multiple parameters that control what data is used to create a datasource:

- **percentBegin**

  Use `percentBegin` to indicate the beginning of the range of the data used to create the Datasource. If you do not include `percentBegin` and `percentEnd`, Amazon ML includes all of the data when creating the datasource.

- **percentEnd**

  Use `percentEnd` to indicate the end of the range of the data used to create the Datasource. If you do not include `percentBegin` and `percentEnd`, Amazon ML includes all of the data when creating the datasource.

- **complement**

  The `complement` parameter instructs Amazon ML to use the data that is not included in the range of `percentBegin` to `percentEnd` to create a datasource. The `complement` parameter is useful if you need to create complementary datasources for training and evaluation. To create

a complementary datasource, use the same values for `percentBegin` and `percentEnd`, along with the `complement` parameter.

For example, the following two datasources do not share any data, and can be used to train and evaluate a model. The first datasource has 25 percent of the data, and the second one has 75 percent of the data.

Datasource for evaluation: `{"splitting":{"percentBegin":0, "percentEnd":25}}`

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

- **strategy**

  To change how Amazon ML splits the data for a datasource, use the `strategy` parameter.

  The default value for the `strategy` parameter is `sequential`, meaning that Amazon ML takes all of the data records between the `percentBegin` and `percentEnd` parameters for the datasource, in the order that the records appear in the input data.

  The following two `DataRearrangement` lines are examples of sequentially ordered training and evaluation datasources:

  Datasource for evaluation: `{"splitting":{"percentBegin":70, "percentEnd":100, "strategy":"sequential"}}`

  Datasource for training: `{"splitting":{"percentBegin":70, "percentEnd":100, "strategy":"sequential", "complement":"true"}}`

  To randomly split the input data into the proportions indicated by the percentBegin and percentEnd parameters, set the `strategy` parameter to `random` and provide a string that is used as the seed value for the random data splitting (for example, you can use the S3 path to your data as the random seed string). If you choose the random split strategy, Amazon ML assigns each row of data a pseudo-random number between 0 and 100, and then selects the rows that have an assigned number between `percentBegin` and `percentEnd`. Pseudo-random numbers are assigned using both the input seed string value and the byte offset as a seed, so changing the data results in a different split. Any existing ordering is preserved. The random splitting strategy ensures that variables in the training and evaluation data are distributed similarly. It is useful in the cases where the input data may have an implicit sort order, which would otherwise result in training and evaluation datasources containing non-similar data records.

  The following two `DataRearrangement` lines are examples of non-sequentially ordered training and evaluation datasources:

  Datasource for evaluation: `{"splitting":{"percentBegin":70, "percentEnd":100, "strategy":"random", "randomSeed"="s3://my_s3_path/bucket/file.csv"}}`

  Datasource for training: `{"splitting":{"percentBegin":70, "percentEnd":100, "strategy":"random", "randomSeed"="s3://my_s3_path/bucket/file.csv", "complement":"true"}}`

Type: String

Required: No

DataSchema

A JSON string that represents the schema for an Amazon Redshift `DataSource`. The `DataSchema` defines the structure of the observation data in the data file(s) referenced in the `DataSource`.

A `DataSchema` is not required if you specify a `DataSchemaUri`.

Define your `DataSchema` as a series of key-value pairs. `attributes` and `excludedVariableNames` have an array of key-value pairs for their value. Use the following format to define your `DataSchema`.

{ "version": "1.0",

"recordAnnotationFieldName": "F1",

"recordWeightFieldName": "F2",

"targetFieldName": "F3",

"dataFormat": "CSV",

"dataFileContainsHeader": true,

"attributes": [

{ "fieldName": "F1", "fieldType": "TEXT" }, { "fieldName": "F2", "fieldType": "NUMERIC" }, { "fieldName": "F3", "fieldType": "CATEGORICAL" }, { "fieldName": "F4", "fieldType": "NUMERIC" }, { "fieldName": "F5", "fieldType": "CATEGORICAL" }, { "fieldName": "F6", "fieldType": "TEXT" }, { "fieldName": "F7", "fieldType": "WEIGHTED_INT_SEQUENCE" }, { "fieldName": "F8", "fieldType": "WEIGHTED_STRING_SEQUENCE" }],

"excludedVariableNames": ["F6"] }

Type: String

Length constraints: Minimum length of 0. Maximum length of 131071.

Required: No

DataSchemaUri

Describes the schema location for an Amazon Redshift `DataSource`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 2048.

Pattern: `s3://([^/]+)(/.*)?`

Required: No

S3StagingLocation

Describes an Amazon S3 location to store the result set of the `SelectSqlQuery` query.

Type: String

Length constraints: Minimum length of 0. Maximum length of 2048.

Pattern: `s3://([^/]+)(/.*)?`

Required: Yes

SelectSqlQuery

Describes the SQL Query to execute on an Amazon Redshift database for an Amazon Redshift `DataSource`.

Type: String

Length constraints: Minimum length of 1. Maximum length of 16777216.

Required: Yes

RedshiftMetadata

Description

Describes the `DataSource` details specific to Amazon Redshift.

Contents

DatabaseUserName
A username to be used by Amazon Machine Learning (Amazon ML)to connect to a database on an Amazon Redshift cluster. The username should have sufficient permissions to execute the `RedshiftSelectSqlQuery` query. The username should be valid for an Amazon Redshift USER.

Type: String

Length constraints: Minimum length of 1. Maximum length of 128.

Required: No

RedshiftDatabase
Describes the database details required to connect to an Amazon Redshift database.

Type: RedshiftDatabase (p. 129) object

Required: No

SelectSqlQuery
The SQL query that is specified during CreateDataSourceFromRedshift (p. 15). Returns only if `Verbose` is true in GetDataSourceInput.

Type: String

Length constraints: Minimum length of 1. Maximum length of 16777216.

Required: No

S3DataSpec

Description

Describes the data specification of a `DataSource`.

Contents

DataLocationS3
The location of the data file(s) used by a `DataSource`. The URI specifies a data file or an Amazon Simple Storage Service (Amazon S3) directory or bucket containing data files.

Type: String

Length constraints: Minimum length of 0. Maximum length of 2048.

Pattern: `s3://([^/]+)(/.*)?`

Required: Yes

DataRearrangement

A JSON string that represents the splitting and rearrangement processing to be applied to a `DataSource`. If the `DataRearrangement` parameter is not provided, all of the input data is used to create the `Datasource`.

There are multiple parameters that control what data is used to create a datasource:

- **percentBegin**

  Use `percentBegin` to indicate the beginning of the range of the data used to create the Datasource. If you do not include `percentBegin` and `percentEnd`, Amazon ML includes all of the data when creating the datasource.

- **percentEnd**

  Use `percentEnd` to indicate the end of the range of the data used to create the Datasource. If you do not include `percentBegin` and `percentEnd`, Amazon ML includes all of the data when creating the datasource.

- **complement**

  The `complement` parameter instructs Amazon ML to use the data that is not included in the range of `percentBegin` to `percentEnd` to create a datasource. The `complement` parameter is useful if you need to create complementary datasources for training and evaluation. To create a complementary datasource, use the same values for `percentBegin` and `percentEnd`, along with the `complement` parameter.

  For example, the following two datasources do not share any data, and can be used to train and evaluate a model. The first datasource has 25 percent of the data, and the second one has 75 percent of the data.

  Datasource for evaluation: `{"splitting":{"percentBegin":0, "percentEnd":25}}`

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

- **strategy**

  To change how Amazon ML splits the data for a datasource, use the `strategy` parameter.

  The default value for the `strategy` parameter is `sequential`, meaning that Amazon ML takes all of the data records between the `percentBegin` and `percentEnd` parameters for the datasource, in the order that the records appear in the input data.

  The following two `DataRearrangement` lines are examples of sequentially ordered training and evaluation datasources:

  Datasource for evaluation: `{"splitting":{"percentBegin":70, "percentEnd":100, "strategy":"sequential"}}`

  Datasource for training: `{"splitting":{"percentBegin":70, "percentEnd":100, "strategy":"sequential", "complement":"true"}}`

  To randomly split the input data into the proportions indicated by the percentBegin and percentEnd parameters, set the `strategy` parameter to `random` and provide a string that is used as the seed value for the random data splitting (for example, you can use the S3 path to your data as the random seed string). If you choose the random split strategy, Amazon ML assigns each row of data a pseudo-random number between 0 and 100, and then selects the rows that have an assigned number between `percentBegin` and `percentEnd`. Pseudo-random numbers are assigned using both the input seed string value and the byte offset as a seed, so changing the data results in a different split. Any existing ordering is preserved. The random splitting strategy ensures that variables in the training and evaluation data are distributed similarly. It is useful in the cases where the input data may have an implicit sort order,

which would otherwise result in training and evaluation datasources containing non-similar data records.

The following two `DataRearrangement` lines are examples of non-sequentially ordered training and evaluation datasources:

Datasource for evaluation: `{"splitting":{"percentBegin":70, "percentEnd":100, "strategy":"random", "randomSeed"="s3://my_s3_path/bucket/file.csv"}}`

Datasource for training: `{"splitting":{"percentBegin":70, "percentEnd":100, "strategy":"random", "randomSeed"="s3://my_s3_path/bucket/file.csv", "complement":"true"}}`

Type: String

Required: No

DataSchema

A JSON string that represents the schema for an Amazon S3 `DataSource`. The `DataSchema` defines the structure of the observation data in the data file(s) referenced in the `DataSource`.

You must provide either the `DataSchema` or the `DataSchemaLocationS3`.

Define your `DataSchema` as a series of key-value pairs. `attributes` and `excludedVariableNames` have an array of key-value pairs for their value. Use the following format to define your `DataSchema`.

{ "version": "1.0",

"recordAnnotationFieldName": "F1",

"recordWeightFieldName": "F2",

"targetFieldName": "F3",

"dataFormat": "CSV",

"dataFileContainsHeader": true,

"attributes": [

{ "fieldName": "F1", "fieldType": "TEXT" }, { "fieldName": "F2", "fieldType": "NUMERIC" }, { "fieldName": "F3", "fieldType": "CATEGORICAL" }, { "fieldName": "F4", "fieldType": "NUMERIC" }, { "fieldName": "F5", "fieldType": "CATEGORICAL" }, { "fieldName": "F6", "fieldType": "TEXT" }, { "fieldName": "F7", "fieldType": "WEIGHTED_INT_SEQUENCE" }, { "fieldName": "F8", "fieldType": "WEIGHTED_STRING_SEQUENCE" }],

"excludedVariableNames": ["F6"] }

Type: String

Length constraints: Minimum length of 0. Maximum length of 131071.

Required: No

DataSchemaLocationS3

Describes the schema location in Amazon S3. You must provide either the `DataSchema` or the `DataSchemaLocationS3`.

Type: String

Length constraints: Minimum length of 0. Maximum length of 2048.

Pattern: `s3://([^/]+)(/.*)?`

Required: No

Tag

Description

A custom key-value pair associated with an ML object, such as an ML model.

Contents

Key

A unique identifier for the tag. Valid characters include Unicode letters, digits, white space, _, ., /, =, +, -, %, and @.

Type: String

Length constraints: Minimum length of 1. Maximum length of 128.

Pattern: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Required: No

Value

An optional string, typically used to describe or define the tag. Valid characters include Unicode letters, digits, white space, _, ., /, =, +, -, %, and @.

Type: String

Length constraints: Minimum length of 0. Maximum length of 256.

Pattern: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Required: No

Common Parameters

This section lists the request parameters that all actions use. Any action-specific parameters are listed in the topic for the action.

Action
The action to be performed.

Default: None

Type: string

Required: Yes

AuthParams
The parameters that are required to authenticate a Conditional request. Contains:

- AWSAccessKeyID
- SignatureVersion
- Timestamp
- Signature

Default: None

Required: Conditional

AWSAccessKeyId
The access key ID that corresponds to the secret access key that you used to sign the request.

Default: None

Type: string

Required: Yes

Expires
The date and time when the request signature expires, expressed in the format YYYY-MM-DDThh:mm:ssZ, as specified in the ISO 8601 standard.

Condition: Requests must include either *Timestamp* or *Expires*, but not both.

Default: None

Type: string

Required: Conditional

SecurityToken
The temporary security token that was obtained through a call to AWS Security Token Service. For a list of services that support AWS Security Token Service, go to Using Temporary Security Credentials to Access AWS in **Using Temporary Security Credentials**.

Default: None

Type: string

Required: No

Signature
The digital signature that you created for the request. For information about generating a signature, go to the service's developer documentation.

Default: None

Type: string

Required: Yes

SignatureMethod
The hash algorithm that you used to create the request signature.

Default: None

Type: string

Valid Values: `HmacSHA256 | HmacSHA1`

Required: Yes

SignatureVersion
The signature version you use to sign the request. Set this to the value that is recommended for your service.

Default: None

Type: string

Required: Yes

Timestamp
The date and time when the request was signed, expressed in the format YYYY-MM-DDThh:mm:ssZ, as specified in the ISO 8601 standard.

Condition: Requests must include either *Timestamp* or *Expires*, but not both.

Default: None

Type: string

Required: Conditional

Version
The API version that the request is written for, expressed in the format YYYY-MM-DD.

Default: None

Type: string

Required: Yes

Common Errors

This section lists the common errors that all actions return. Any action-specific errors are listed in the topic for the action.

IncompleteSignature
The request signature does not conform to AWS standards.

HTTP Status Code: 400

InternalFailure
The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

InvalidAction
The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 400

InvalidClientTokenId
The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

InvalidParameterCombination
Parameters that must not be used together were used together.

HTTP Status Code: 400

InvalidParameterValue
An invalid or out-of-range value was supplied for the input parameter.

HTTP Status Code: 400

InvalidQueryParameter
The AWS query string is malformed or does not adhere to AWS standards.

HTTP Status Code: 400

MalformedQueryString
The query string contains a syntax error.

HTTP Status Code: 404

MissingAction
The request is missing an action or a required parameter.

HTTP Status Code: 400

MissingAuthenticationToken
The request must contain either a valid (registered) AWS access key ID or X.509 certificate.

HTTP Status Code: 403

MissingParameter
A required parameter for the specified action is not supplied.

HTTP Status Code: 400

OptInRequired
The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

RequestExpired
The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

ServiceUnavailable
The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

Throttling
The request was denied due to request throttling.

HTTP Status Code: 400

ValidationError
The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400