sagemaker$_{pyspark}$Documentation

Release 1.4.6.dev0

Amazon Web Services

Aug 26, 2022

Contents:

The SageMaker PySpark SDK provides a pyspark interface to Amazon SageMaker, allowing customers to train using the Spark Estimator API, host their model on Amazon SageMaker, and make predictions with their model using the Spark Transformer API. This page is a quick guide on the basics of SageMaker PySpark. You can also check the *API* docs.

CHAPTER 1

Quick Start

First, install the library:

```
$ pip install sagemaker_pyspark
```

Next, set up credentials (in e.g. `~/.aws/credentials`):

```
[default]
aws_access_key_id = YOUR_KEY
aws_secret_access_key = YOUR_KEY
```

Then, set up a default region (in e.g. `~/.aws/config`):

```
[default]
region=us-west-2
```

Then, to load the sagemaker jars programatically:

```
from pyspark import SparkContext, SparkConf
import sagemaker_pyspark

conf = (SparkConf()
        .set("spark.driver.extraClassPath", ":".join(sagemaker_pyspark.classpath_
↪jars())))
SparkContext(conf=conf)
```

Alternatively pass the jars to your pyspark job via the –jars flag:

```
$ spark-submit --jars `sagemakerpyspark-jars`
```

If you want to play around in interactive mode, the pyspark shell can be used too:

```
$ pyspark --jars `sagemakerpyspark-jars`
```

You can also use the –packages flag and pass in the Maven coordinates for SageMaker Spark:

```
$ pyspark --packages com.amazonaws:sagemaker-spark_2.11:spark_2.1.1-1.0
```

CHAPTER 2

Training and Hosting a K-Means Clustering model using SageMaker PySpark

A KMeansSageMakerEstimator runs a training job using the Amazon SageMaker KMeans algorithm upon invocation of fit(), returning a SageMakerModel.

```
from sagemaker_pyspark import IAMRole
from sagemaker_pyspark.algorithms import KMeansSageMakerEstimator

iam_role = "arn:aws:iam:0123456789012:role/MySageMakerRole"

region = "us-east-1"
training_data = spark.read.format("libsvm").option("numFeatures", "784")
  .load("s3a://sagemaker-sample-data-{}/spark/mnist/train/".format(region))

test_data = spark.read.format("libsvm").option("numFeatures", "784")
  .load("s3a://sagemaker-sample-data-{}/spark/mnist/train/".format(region))

kmeans_estimator = KMeansSageMakerEstimator(
    trainingInstanceType="ml.m4.xlarge",
    trainingInstanceCount=1,
    endpointInstanceType="ml.m4.xlarge",
    endpointInitialInstanceCount=1,
    sagemakerRole=IAMRole(iam_role))

kmeans_estimator.setK(10)
kmeans_estimator.setFeatureDim(784)

kmeans_model = kmeans_estimator.fit(training_data)

transformed_data = kmeans_model.transform(test_data)
transformed_data.show()
```

The SageMakerEstimator expects an input DataFrame with a column named “features” that holds a Spark ML Vector. The estimator also serializes a “label” column of Doubles if present. Other columns are ignored. The dimension of this input vector should be equal to the feature dimension given as a hyperparameter.

The Amazon SageMaker KMeans algorithm accepts many parameters, but K (the number of clusters) and FeatureDim (the number of features per Row) are required.

You can set other hyperparameters, See the docs (link), or run

```
kmeans_estimator.explainParams()
```

After training is complete, an Amazon SageMaker Endpoint is created to host the model and serve predictions. Upon invocation of transform(), the SageMakerModel predicts against their hosted model. Like the SageMakerEstimator, the SageMakerModel expects an input DataFrame with a column named "features" that holds a Spark ML Vector equal in dimension to the value of the FeatureDim parameter.

CHAPTER 3

API Reference

If you are looking for information on a specific class or method, this is where its found.

3.1 API

This part of the documentation covers all the interfaces of sagemaker_pyspark.

3.1.1 SageMakerModel

class SageMakerModel(*endpointInstanceType*, *endpointInitialInstanceCount*, *requestRowSerializer*, *responseRowDeserializer*, *existingEndpointName=None*, *modelImage=None*, *modelPath=None*, *modelEnvironmentVariables=None*, *modelExecutionRoleARN=None*, *endpointCreationPolicy=<sagemaker_pyspark.SageMakerEstimator.EndpointCreationPolicy._CreateOnConstruct object>*, *sagemakerClient=<sagemaker_pyspark.SageMakerClients.SageMakerClients._SageMakerDef object>*, *prependResultRows=True*, *namePolicy=<sagemaker_pyspark.NamePolicy.RandomNamePolicy object>*, *uid=None*, *javaObject=None*)

Bases: `sagemaker_pyspark.wrapper.SageMakerJavaWrapper`, `pyspark.ml.wrapper.JavaModel`

A Model implementation which transforms a DataFrame by making requests to a SageMaker Endpoint. Manages life cycle of all necessary SageMaker entities, including Model, EndpointConfig, and Endpoint.

This Model transforms one DataFrame to another by repeated, distributed SageMaker Endpoint invocation. Each invocation request body is formed by concatenating input DataFrame Rows serialized to Byte Arrays by the specified *RequestRowSerializer*. The invocation request content-type property is set from `RequestRowSerializer.contentType`. The invocation request accepts property is set from `ResponseRowDeserializer.accepts`.

The transformed DataFrame is produced by deserializing each invocation response body into a series of Rows. Row deserialization is delegated to the specified *ResponseRowDeserializer*. If prependInputRows is

false, the transformed DataFrame will contain just these Rows. If prependInputRows is true, then each transformed Row is a concatenation of the input Row with its corresponding SageMaker invocation deserialized Row.

Each invocation of `transform()` passes the `Dataset.schema` of the input DataFrame to requestRowSerialize by invoking `RequestRowSerializer.setSchema()`.

The specified RequestRowSerializer also controls the validity of input Row Schemas for this Model. Schema validation is carried out on each call to `transformSchema()`, which invokes `RequestRowSerializer.validateSchema()`.

Adapting this SageMaker model to the data format and type of a specific Endpoint is achieved by subclassing RequestRowSerializer and ResponseRowDeserializer. Examples of a Serializer and Deseralizer are *LibSVMRequestRowSerializer* and *LibSVMResponseRowDeserializer* respectively.

Parameters

- **endpointInstanceType** (*str*) – The instance type used to run the model container
- **endpointInitialInstanceCount** (*int*) – The initial number of instances used to host the model
- **requestRowSerializer** (RequestRowSerializer) – Serializes a Row to an Array of Bytes
- **responseRowDeserializer** (ResponseRowDeserializer) – Deserializes an Array of Bytes to a series of Rows
- **existingEndpointName** (*str*) – An endpoint name
- **modelImage** (*str*) – A Docker image URI
- **modelPath** (*str*) – An S3 location that a successfully completed SageMaker Training Job has stored its model output to.
- **modelEnvironmentVariables** (*dict*) – The environment variables that SageMaker will set on the model container during execution.
- **modelExecutionRoleARN** (*str*) – The IAM Role used by SageMaker when running the hosted Model and to download model data from S3
- **endpointCreationPolicy** (EndpointCreationPolicy) – Whether the endpoint is created upon SageMakerModel construction, transformation, or not at all.
- **sagemakerClient** (*AmazonSageMaker*) – CreateModel, and CreateEndpoint requests.
- **prependResultRows** (*bool*) – Whether the transformation result should also include the input Rows. If true, each output Row is formed by a concatenation of the input Row with the corresponding Row produced by SageMaker invocation, produced by responseRowDeserializer. If false, each output Row is just taken from responseRowDeserializer.
- **namePolicy** (*NamePolicy*) – The NamePolicy to use when naming SageMaker entities created during usage of this Model.
- **uid** (*str*) – The unique identifier of this Estimator. Used to represent this stage in Spark ML pipelines.

copy(*extra=None*)

Creates a copy of this instance with the same uid and some extra params. This implementation first calls Params.copy and then make a copy of the companion Java pipeline component with extra params. So both the Python wrapper and the Java pipeline component get copied.

Parameters **extra** – Extra parameters to copy to the new instance

Returns Copy of this instance

explainParam(*param*)
Explains a single param and returns its name, doc, and optional default value and user-supplied value in a string.

explainParams()
Returns the documentation of all params with their optionally default values and user-supplied values.

extractParamMap(*extra=None*)
Extracts the embedded default param values and user-supplied values, and then merges them with extra values from input into a flat param map, where the latter value is used if there exist conflicts, i.e., with ordering: default param values < user-supplied values < extra.

Parameters **extra** – extra param values

Returns merged param map

classmethod fromEndpoint(*endpointName*, *requestRowSerializer*, *responseRowDeserializer*, *modelEnvironmentVariables=None*, *sagemakerClient=<sagemaker_pyspark.SageMakerClients.SageMakerClients._SageMakerDefaultClie object>*, *prependResultRows=True*, *namePolicy=<sagemaker_pyspark.NamePolicy.RandomNamePolicy object>*, *uid='sagemaker'*)
Creates a JavaSageMakerModel from existing model data in S3.

The returned JavaSageMakerModel can be used to transform Dataframes.

Parameters

- **endpointName** (*str*) – The name of an endpoint that is currently in service.
- **requestRowSerializer** (RequestRowSerializer) – Serializes a row to an array of bytes.
- **responseRowDeserializer** (ResponseRowDeserializer) – Deserializes an array of bytes to a series of rows.
- **modelEnvironmentVariables** – The environment variables that SageMaker will set on the model container during execution.
- **sagemakerClient** (*AmazonSageMaker*) – CreateTrainingJob, CreateModel, and CreateEndpoint requests.
- **prependResultRows** (*bool*) – Whether the transformation result should also include the input Rows. If true, each output Row is formed by a concatenation of the input Row with the corresponding Row produced by SageMaker invocation, produced by responseRowDeserializer. If false, each output Row is just taken from responseRowDeserializer.
- **namePolicy** (*NamePolicy*) – The NamePolicy to use when naming SageMaker entities created during usage of the returned model.
- **uid** (*String*) – The unique identifier of the SageMakerModel. Used to represent the stage in Spark ML pipelines.

Returns A JavaSageMakerModel that sends InvokeEndpoint requests to an endpoint hosting the training job's model.

Return type JavaSageMakerModel

classmethod fromModelS3Path(*modelPath*, *modelImage*, *modelExecutionRoleARN*, *endpointInstanceType*, *endpointInitialInstanceCount*, *requestRowSerializer*, *responseRowDeserializer*, *modelEnvironmentVariables=None*, *endpointCreationPolicy=<sagemaker_pyspark.SageMakerEstimator.EndpointCreationPolicy._CreateOnCo object>*, *sagemakerClient=<sagemaker_pyspark.SageMakerClients.SageMakerClients object>*, *prependResultRows=True*, *namePolicy=<sagemaker_pyspark.NamePolicy.RandomNamePolicy object>*, *uid='sagemaker'*)

Creates a JavaSageMakerModel from existing model data in S3.

The returned JavaSageMakerModel can be used to transform Dataframes.

Parameters

- **modelPath** (*str*) – The S3 URI to the model data to host.
- **modelImage** (*str*) – The URI of the image that will serve model inferences.
- **modelExecutionRoleARN** (*str*) – The IAM Role used by SageMaker when running the hosted Model and to download model data from S3.
- **endpointInstanceType** (*str*) – The instance type used to run the model container.
- **endpointInitialInstanceCount** (*int*) – The initial number of instances used to host the model.
- **requestRowSerializer** (RequestRowSerializer) – Serializes a row to an array of bytes.
- **responseRowDeserializer** (ResponseRowDeserializer) – Deserializes an array of bytes to a series of rows.
- **modelEnvironmentVariables** – The environment variables that SageMaker will set on the model container during execution.
- **endpointCreationPolicy** (EndpointCreationPolicy) – Whether the endpoint is created upon SageMakerModel construction, transformation, or not at all.
- **sagemakerClient** (*AmazonSageMaker*) – CreateTrainingJob, CreateModel, and CreateEndpoint requests.
- **prependResultRows** (*bool*) – Whether the transformation result should also include the input Rows. If true, each output Row is formed by a concatenation of the input Row with the corresponding Row produced by SageMaker invocation, produced by responseRowDeserializer. If false, each output Row is just taken from responseRowDeserializer.
- **namePolicy** (*NamePolicy*) – The NamePolicy to use when naming SageMaker entities created during usage of the returned model.
- **uid** (*String*) – The unique identifier of the SageMakerModel. Used to represent the stage in Spark ML pipelines.

Returns A JavaSageMakerModel that sends InvokeEndpoint requests to an endpoint hosting the training job's model.

Return type JavaSageMakerModel

classmethod fromTrainingJob(*trainingJobName*, *modelImage*, *modelExecutionRoleARN*, *endpointInstanceType*, *endpointInitialInstanceCount*, *requestRowSerializer*, *responseRowDeserializer*, *modelEnvironmentVariables=None*, *endpointCreationPolicy=<sagemaker_pyspark.SageMakerEstimator.EndpointCreationPolicy._CreateOnCo object>*, *sagemakerClient=<sagemaker_pyspark.SageMakerClients.SageMakerClients object>*, *prependResultRows=True*, *namePolicy=<sagemaker_pyspark.NamePolicy.RandomNamePolicy object>*, *uid='sagemaker'*)

Creates a JavaSageMakerModel from a successfully completed training job name.

The returned JavaSageMakerModel can be used to transform DataFrames.

Parameters

- **trainingJobName** (*str*) – Name of the successfully completed training job.
- **modelImage** (*str*) – URI of the image that will serve model inferences.
- **modelExecutionRoleARN** (*str*) – The IAM Role used by SageMaker when running the hosted Model and to download model data from S3.
- **endpointInstanceType** (*str*) – The instance type used to run the model container.
- **endpointInitialInstanceCount** (*int*) – The initial number of instances used to host the model.
- **requestRowSerializer** (RequestRowSerializer) – Serializes a row to an array of bytes.
- **responseRowDeserializer** (ResponseRowDeserializer) – Deserializes an array of bytes to a series of rows.
- **modelEnvironmentVariables** – The environment variables that SageMaker will set on the model container during execution.
- **endpointCreationPolicy** (EndpointCreationPolicy) – Whether the endpoint is created upon SageMakerModel construction, transformation, or not at all.
- **sagemakerClient** (*AmazonSageMaker*) – CreateTrainingJob, CreateModel, and CreateEndpoint requests.
- **prependResultRows** (*bool*) – Whether the transformation result should also include the input Rows. If true, each output Row is formed by a concatenation of the input Row with the corresponding Row produced by SageMaker invocation, produced by responseRowDeserializer. If false, each output Row is just taken from responseRowDeserializer.
- **namePolicy** (*NamePolicy*) – The NamePolicy to use when naming SageMaker entities created during usage of the returned model.
- **uid** (*String*) – The unique identifier of the SageMakerModel. Used to represent the stage in Spark ML pipelines.

Returns

a JavaSageMakerModel that sends InvokeEndpoint requests to an endpoint hosting the training job's model.

Return type JavaSageMakerModel

getOrDefault(*param*)

Gets the value of a param in the user-supplied param map or its default value. Raises an error if neither is set.

getParam(*paramName*)
Gets a param by its name.

hasDefault(*param*)
Checks whether a param has a default value.

hasParam(*paramName*)
Tests whether this instance contains a param with a given (string) name.

isDefined(*param*)
Checks whether a param is explicitly set by user or has a default value.

isSet(*param*)
Checks whether a param is explicitly set by user.

params
Returns all params ordered by name. The default implementation uses `dir()` to get all attributes of type `Param`.

transform(*dataset*)
Transforms the input dataset with optional parameters.

Parameters

- **dataset** – input dataset, which is an instance of `pyspark.sql.DataFrame`
- **params** – an optional param map that overrides embedded params.

Returns transformed dataset

New in version 1.3.0.

3.1.2 SageMakerEstimator

class SageMakerEstimator(*trainingImage*, *modelImage*, *trainingInstanceType*, *trainingInstanceCount*, *endpointInstanceType*, *endpointInitialInstanceCount*, *requestRowSerializer*, *responseRowDeserializer*, *hyperParameters=None*, *trainingInputS3DataPath=<sagemaker_pyspark.S3Resources.S3AutoCreatePath object>*, *trainingOutputS3DataPath=<sagemaker_pyspark.S3Resources.S3AutoCreatePath object>*, *trainingInstanceVolumeSizeInGB=1024*, *trainingProjectedColumns=None*, *trainingChannelName='train'*, *trainingContentType=None*, *trainingS3DataDistribution='ShardedByS3Key'*, *trainingSparkDataFormat='sagemaker'*, *trainingSparkDataFormatOptions=None*, *trainingInputMode='File'*, *trainingCompressionCodec=None*, *trainingMaxRuntimeInSeconds=86400*, *trainingKmsKeyId=None*, *modelEnvironmentVariables=None*, *endpointCreationPolicy=<sagemaker_pyspark.SageMakerEstimator.EndpointCreationPolicy._CreateOnConstruct object>*, *sagemakerClient=<sagemaker_pyspark.SageMakerClients.SageMakerClients._SageMak object>*, *sagemakerRole=<sagemaker_pyspark.IAMRoleResource.IAMRoleFromConfig object>*, *s3Client=<sagemaker_pyspark.SageMakerClients.SageMakerClients._S3DefaultClient object>*, *stsClient=<sagemaker_pyspark.SageMakerClients.SageMakerClients._STSDefaultClient object>*, *modelPrependInputRowsToTransformationRows=True*, *deleteStagingDataAfterTraining=True*, *namePolicyFactory=<sagemaker_pyspark.NamePolicy.RandomNamePolicyFactory object>*, *uid=None*)

Bases: `sagemaker_pyspark.SageMakerEstimator.SageMakerEstimatorBase`

Adapts a SageMaker learning Algorithm to a Spark Estimator.

Fits a *SageMakerModel* by running a SageMaker Training Job on a Spark Dataset. Each call to *fit()* submits a new SageMaker Training Job, creates a new SageMaker Model, and creates a new SageMaker Endpoint Config. A new Endpoint is either created by or the returned SageMakerModel is configured to generate an Endpoint on SageMakerModel transform.

On fit, the input `Dataset` is serialized with the specified trainingSparkDataFormat using the specified trainingSparkDataFormatOptions and uploaded to an S3 location specified by `trainingInputS3DataPath`. The serialized Dataset is compressed with `trainingCompressionCodec`, if not None.

`trainingProjectedColumns` can be used to control which columns on the input Dataset are transmitted to SageMaker. If not None, then only those column names will be serialized as input to the SageMaker Training Job.

A Training Job is created with the uploaded Dataset being input to the specified `trainingChannelName`, with the specified `trainingInputMode`. The algorithm is specified `trainingImage`, a Docker image URI reference. The Training Job is created with trainingInstanceCount instances of type `trainingInstanceType`. The Training Job will time-out after attr:*trainingMaxRuntimeInSeconds*, if not None.

SageMaker Training Job hyperparameters are built from the `params` on this Estimator. Param objects with neither a default value nor a set value are ignored. If a Param is not set but has a default value, the default value will be used. Param values are converted to SageMaker hyperparameter String values.

SageMaker uses the IAM Role with ARN `sagemakerRole` to access the input and output S3 buckets and trainingImage if the image is hosted in ECR. SageMaker Training Job output is stored in a Training Job specific sub-prefix of `trainingOutputS3DataPath`. This contains the SageMaker Training Job output file as well as the SageMaker Training Job model file.

After the Training Job is created, this Estimator will poll for success. Upon success a SageMakerModel is created and returned from fit. The SageMakerModel is created with a modelImage Docker image URI, defining the SageMaker model primary container and with `modelEnvironmentVariables` environment variables. Each SageMakerModel has a corresponding SageMaker hosting Endpoint. This Endpoint runs on at least endpointInitialInstanceCount instances of type endpointInstanceType. The Endpoint is created either during construction of the SageMakerModel or on the first call to `transform`, controlled by `endpointCreationPolicy`. Each Endpointinstance runs with sagemakerRole IAMRole.

The transform method on SageMakerModel uses `requestRowSerializer` to serialize Rows from the Dataset undergoing transformation, to requests on the hosted SageMaker Endpoint. The `responseRowDeserializer` is used to convert the response from the Endpoint to a series of Rows, forming the transformed Dataset. If `modelPrependInputRowsToTransformationRows` is true, then each transformed Row is also prepended with its corresponding input Row.

Parameters

- **trainingImage** (*String*) – A SageMaker Training Job Algorithm Specification Training Image Docker image URI.
- **modelImage** (*String*) – A SageMaker Model hosting Docker image URI.
- **sageMakerRole** (IAMRole) – The SageMaker TrainingJob and Hosting IAM Role. Used by SageMaker to access S3 and ECR Resources. SageMaker hosted Endpoint instances launched by this Estimator run with this role.
- **trainingInstanceType** (*str*) – The SageMaker TrainingJob Instance Type to use.
- **trainingInstanceCount** (*int*) – The number of instances of instanceType to run an SageMaker Training Job with.
- **endpointInstanceType** (*str*) – The SageMaker Endpoint Config instance type.

- **endpointInitialInstanceCount** (*int*) – The SageMaker Endpoint Config minimum number of instances that can be used to host modelImage.
- **requestRowSerializer** (RequestRowSerializer) – Serializes Spark DataFrame Rows for transformation by Models built from this Estimator.
- **responseRowDeserializer** (ResponseRowDeserializer) – Deserializes an Endpoint response into a series of Rows.
- **hyperParameters** (*dict*) – A dict from hyperParameter names to their respective values for training.
- **trainingInputS3DataPath** (S3Resource) – An S3 location to upload SageMaker Training Job input data to.
- **trainingOutputS3DataPath** (S3Resource) – An S3 location for SageMaker to store Training Job output data to.
- **trainingInstanceVolumeSizeInGB** (*int*) – The EBS volume size in gigabytes of each instance.
- **trainingProjectedColumns** (*List*) – The columns to project from the Dataset being fit before training. If an Optional.empty is passed then no specific projection will occur and all columns will be serialized.
- **trainingChannelName** (*str*) – The SageMaker Channel name to input serialized Dataset fit input to.
- **trainingContentType** (*str*) – The MIME type of the training data.
- **trainingS3DataDistribution** (*str*) – The SageMaker Training Job S3 data distribution scheme.
- **trainingSparkDataFormat** (*str*) – The Spark Data Format name used to serialize the Dataset being fit for input to SageMaker.
- **trainingSparkDataFormatOptions** (*dict*) – The Spark Data Format Options used during serialization of the Dataset being fit.
- **trainingInputMode** (*str*) – The SageMaker Training Job Channel input mode.
- **trainingCompressionCodec** (*str*) – The type of compression to use when serializing the Dataset being fit for input to SageMaker.
- **trainingMaxRuntimeInSeconds** (*int*) – A SageMaker Training Job Termination Condition MaxRuntimeInHours.
- **trainingKmsKeyId** (*str*) – A KMS key ID for the Output Data Source.
- **modelEnvironmentVariables** (*dict*) – The environment variables that SageMaker will set on the model container during execution.
- **endpointCreationPolicy** (EndpointCreationPolicy) – Defines how a SageMaker Endpoint referenced by a SageMakerModel is created.
- **sagemakerClient** (*AmazonSageMaker*) – CreateModel, and CreateEndpoint requests.
- **s3Client** (*AmazonS3*) – Used to create a bucket for staging SageMaker Training Job input and/or output if either are set to S3AutoCreatePath.
- **stsClient** (*AmazonSTS*) – Used to resolve the account number when creating staging input / output buckets.

- **modelPrependInputRowsToTransformationRows** (*bool*) – Whether the transformation result on Models built by this Estimator should also include the input Rows. If true, each output Row is formed by a concatenation of the input Row with the corresponding Row produced by SageMaker Endpoint invocation, produced by responseRowDeserializer. If false, each output Row is just taken from responseRowDeserializer.
- **deleteStagingDataAfterTraining** (*bool*) – Whether to remove the training data on s3 after training is complete or failed.
- **namePolicyFactory** (*NamePolicyFactory*) – The NamePolicyFactory to use when naming SageMaker entities created during fit.
- **uid** (*str*) – The unique identifier of this Estimator. Used to represent this stage in Spark ML pipelines.

copy(*extra*)

Creates a copy of this instance with the same uid and some extra params. This implementation first calls Params.copy and then make a copy of the companion Java pipeline component with extra params. So both the Python wrapper and the Java pipeline component get copied.

Parameters **extra** – Extra parameters to copy to the new instance

Returns Copy of this instance

explainParam(*param*)

Explains a single param and returns its name, doc, and optional default value and user-supplied value in a string.

explainParams()

Returns the documentation of all params with their optionally default values and user-supplied values.

extractParamMap(*extra=None*)

Extracts the embedded default param values and user-supplied values, and then merges them with extra values from input into a flat param map, where the latter value is used if there exist conflicts, i.e., with ordering: default param values < user-supplied values < extra.

Parameters **extra** – extra param values

Returns merged param map

fit(*dataset*)

Fits a SageMakerModel on dataset by running a SageMaker training job.

Parameters **dataset** (*Dataset*) – the dataset to use for the training job.

Returns The Model created by the training job.

Return type JavaSageMakerModel

getOrDefault(*param*)

Gets the value of a param in the user-supplied param map or its default value. Raises an error if neither is set.

getParam(*paramName*)

Gets a param by its name.

hasDefault(*param*)

Checks whether a param has a default value.

hasParam(*paramName*)

Tests whether this instance contains a param with a given (string) name.

isDefined(*param*)

Checks whether a param is explicitly set by user or has a default value.

isSet(*param*)
Checks whether a param is explicitly set by user.

params
Returns all params ordered by name. The default implementation uses `dir()` to get all attributes of type `Param`.

3.1.3 Algorithms

K Means

class KMeansSageMakerEstimator(*trainingInstanceType*, *trainingInstanceCount*, *endpointInstanceType*, *endpointInitialInstanceCount*, *sagemakerRole=<sagemaker_pyspark.IAMRoleResource.IAMRoleFromConfig object>*, *requestRowSerializer=<sagemaker_pyspark.transformation.serializers.serialize object>*, *responseRowDeserializer=<sagemaker_pyspark.transformation.deserializers.deserializers.KMeansProtobufR object>*, *trainingInputS3DataPath=<sagemaker_pyspark.S3Resources.S3AutoCreatePat object>*, *trainingOutputS3DataPath=<sagemaker_pyspark.S3Resources.S3AutoCreateP object>*, *trainingInstanceVolumeSizeInGB=1024*, *trainingProjectedColumns=None*, *trainingChannelName='train'*, *trainingContentType=None*, *trainingS3DataDistribution='ShardedByS3Key'*, *trainingSparkDataFormat='sagemaker'*, *trainingSparkDataFormatOptions=None*, *trainingInputMode='File'*, *trainingCompressionCodec=None*, *trainingMaxRuntimeInSeconds=86400*, *trainingKmsKeyId=None*, *modelEnvironmentVariables=None*, *endpointCreationPolicy=<sagemaker_pyspark.SageMakerEstimator.EndpointCreationPolicy._CreateOnCons object>*, *sagemakerClient=<sagemaker_pyspark.SageMakerClients.SageMakerClients._ object>*, *region=None*, *s3Client=<sagemaker_pyspark.SageMakerClients.SageMakerCli object>*, *stsClient=<sagemaker_pyspark.SageMakerClients.SageMakerClients._STSDefa object>*, *modelPrependInputRowsToTransformationRows=True*, *deleteStagingDataAfterTraining=True*, *namePolicyFactory=<sagemaker_pyspark.NamePolicy.RandomNamePolicyFactory object>*, *uid=None*)

Bases: `sagemaker_pyspark.SageMakerEstimator.SageMakerEstimatorBase`

A *SageMakerEstimator* that runs a KMeans training job on Amazon SageMaker upon a call to *fit()* and returns a *SageMakerModel* that can be used to transform a DataFrame using the hosted K-Means model. K-Means Clustering is useful for grouping similar examples in your dataset.

Amazon SageMaker K-Means clustering trains on RecordIO-encoded Amazon Record protobuf data. SageMaker pyspark writes a DataFrame to S3 by selecting a column of Vectors named "features" and, if present, a column of Doubles named "label". These names are configurable by passing a dictionary with entries in *trainingSparkDataFormatOptions* with key "labelColumnName" or "featuresColumnName", with values corresponding to the desired label and features columns.

For inference, the SageMakerModel returned by *fit()* by the KMeansSageMakerEstimator uses *ProtobufRequestRowSerializer* to serialize Rows into RecordIO-encoded Amazon Record protobuf messages for inference, by default selecting the column named "features" expected to contain a Vector of Doubles.

Inferences made against an Endpoint hosting a K-Means model contain a "closest_cluster" field and a "distance_to_cluster" field, both appended to the input DataFrame as columns of Double.

Parameters

- **sageMakerRole** (IAMRole) – The SageMaker TrainingJob and Hosting IAM Role. Used by SageMaker to access S3 and ECR Resources. SageMaker hosted Endpoint instances launched by this Estimator run with this role.
- **trainingInstanceType** (*str*) – The SageMaker TrainingJob Instance Type to use.
- **trainingInstanceCount** (*int*) – The number of instances of instanceType to run an SageMaker Training Job with.
- **endpointInstanceType** (*str*) – The SageMaker Endpoint Config instance type.
- **endpointInitialInstanceCount** (*int*) – The SageMaker Endpoint Config minimum number of instances that can be used to host modelImage.
- **requestRowSerializer** (RequestRowSerializer) – Serializes Spark DataFrame Rows for transformation by Models built from this Estimator.
- **responseRowDeserializer** (ResponseRowDeserializer) – Deserializes an Endpoint response into a series of Rows.
- **trainingInputS3DataPath** (S3Resource) – An S3 location to upload SageMaker Training Job input data to.
- **trainingOutputS3DataPath** (S3Resource) – An S3 location for SageMaker to store Training Job output data to.
- **trainingInstanceVolumeSizeInGB** (*int*) – The EBS volume size in gigabytes of each instance.
- **trainingProjectedColumns** (*List*) – The columns to project from the Dataset being fit before training. If an Optional.empty is passed then no specific projection will occur and all columns will be serialized.
- **trainingChannelName** (*str*) – The SageMaker Channel name to input serialized Dataset fit input to.
- **trainingContentType** (*str*) – The MIME type of the training data.
- **trainingS3DataDistribution** (*str*) – The SageMaker Training Job S3 data distribution scheme.
- **trainingSparkDataFormat** (*str*) – The Spark Data Format name used to serialize the Dataset being fit for input to SageMaker.
- **trainingSparkDataFormatOptions** (*dict*) – The Spark Data Format Options used during serialization of the Dataset being fit.
- **trainingInputMode** (*str*) – The SageMaker Training Job Channel input mode.
- **trainingCompressionCodec** (*str*) – The type of compression to use when serializing the Dataset being fit for input to SageMaker.
- **trainingMaxRuntimeInSeconds** (*int*) – A SageMaker Training Job Termination Condition MaxRuntimeInHours.
- **trainingKmsKeyId** (*str*) – A KMS key ID for the Output Data Source.
- **modelEnvironmentVariables** (*dict*) – The environment variables that SageMaker will set on the model container during execution.
- **endpointCreationPolicy** (EndpointCreationPolicy) – Defines how a SageMaker Endpoint referenced by a SageMakerModel is created.

- **sagemakerClient** (*AmazonSageMaker*) – CreateModel, and CreateEndpoint requests.
- **region** (*str*) – The region in which to run the algorithm. If not specified, gets the region from the DefaultAwsRegionProviderChain.
- **s3Client** (*AmazonS3*) – Used to create a bucket for staging SageMaker Training Job input and/or output if either are set to S3AutoCreatePath.
- **stsClient** (*AmazonSTS*) – Used to resolve the account number when creating staging input / output buckets.
- **modelPrependInputRowsToTransformationRows** (*bool*) – Whether the transformation result on Models built by this Estimator should also include the input Rows. If true, each output Row is formed by a concatenation of the input Row with the corresponding Row produced by SageMaker Endpoint invocation, produced by responseRowDeserializer. If false, each output Row is just taken from responseRowDeserializer.
- **deleteStagingDataAfterTraining** (*bool*) – Whether to remove the training data on s3 after training is complete or failed.
- **namePolicyFactory** (*NamePolicyFactory*) – The NamePolicyFactory to use when naming SageMaker entities created during fit.
- **uid** (*str*) – The unique identifier of this Estimator. Used to represent this stage in Spark ML pipelines.

copy(*extra*)

Creates a copy of this instance with the same uid and some extra params. This implementation first calls Params.copy and then make a copy of the companion Java pipeline component with extra params. So both the Python wrapper and the Java pipeline component get copied.

Parameters **extra** – Extra parameters to copy to the new instance

Returns Copy of this instance

explainParam(*param*)

Explains a single param and returns its name, doc, and optional default value and user-supplied value in a string.

explainParams()

Returns the documentation of all params with their optionally default values and user-supplied values.

extractParamMap(*extra=None*)

Extracts the embedded default param values and user-supplied values, and then merges them with extra values from input into a flat param map, where the latter value is used if there exist conflicts, i.e., with ordering: default param values < user-supplied values < extra.

Parameters **extra** – extra param values

Returns merged param map

fit(*dataset*)

Fits a SageMakerModel on dataset by running a SageMaker training job.

Parameters **dataset** (*Dataset*) – the dataset to use for the training job.

Returns The Model created by the training job.

Return type JavaSageMakerModel

getOrDefault(*param*)

Gets the value of a param in the user-supplied param map or its default value. Raises an error if neither is set.

getParam(*paramName*)
Gets a param by its name.

hasDefault(*param*)
Checks whether a param has a default value.

hasParam(*paramName*)
Tests whether this instance contains a param with a given (string) name.

isDefined(*param*)
Checks whether a param is explicitly set by user or has a default value.

isSet(*param*)
Checks whether a param is explicitly set by user.

params
Returns all params ordered by name. The default implementation uses `dir()` to get all attributes of type `Param`.

Linear Learner Regressor

class LinearLearnerRegressor(*trainingInstanceType*, *trainingInstanceCount*, *endpointInstanceType*, *endpointInitialInstanceCount*, *sagemakerRole=<sagemaker_pyspark.IAMRoleResource.IAMRoleFromConfig object>*, *requestRowSerializer=<sagemaker_pyspark.transformation.serializers.serializers. object>*, *responseRowDeserializer=<sagemaker_pyspark.transformation.deserializers.dese object>*, *trainingInputS3DataPath=<sagemaker_pyspark.S3Resources.S3AutoCreatePath object>*, *trainingOutputS3DataPath=<sagemaker_pyspark.S3Resources.S3AutoCreatePath object>*, *trainingInstanceVolumeSizeInGB=1024*, *trainingProjectedColumns=None*, *trainingChannelName='train'*, *trainingContentType=None*, *trainingS3DataDistribution='ShardedByS3Key'*, *trainingSparkDataFormat='sagemaker'*, *trainingSparkDataFormatOptions=None*, *trainingInputMode='File'*, *trainingCompressionCodec=None*, *trainingMaxRuntimeInSeconds=86400*, *trainingKmsKeyId=None*, *modelEnvironmentVariables=None*, *endpointCreationPolicy=<sagemaker_pyspark.SageMakerEstimator.EndpointCreationPolicy._CreateOnConstru object>*, *sagemakerClient=<sagemaker_pyspark.SageMakerClients.SageMakerClients._Sag object>*, *region=None*, *s3Client=<sagemaker_pyspark.SageMakerClients.SageMakerClient object>*, *stsClient=<sagemaker_pyspark.SageMakerClients.SageMakerClients._STSDefault object>*, *modelPrependInputRowsToTransformationRows=True*, *deleteStagingDataAfterTraining=True*, *namePolicyFactory=<sagemaker_pyspark.NamePolicy.RandomNamePolicyFactory object>*, *uid=None*, *javaObject=None*)

Bases: `sagemaker_pyspark.SageMakerEstimator.SageMakerEstimatorBase`, `sagemaker_pyspark.algorithms.LinearLearnerSageMakerEstimator.LinearLearnerParams`

A *SageMakerEstimator* that runs a Linear Learner training job in "regressor" mode in SageMaker and returns a *SageMakerModel* that can be used to transform a DataFrame using the hosted Linear Learner model. The Linear Learner Regressor is useful for predicting a real-valued label from training examples.

Amazon SageMaker Linear Learner trains on RecordIO-encoded Amazon Record protobuf data. SageMaker pyspark writes a DataFrame to S3 by selecting a column of Vectors named "features" and, if present, a column of Doubles named "label". These names are configurable by passing a dictionary with entries in trainingSparkDataFormatOptions with key "labelColumnName" or "featuresColumnName", with values corresponding to the desired label and features columns.

For inference against a hosted Endpoint, the SageMakerModel returned by :meth *:fit()* by Linear Learner uses `ProtobufRequestRowSerializer` to serialize Rows into RecordIO-encoded Amazon Record protobuf messages, by default selecting the column named "features" expected to contain a Vector of Doubles.

Inferences made against an Endpoint hosting a Linear Learner Regressor model contain a "score" field appended to the input DataFrame as a Double.

Parameters

- **sageMakerRole** (`IAMRole`) – The SageMaker TrainingJob and Hosting IAM Role. Used by SageMaker to access S3 and ECR Resources. SageMaker hosted Endpoint instances launched by this Estimator run with this role.
- **trainingInstanceType** (*str*) – The SageMaker TrainingJob Instance Type to use.
- **trainingInstanceCount** (*int*) – The number of instances of instanceType to run an SageMaker Training Job with.
- **endpointInstanceType** (*str*) – The SageMaker Endpoint Config instance type.
- **endpointInitialInstanceCount** (*int*) – The SageMaker Endpoint Config minimum number of instances that can be used to host modelImage.
- **requestRowSerializer** (`RequestRowSerializer`) – Serializes Spark DataFrame Rows for transformation by Models built from this Estimator.
- **responseRowDeserializer** (`ResponseRowDeserializer`) – Deserializes an Endpoint response into a series of Rows.
- **trainingInputS3DataPath** (`S3Resource`) – An S3 location to upload SageMaker Training Job input data to.
- **trainingOutputS3DataPath** (`S3Resource`) – An S3 location for SageMaker to store Training Job output data to.
- **trainingInstanceVolumeSizeInGB** (*int*) – The EBS volume size in gigabytes of each instance.
- **trainingProjectedColumns** (*List*) – The columns to project from the Dataset being fit before training. If an Optional.empty is passed then no specific projection will occur and all columns will be serialized.
- **trainingChannelName** (*str*) – The SageMaker Channel name to input serialized Dataset fit input to.
- **trainingContentType** (*str*) – The MIME type of the training data.
- **trainingS3DataDistribution** (*str*) – The SageMaker Training Job S3 data distribution scheme.
- **trainingSparkDataFormat** (*str*) – The Spark Data Format name used to serialize the Dataset being fit for input to SageMaker.
- **trainingSparkDataFormatOptions** (*dict*) – The Spark Data Format Options used during serialization of the Dataset being fit.
- **trainingInputMode** (*str*) – The SageMaker Training Job Channel input mode.
- **trainingCompressionCodec** (*str*) – The type of compression to use when serializing the Dataset being fit for input to SageMaker.
- **trainingMaxRuntimeInSeconds** (*int*) – A SageMaker Training Job Termination Condition MaxRuntimeInHours.
- **trainingKmsKeyId** (*str*) – A KMS key ID for the Output Data Source.

- **modelEnvironmentVariables** (*dict*) – The environment variables that SageMaker will set on the model container during execution.
- **endpointCreationPolicy** (EndpointCreationPolicy) – Defines how a SageMaker Endpoint referenced by a SageMakerModel is created.
- **sagemakerClient** (*AmazonSageMaker*) – CreateModel, and CreateEndpoint requests.
- **region** (*str*) – The region in which to run the algorithm. If not specified, gets the region from the DefaultAwsRegionProviderChain.
- **s3Client** (*AmazonS3*) – Used to create a bucket for staging SageMaker Training Job input and/or output if either are set to S3AutoCreatePath.
- **stsClient** (*AmazonSTS*) – Used to resolve the account number when creating staging input / output buckets.
- **modelPrependInputRowsToTransformationRows** (*bool*) – Whether the transformation result on Models built by this Estimator should also include the input Rows. If true, each output Row is formed by a concatenation of the input Row with the corresponding Row produced by SageMaker Endpoint invocation, produced by responseRowDeserializer. If false, each output Row is just taken from responseRowDeserializer.
- **deleteStagingDataAfterTraining** (*bool*) – Whether to remove the training data on s3 after training is complete or failed.
- **namePolicyFactory** (*NamePolicyFactory*) – The NamePolicyFactory to use when naming SageMaker entities created during fit.
- **uid** (*str*) – The unique identifier of this Estimator. Used to represent this stage in Spark ML pipelines.

copy(*extra*)

Creates a copy of this instance with the same uid and some extra params. This implementation first calls Params.copy and then make a copy of the companion Java pipeline component with extra params. So both the Python wrapper and the Java pipeline component get copied.

Parameters **extra** – Extra parameters to copy to the new instance

Returns Copy of this instance

explainParam(*param*)

Explains a single param and returns its name, doc, and optional default value and user-supplied value in a string.

explainParams()

Returns the documentation of all params with their optionally default values and user-supplied values.

extractParamMap(*extra=None*)

Extracts the embedded default param values and user-supplied values, and then merges them with extra values from input into a flat param map, where the latter value is used if there exist conflicts, i.e., with ordering: default param values < user-supplied values < extra.

Parameters **extra** – extra param values

Returns merged param map

fit(*dataset*)

Fits a SageMakerModel on dataset by running a SageMaker training job.

Parameters **dataset** (*Dataset*) – the dataset to use for the training job.

Returns The Model created by the training job.

Return type JavaSageMakerModel

getOrDefault(*param*)
Gets the value of a param in the user-supplied param map or its default value. Raises an error if neither is set.

getParam(*paramName*)
Gets a param by its name.

hasDefault(*param*)
Checks whether a param has a default value.

hasParam(*paramName*)
Tests whether this instance contains a param with a given (string) name.

isDefined(*param*)
Checks whether a param is explicitly set by user or has a default value.

isSet(*param*)
Checks whether a param is explicitly set by user.

params
Returns all params ordered by name. The default implementation uses `dir()` to get all attributes of type `Param`.

Linear Learner Binary Classifier

class LinearLearnerBinaryClassifier(*trainingInstanceType*, *trainingInstanceCount*, *endpointInstanceType*, *endpointInitialInstanceCount*, *sagemakerRole=<sagemaker_pyspark.IAMRoleResource.IAMRoleFromConfig object>*, *requestRowSerializer=<sagemaker_pyspark.transformation.serializers.serializers.ProtobufReques object>*, *responseRowDeserializer=<sagemaker_pyspark.transformation.deserializers.deserializers.LinearLear object>*, *trainingInputS3DataPath=<sagemaker_pyspark.S3Resources.S3AutoCr object>*, *trainingOutputS3DataPath=<sagemaker_pyspark.S3Resources.S3AutoC object>*, *trainingInstanceVolumeSizeInGB=1024*, *trainingProjectedColumns=None*, *trainingChannelName='train'*, *trainingContentType=None*, *trainingS3DataDistribution='ShardedByS3Key'*, *trainingSparkDataFormat='sagemaker'*, *trainingSparkDataFormatOptions=None*, *trainingInputMode='File'*, *trainingCompressionCodec=None*, *trainingMaxRuntimeInSeconds=86400*, *trainingKmsKeyId=None*, *modelEnvironmentVariables=None*, *endpointCreationPolicy=<sagemaker_pyspark.SageMakerEstimator.EndpointCreationPolicy._Create object>*, *sagemakerClient=<sagemaker_pyspark.SageMakerClients.SageMakerC object>*, *region=None*, *s3Client=<sagemaker_pyspark.SageMakerClients.SageMakerClients._S3Default object>*, *stsClient=<sagemaker_pyspark.SageMakerClients.SageMakerClients._S object>*, *modelPrependInputRowsToTransformationRows=True*, *deleteStagingDataAfterTraining=True*, *namePolicyFactory=<sagemaker_pyspark.NamePolicy.RandomNamePolicyFactory object>*, *uid=None*, *javaObject=None*)

Bases: `sagemaker_pyspark.SageMakerEstimator.SageMakerEstimatorBase`,

sagemaker_pyspark.algorithms.LinearLearnerSageMakerEstimator.
LinearLearnerParams

A *SageMakerEstimator* that runs a Linear Learner training job in "binary classifier" mode in SageMaker and returns a *SageMakerModel* that can be used to transform a DataFrame using the hosted Linear Learner model. The Linear Learner Binary Classifier is useful for classifying examples into one of two classes.

Amazon SageMaker Linear Learner trains on RecordIO-encoded Amazon Record protobuf data. SageMaker pyspark writes a DataFrame to S3 by selecting a column of Vectors named "features" and, if present, a column of Doubles named "label". These names are configurable by passing a dictionary with entries in trainingSparkDataFormatOptions with key "labelColumnName" or "featuresColumnName", with values corresponding to the desired label and features columns.

Inferences made against an Endpoint hosting a Linear Learner Binary classifier model contain a "score" field and a "predicted_label" field, both appended to the input DataFrame as Doubles.

Parameters

- **sageMakerRole** (IAMRole) – The SageMaker TrainingJob and Hosting IAM Role. Used by SageMaker to access S3 and ECR Resources. SageMaker hosted Endpoint instances launched by this Estimator run with this role.
- **trainingInstanceType** (*str*) – The SageMaker TrainingJob Instance Type to use.
- **trainingInstanceCount** (*int*) – The number of instances of instanceType to run an SageMaker Training Job with.
- **endpointInstanceType** (*str*) – The SageMaker Endpoint Config instance type.
- **endpointInitialInstanceCount** (*int*) – The SageMaker Endpoint Config minimum number of instances that can be used to host modelImage.
- **requestRowSerializer** (RequestRowSerializer) – Serializes Spark DataFrame Rows for transformation by Models built from this Estimator.
- **responseRowDeserializer** (ResponseRowDeserializer) – Deserializes an Endpoint response into a series of Rows.
- **trainingInputS3DataPath** (S3Resource) – An S3 location to upload SageMaker Training Job input data to.
- **trainingOutputS3DataPath** (S3Resource) – An S3 location for SageMaker to store Training Job output data to.
- **trainingInstanceVolumeSizeInGB** (*int*) – The EBS volume size in gigabytes of each instance.
- **trainingProjectedColumns** (*List*) – The columns to project from the Dataset being fit before training. If an Optional.empty is passed then no specific projection will occur and all columns will be serialized.
- **trainingChannelName** (*str*) – The SageMaker Channel name to input serialized Dataset fit input to.
- **trainingContentType** (*str*) – The MIME type of the training data.
- **trainingS3DataDistribution** (*str*) – The SageMaker Training Job S3 data distribution scheme.
- **trainingSparkDataFormat** (*str*) – The Spark Data Format name used to serialize the Dataset being fit for input to SageMaker.
- **trainingSparkDataFormatOptions** (*dict*) – The Spark Data Format Options used during serialization of the Dataset being fit.

- **trainingInputMode** (*str*) – The SageMaker Training Job Channel input mode.
- **trainingCompressionCodec** (*str*) – The type of compression to use when serializing the Dataset being fit for input to SageMaker.
- **trainingMaxRuntimeInSeconds** (*int*) – A SageMaker Training Job Termination Condition MaxRuntimeInHours.
- **trainingKmsKeyId** (*str*) – A KMS key ID for the Output Data Source.
- **modelEnvironmentVariables** (*dict*) – The environment variables that SageMaker will set on the model container during execution.
- **endpointCreationPolicy** (EndpointCreationPolicy) – Defines how a SageMaker Endpoint referenced by a SageMakerModel is created.
- **sagemakerClient** (*AmazonSageMaker*) – CreateModel, and CreateEndpoint requests.
- **region** (*str*) – The region in which to run the algorithm. If not specified, gets the region from the DefaultAwsRegionProviderChain.
- **s3Client** (*AmazonS3*) – Used to create a bucket for staging SageMaker Training Job input and/or output if either are set to S3AutoCreatePath.
- **stsClient** (*AmazonSTS*) – Used to resolve the account number when creating staging input / output buckets.
- **modelPrependInputRowsToTransformationRows** (*bool*) – Whether the transformation result on Models built by this Estimator should also include the input Rows. If true, each output Row is formed by a concatenation of the input Row with the corresponding Row produced by SageMaker Endpoint invocation, produced by responseRowDeserializer. If false, each output Row is just taken from responseRowDeserializer.
- **deleteStagingDataAfterTraining** (*bool*) – Whether to remove the training data on s3 after training is complete or failed.
- **namePolicyFactory** (*NamePolicyFactory*) – The NamePolicyFactory to use when naming SageMaker entities created during fit.
- **uid** (*str*) – The unique identifier of this Estimator. Used to represent this stage in Spark ML pipelines.

copy(*extra*)

Creates a copy of this instance with the same uid and some extra params. This implementation first calls Params.copy and then make a copy of the companion Java pipeline component with extra params. So both the Python wrapper and the Java pipeline component get copied.

Parameters **extra** – Extra parameters to copy to the new instance

Returns Copy of this instance

explainParam(*param*)

Explains a single param and returns its name, doc, and optional default value and user-supplied value in a string.

explainParams()

Returns the documentation of all params with their optionally default values and user-supplied values.

extractParamMap(*extra=None*)

Extracts the embedded default param values and user-supplied values, and then merges them with extra values from input into a flat param map, where the latter value is used if there exist conflicts, i.e., with ordering: default param values < user-supplied values < extra.

Parameters **extra** – extra param values

Returns merged param map

fit(*dataset*)
Fits a SageMakerModel on dataset by running a SageMaker training job.

Parameters **dataset** (*Dataset*) – the dataset to use for the training job.

Returns The Model created by the training job.

Return type JavaSageMakerModel

getOrDefault(*param*)
Gets the value of a param in the user-supplied param map or its default value. Raises an error if neither is set.

getParam(*paramName*)
Gets a param by its name.

hasDefault(*param*)
Checks whether a param has a default value.

hasParam(*paramName*)
Tests whether this instance contains a param with a given (string) name.

isDefined(*param*)
Checks whether a param is explicitly set by user or has a default value.

isSet(*param*)
Checks whether a param is explicitly set by user.

params
Returns all params ordered by name. The default implementation uses `dir()` to get all attributes of type `Param`.

PCA

class PCASageMakerEstimator(*trainingInstanceType*, *trainingInstanceCount*, *endpointInstanceType*, *endpointInitialInstanceCount*, *sagemakerRole=<sagemaker_pyspark.IAMRoleResource.IAMRoleFromConfig object>*, *requestRowSerializer=<sagemaker_pyspark.transformation.serializers.serializers.P object>*, *responseRowDeserializer=<sagemaker_pyspark.transformation.deserializers.deseri object>*, *trainingInputS3DataPath=<sagemaker_pyspark.S3Resources.S3AutoCreatePath object>*, *trainingOutputS3DataPath=<sagemaker_pyspark.S3Resources.S3AutoCreatePath object>*, *trainingInstanceVolumeSizeInGB=1024*, *trainingProjectedColumns=None*, *trainingChannelName='train'*, *trainingContentType=None*, *trainingS3DataDistribution='ShardedByS3Key'*, *trainingSparkDataFormat='sagemaker'*, *trainingSparkDataFormatOptions=None*, *trainingInputMode='File'*, *trainingCompressionCodec=None*, *trainingMaxRuntimeInSeconds=86400*, *trainingKmsKeyId=None*, *modelEnvironmentVariables=None*, *endpointCreationPolicy=<sagemaker_pyspark.SageMakerEstimator.EndpointCreationPolicy._CreateOnConstruct object>*, *sagemakerClient=<sagemaker_pyspark.SageMakerClients.SageMakerClients._Sage object>*, *region=None*, *s3Client=<sagemaker_pyspark.SageMakerClients.SageMakerClients. object>*, *stsClient=<sagemaker_pyspark.SageMakerClients.SageMakerClients._STSDefaultC object>*, *modelPrependInputRowsToTransformationRows=True*, *deleteStagingDataAfterTraining=True*, *namePolicyFactory=<sagemaker_pyspark.NamePolicy.RandomNamePolicyFactory object>*, *uid=None*)

Bases: `sagemaker_pyspark.SageMakerEstimator.SageMakerEstimatorBase`

A *SageMakerEstimator* that runs a PCA training job in SageMaker and returns a *SageMakerModel* that can be used to transform a DataFrame using the hosted PCA model. PCA, or Principal Component Analysis, is useful for reducing the dimensionality of data before training with another algorithm.

Amazon SageMaker PCA trains on RecordIO-encoded Amazon Record protobuf data. SageMaker pyspark writes a DataFrame to S3 by selecting a column of Vectors named "features" and, if present, a column of Doubles named "label". These names are configurable by passing a dictionary with entries in trainingSparkDataFormatOptions with key "labelColumnName" or "featuresColumnName", with values corresponding to the desired label and features columns.

PCASageMakerEstimator uses *ProtobufRequestRowSerializer* to serialize

Rows into RecordIO-encoded Amazon Record protobuf messages for inference, by default selecting

the column named "features" expected to contain a Vector of Doubles.

Inferences made against an Endpoint hosting a PCA model contain a "projection" field appended to the input DataFrame as a Dense Vector of Doubles.

Parameters

- **sageMakerRole** (*IAMRole*) – The SageMaker TrainingJob and Hosting IAM Role. Used by SageMaker to access S3 and ECR Resources. SageMaker hosted Endpoint instances launched by this Estimator run with this role.
- **trainingInstanceType** (*str*) – The SageMaker TrainingJob Instance Type to use.
- **trainingInstanceCount** (*int*) – The number of instances of instanceType to run an SageMaker Training Job with.
- **endpointInstanceType** (*str*) – The SageMaker Endpoint Config instance type.

- **endpointInitialInstanceCount** (*int*) – The SageMaker Endpoint Config minimum number of instances that can be used to host modelImage.
- **requestRowSerializer** (RequestRowSerializer) – Serializes Spark DataFrame Rows for transformation by Models built from this Estimator.
- **responseRowDeserializer** (ResponseRowDeserializer) – Deserializes an Endpoint response into a series of Rows.
- **trainingInputS3DataPath** (S3Resource) – An S3 location to upload SageMaker Training Job input data to.
- **trainingOutputS3DataPath** (S3Resource) – An S3 location for SageMaker to store Training Job output data to.
- **trainingInstanceVolumeSizeInGB** (*int*) – The EBS volume size in gigabytes of each instance.
- **trainingProjectedColumns** (*List*) – The columns to project from the Dataset being fit before training. If an Optional.empty is passed then no specific projection will occur and all columns will be serialized.
- **trainingChannelName** (*str*) – The SageMaker Channel name to input serialized Dataset fit input to.
- **trainingContentType** (*str*) – The MIME type of the training data.
- **trainingS3DataDistribution** (*str*) – The SageMaker Training Job S3 data distribution scheme.
- **trainingSparkDataFormat** (*str*) – The Spark Data Format name used to serialize the Dataset being fit for input to SageMaker.
- **trainingSparkDataFormatOptions** (*dict*) – The Spark Data Format Options used during serialization of the Dataset being fit.
- **trainingInputMode** (*str*) – The SageMaker Training Job Channel input mode.
- **trainingCompressionCodec** (*str*) – The type of compression to use when serializing the Dataset being fit for input to SageMaker.
- **trainingMaxRuntimeInSeconds** (*int*) – A SageMaker Training Job Termination Condition MaxRuntimeInHours.
- **trainingKmsKeyId** (*str*) – A KMS key ID for the Output Data Source.
- **modelEnvironmentVariables** (*dict*) – The environment variables that SageMaker will set on the model container during execution.
- **endpointCreationPolicy** (EndpointCreationPolicy) – Defines how a SageMaker Endpoint referenced by a SageMakerModel is created.
- **sagemakerClient** (*AmazonSageMaker*) – CreateModel, and CreateEndpoint requests.
- **region** (*str*) – The region in which to run the algorithm. If not specified, gets the region from the DefaultAwsRegionProviderChain.
- **s3Client** (*AmazonS3*) – Used to create a bucket for staging SageMaker Training Job input and/or output if either are set to S3AutoCreatePath.
- **stsClient** (*AmazonSTS*) – Used to resolve the account number when creating staging input / output buckets.

- **modelPrependInputRowsToTransformationRows** (*bool*) – Whether the transformation result on Models built by this Estimator should also include the input Rows. If true, each output Row is formed by a concatenation of the input Row with the corresponding Row produced by SageMaker Endpoint invocation, produced by responseRowDeserializer. If false, each output Row is just taken from responseRowDeserializer.
- **deleteStagingDataAfterTraining** (*bool*) – Whether to remove the training data on s3 after training is complete or failed.
- **namePolicyFactory** (*NamePolicyFactory*) – The NamePolicyFactory to use when naming SageMaker entities created during fit.
- **uid** (*str*) – The unique identifier of this Estimator. Used to represent this stage in Spark ML pipelines.

copy(*extra*)

Creates a copy of this instance with the same uid and some extra params. This implementation first calls Params.copy and then make a copy of the companion Java pipeline component with extra params. So both the Python wrapper and the Java pipeline component get copied.

Parameters **extra** – Extra parameters to copy to the new instance

Returns Copy of this instance

explainParam(*param*)

Explains a single param and returns its name, doc, and optional default value and user-supplied value in a string.

explainParams()

Returns the documentation of all params with their optionally default values and user-supplied values.

extractParamMap(*extra=None*)

Extracts the embedded default param values and user-supplied values, and then merges them with extra values from input into a flat param map, where the latter value is used if there exist conflicts, i.e., with ordering: default param values < user-supplied values < extra.

Parameters **extra** – extra param values

Returns merged param map

fit(*dataset*)

Fits a SageMakerModel on dataset by running a SageMaker training job.

Parameters **dataset** (*Dataset*) – the dataset to use for the training job.

Returns The Model created by the training job.

Return type JavaSageMakerModel

getOrDefault(*param*)

Gets the value of a param in the user-supplied param map or its default value. Raises an error if neither is set.

getParam(*paramName*)

Gets a param by its name.

hasDefault(*param*)

Checks whether a param has a default value.

hasParam(*paramName*)

Tests whether this instance contains a param with a given (string) name.

isDefined(*param*)

Checks whether a param is explicitly set by user or has a default value.

isSet(*param*)
Checks whether a param is explicitly set by user.

params
Returns all params ordered by name. The default implementation uses `dir()` to get all attributes of type `Param`.

XGBoost

class XGBoostSageMakerEstimator(*trainingInstanceType*, *trainingInstanceCount*, *endpointInstanceType*, *endpointInitialInstanceCount*, *sagemakerRole=<sagemaker_pyspark.IAMRoleResource.IAMRoleFromConfig object>*, *requestRowSerializer=<sagemaker_pyspark.transformation.serializers.seriali object>*, *responseRowDeserializer=<sagemaker_pyspark.transformation.deserializers.deserializers.XGBoostCSVRow object>*, *trainingInputS3DataPath=<sagemaker_pyspark.S3Resources.S3AutoCreatePa object>*, *trainingOutputS3DataPath=<sagemaker_pyspark.S3Resources.S3AutoCreate object>*, *trainingInstanceVolumeSizeInGB=1024*, *trainingProjectedColumns=None*, *trainingChannelName='train'*, *trainingContentType=None*, *trainingS3DataDistribution='ShardedByS3Key'*, *trainingSparkDataFormat='libsvm'*, *trainingSparkDataFormatOptions=None*, *trainingInputMode='File'*, *trainingCompressionCodec=None*, *trainingMaxRuntimeInSeconds=86400*, *trainingKmsKeyId=None*, *modelEnvironmentVariables=None*, *endpointCreationPolicy=<sagemaker_pyspark.SageMakerEstimator.EndpointCreationPolicy._CreateOnCo object>*, *sagemakerClient=<sagemaker_pyspark.SageMakerClients.SageMakerClients. object>*, *region=None*, *s3Client=<sagemaker_pyspark.SageMakerClients.SageMakerC object>*, *stsClient=<sagemaker_pyspark.SageMakerClients.SageMakerClients._STSDe object>*, *modelPrependInputRowsToTransformationRows=True*, *deleteStagingDataAfterTraining=True*, *namePolicyFactory=<sagemaker_pyspark.NamePolicy.RandomNamePolicyFactory object>*, *uid=None*)

Bases: `sagemaker_pyspark.SageMakerEstimator.SageMakerEstimatorBase`

A *SageMakerEstimator* that runs an XGBoost training job in Amazon SageMaker and returns a *SageMakerModel* that can be used to transform a DataFrame using he hosted XGBoost model. XGBoost is an open-source distributed gradient boosting library that Amazon SageMaker has adapted to run on Amazon SageMaker.

XGBoost trains and infers on LibSVM-formatted data. XGBoostSageMakerEstimator uses Spark's LibSVMFileFormat to write the training DataFrame to S3, and serializes Rows to LibSVM for inference, selecting the column named "features" by default, expected to contain a Vector of Doubles.

Inferences made against an Endpoint hosting an XGBoost model contain a "prediction" field appended to the input DataFrame as a column of Doubles, containing the prediction corresponding to the given Vector of features.

See XGBoost github for more on XGBoost

Parameters

- **sageMakerRole** (IAMRole) – The SageMaker TrainingJob and Hosting IAM Role. Used by SageMaker to access S3 and ECR Resources. SageMaker hosted Endpoint instances launched by this Estimator run with this role.

- **trainingInstanceType** (*str*) – The SageMaker TrainingJob Instance Type to use.
- **trainingInstanceCount** (*int*) – The number of instances of instanceType to run an SageMaker Training Job with.
- **endpointInstanceType** (*str*) – The SageMaker Endpoint Config instance type.
- **endpointInitialInstanceCount** (*int*) – The SageMaker Endpoint Config minimum number of instances that can be used to host modelImage.
- **requestRowSerializer** (RequestRowSerializer) – Serializes Spark DataFrame Rows for transformation by Models built from this Estimator.
- **responseRowDeserializer** (ResponseRowDeserializer) – Deserializes an Endpoint response into a series of Rows.
- **trainingInputS3DataPath** (S3Resource) – An S3 location to upload SageMaker Training Job input data to.
- **trainingOutputS3DataPath** (S3Resource) – An S3 location for SageMaker to store Training Job output data to.
- **trainingInstanceVolumeSizeInGB** (*int*) – The EBS volume size in gigabytes of each instance.
- **trainingProjectedColumns** (*List*) – The columns to project from the Dataset being fit before training. If an Optional.empty is passed then no specific projection will occur and all columns will be serialized.
- **trainingChannelName** (*str*) – The SageMaker Channel name to input serialized Dataset fit input to.
- **trainingContentType** (*str*) – The MIME type of the training data.
- **trainingS3DataDistribution** (*str*) – The SageMaker Training Job S3 data distribution scheme.
- **trainingSparkDataFormat** (*str*) – The Spark Data Format name used to serialize the Dataset being fit for input to SageMaker.
- **trainingSparkDataFormatOptions** (*dict*) – The Spark Data Format Options used during serialization of the Dataset being fit.
- **trainingInputMode** (*str*) – The SageMaker Training Job Channel input mode.
- **trainingCompressionCodec** (*str*) – The type of compression to use when serializing the Dataset being fit for input to SageMaker.
- **trainingMaxRuntimeInSeconds** (*int*) – A SageMaker Training Job Termination Condition MaxRuntimeInHours.
- **trainingKmsKeyId** (*str*) – A KMS key ID for the Output Data Source.
- **modelEnvironmentVariables** (*dict*) – The environment variables that SageMaker will set on the model container during execution.
- **endpointCreationPolicy** (EndpointCreationPolicy) – Defines how a SageMaker Endpoint referenced by a SageMakerModel is created.
- **sagemakerClient** (*AmazonSageMaker*) – CreateModel, and CreateEndpoint requests.
- **region** (*str*) – The region in which to run the algorithm. If not specified, gets the region from the DefaultAwsRegionProviderChain.

- **s3Client** (*AmazonS3*) – Used to create a bucket for staging SageMaker Training Job input and/or output if either are set to S3AutoCreatePath.
- **stsClient** (*AmazonSTS*) – Used to resolve the account number when creating staging input / output buckets.
- **modelPrependInputRowsToTransformationRows** (*bool*) – Whether the transformation result on Models built by this Estimator should also include the input Rows. If true, each output Row is formed by a concatenation of the input Row with the corresponding Row produced by SageMaker Endpoint invocation, produced by responseRowDeserializer. If false, each output Row is just taken from responseRowDeserializer.
- **deleteStagingDataAfterTraining** (*bool*) – Whether to remove the training data on s3 after training is complete or failed.
- **namePolicyFactory** (*NamePolicyFactory*) – The NamePolicyFactory to use when naming SageMaker entities created during fit.
- **uid** (*str*) – The unique identifier of this Estimator. Used to represent this stage in Spark ML pipelines.

copy(*extra*)

Creates a copy of this instance with the same uid and some extra params. This implementation first calls Params.copy and then make a copy of the companion Java pipeline component with extra params. So both the Python wrapper and the Java pipeline component get copied.

Parameters **extra** – Extra parameters to copy to the new instance

Returns Copy of this instance

explainParam(*param*)

Explains a single param and returns its name, doc, and optional default value and user-supplied value in a string.

explainParams()

Returns the documentation of all params with their optionally default values and user-supplied values.

extractParamMap(*extra=None*)

Extracts the embedded default param values and user-supplied values, and then merges them with extra values from input into a flat param map, where the latter value is used if there exist conflicts, i.e., with ordering: default param values < user-supplied values < extra.

Parameters **extra** – extra param values

Returns merged param map

fit(*dataset*)

Fits a SageMakerModel on dataset by running a SageMaker training job.

Parameters **dataset** (*Dataset*) – the dataset to use for the training job.

Returns The Model created by the training job.

Return type JavaSageMakerModel

getOrDefault(*param*)

Gets the value of a param in the user-supplied param map or its default value. Raises an error if neither is set.

getParam(*paramName*)

Gets a param by its name.

hasDefault(*param*)

Checks whether a param has a default value.

hasParam(*paramName*)
Tests whether this instance contains a param with a given (string) name.

isDefined(*param*)
Checks whether a param is explicitly set by user or has a default value.

isSet(*param*)
Checks whether a param is explicitly set by user.

params
Returns all params ordered by name. The default implementation uses `dir()` to get all attributes of type `Param`.

3.1.4 Serializers

class RequestRowSerializer
Bases: `sagemaker_pyspark.wrapper.SageMakerJavaWrapper`

setSchema(*schema*)
Sets the rowSchema for this RequestRowSerializer.

Parameters **schema** (*StructType*) – the schema that this RequestRowSerializer will use.

class UnlabeledCSVRequestRowSerializer(*schema=None*, *featuresColumnName='features'*)
Bases: `sagemaker_pyspark.transformation.serializers.serializers.RequestRowSerializer`

Serializes according to the current implementation of the scoring service.

Parameters

- **schema** (*StructType*) – tbd
- **featuresColumnName** (*str*) – name of the features column.

class ProtobufRequestRowSerializer(*schema=None*, *featuresColumnName='features'*)
Bases: `sagemaker_pyspark.transformation.serializers.serializers.RequestRowSerializer`

A RequestRowSerializer for converting labeled rows to SageMaker Protobuf-in-recordio request data.

Parameters **schema** (*StructType*) – The schema of Rows being serialized. This parameter is optional as the schema may not be known when this serializer is constructed.

class LibSVMRequestRowSerializer(*schema=None*, *labelColumnName='label'*, *featuresColumnName='features'*)
Bases: `sagemaker_pyspark.transformation.serializers.serializers.RequestRowSerializer`

Extracts a label column and features column from a Row and serializes as a LibSVM record.

Each Row must contain a Double column and a Vector column containing the label and features respectively. Row field indexes for the label and features are obtained by looking up the index of labelColumnName and featuresColumnName respectively in the specified schema.

A schema must be specified before this RequestRowSerializer can be used by a client. The schema is set either on instantiation of this RequestRowSerializer or by *RequestRowSerializer.setSchema()*.

Parameters

- **schema** (*StructType*) – The schema of Rows being serialized. This parameter is optional as the schema may not be known when this serializer is constructed.

- **labelColumnName** (*str*) – The name of the label column.
- **featuresColumnName** (*str*) – The name of the features column.

3.1.5 Deserializers

class ResponseRowDeserializer
Bases: sagemaker_pyspark.wrapper.SageMakerJavaWrapper

class XGBoostCSVRowDeserializer(*prediction_column_name='prediction'*)
Bases: sagemaker_pyspark.transformation.deserializers.deserializers.ResponseRowDeserializer

A *ResponseRowDeserializer* for converting a comma-delimited string of predictions to labeled Vectors.

Parameters **prediction_column_name** (*str*) – the name of the output predictions column.

class ProtobufResponseRowDeserializer(*schema*, *protobufKeys=None*)
Bases: sagemaker_pyspark.transformation.deserializers.deserializers.ResponseRowDeserializer

A *ResponseRowDeserializer* for converting SageMaker Protobuf-in-recordio response data to Spark rows.

Parameters **schema** (*StructType*) – The schema of rows in the response.

class PCAProtobufResponseRowDeserializer(*projection_column_name='projection'*)
Bases: sagemaker_pyspark.transformation.deserializers.deserializers.ResponseRowDeserializer

Deserializes a Protobuf response from the PCA model image to a Vector of Doubles containing the projection of the input vector.

Parameters **projection_column_name** (*str*) – name of the column holding Vectors of Doubles representing the projected vectors.

class LDAProtobufResponseRowDeserializer(*projection_column_name='topic_mixture'*)
Bases: sagemaker_pyspark.transformation.deserializers.deserializers.ResponseRowDeserializer

Deserializes a Protobuf response from the LDA model image to a Vector of Doubles representing the topic mixture for the document represented by the input vector.

Parameters **projection_column_name** (*str*) – name of the column holding Vectors of Doubles representing the topic mixtures for the documents.

class KMeansProtobufResponseRowDeserializer(*distance_to_cluster_column_name='distance_to_cluster'*, *closest_cluster_column_name='closest_cluster'*)
Bases: sagemaker_pyspark.transformation.deserializers.deserializers.ResponseRowDeserializer

Deserializes a Protobuf response from the Kmeans model image into Rows in a Spark DataFrame.

Parameters

- **distance_to_cluster_column_name** (*str*) – name of the column of doubles indicating the distance to the nearest cluster from the input record.
- **closest_cluster_column_name** (*str*) – name of the column of doubles indicating the label of the closest cluster for the input record.

class LinearLearnerBinaryClassifierProtobufResponseRowDeserializer(*score_column_name='score'*, *predicted_label_column_name='predi*

Bases: sagemaker_pyspark.transformation.deserializers.deserializers.ResponseRowDeserializer

Deserializes a Protobuf response from the LinearLearner model image with predictorType "binary_classifier" into Rows in a Spark Dataframe.

Parameters

- **score_column_name** (*str*) – name of the column indicating the output score for the record.
- **predicted_label_column_name** (*str*) – name of the column indicating the predicted label for the record.

class LinearLearnerMultiClassClassifierProtobufResponseRowDeserializer(*score_column_name='score'*, *predicted_label_column_name=*

Bases: sagemaker_pyspark.transformation.deserializers.deserializers.ResponseRowDeserializer

Deserializes a Protobuf response from the LinearLearner model image with predictorType "multiclass_classifier" into Rows in a Spark Dataframe.

Parameters

- **score_column_name** (*str*) – name of the column indicating the output score for the record.
- **predicted_label_column_name** (*str*) – name of the column indicating the predicted label for the record.

class LinearLearnerRegressorProtobufResponseRowDeserializer(*score_column_name='score'*)

Bases: sagemaker_pyspark.transformation.deserializers.deserializers.ResponseRowDeserializer

Deserializes a Protobuf response from the LinearLearner model image with predictorType "regressor" into Rows in a Spark DataFrame.

Parameters **score_column_name** (*str*) – name of the column of Doubles indicating the output score for the record.

class FactorizationMachinesBinaryClassifierDeserializer(*score_column_name='score'*, *predicted_label_column_name='predicted_label'*)

Bases: sagemaker_pyspark.transformation.deserializers.deserializers.ResponseRowDeserializer

Deserializes a Protobuf response from the Factorization Machines model image with predictorType "binary_classifier" into Rows in a Spark Dataframe.

Parameters

- **score_column_name** (*str*) – name of the column indicating the output score for the record.
- **predicted_label_column_name** (*str*) – name of the column indicating the predicted label for the record.

class FactorizationMachinesRegressorDeserializer(*score_column_name='score'*)
Bases: sagemaker_pyspark.transformation.deserializers.deserializers.ResponseRowDeserializer

Deserializes a Protobuf response from the Factorization Machines model image with predictorType "regressor" into Rows in a Spark DataFrame.

Parameters **score_column_name** (*str*) – name of the column of Doubles indicating the output score for the record.

class LibSVMResponseRowDeserializer(*dim*, *labelColumnName*, *featuresColumnName*)
Bases: sagemaker_pyspark.transformation.deserializers.deserializers.ResponseRowDeserializer

A *ResponseRowDeserializer* for converting LibSVM response data to labeled vectors.

Parameters

- **dim** (*int*) – The vector dimension
- **labelColumnName** (*str*) – The name of the label column
- **featuresColumnName** (*str*) – The name of the features column

3.1.6 Other Classes

Top-level module for sagemaker_pyspark

class SageMakerJavaWrapper
Bases: pyspark.ml.wrapper.JavaWrapper

class IAMRole(*role*)
Bases: sagemaker_pyspark.IAMRoleResource.IAMRoleResource

Specifies an IAM Role by ARN or Name.

Parameters **role** (*str*) – IAM Role Name or ARN.

class IAMRoleFromConfig(*configKey='com.amazonaws.services.sagemaker.sparksdk.sagemakerrole'*)
Bases: sagemaker_pyspark.IAMRoleResource.IAMRoleResource

Gets an IAM role from Spark config

Parameters **configKey** (*str*) – key in Spark config corresponding to IAM Role ARN.

class S3DataPath(*bucket*, *objectPath*)
Bases: sagemaker_pyspark.S3Resources.S3Resource, sagemaker_pyspark.wrapper.SageMakerJavaWrapper

Represents a location within an S3 Bucket.

Parameters

- **bucket** (*str*) – An S3 Bucket Name.
- **objectPath** (*str*) – An S3 key or key prefix.

class S3AutoCreatePath
Bases: sagemaker_pyspark.S3Resources.S3Resource, sagemaker_pyspark.wrapper.SageMakerJavaWrapper

Defines an S3 location that will be auto-created at runtime.

class S3Resource
Bases: sagemaker_pyspark.wrapper.SageMakerJavaWrapper

An S3 Resource for SageMaker to use.

class EndpointCreationPolicy
Bases: object

Determines whether and when to create the Endpoint and other Hosting resources.

CREATE_ON_CONSTRUCT
create the Endpoint upon creation of the SageMakerModel, at the end of fit()

CREATE_ON_TRANSFORM
create the Endpoint upon invocation of SageMakerModel.transform().

DO_NOT_CREATE
do not create the Endpoint.

class Option(*value*)
Bases: sagemaker_pyspark.wrapper.SageMakerJavaWrapper

class RandomNamePolicy(*prefix=''*)
Bases: sagemaker_pyspark.NamePolicy.NamePolicy

Provides random, unique SageMaker entity names that begin with the specified prefix.

Parameters **prefix** (*str*) – The common name prefix for all SageMaker entities named with this NamePolicy.

class RandomNamePolicyFactory(*prefix=''*)
Bases: sagemaker_pyspark.NamePolicy.NamePolicyFactory

Creates a RandomNamePolicy upon a call to createNamePolicy

Parameters **prefix** (*str*) – The common name prefix for all SageMaker entities named with this NamePolicy.

class CustomNamePolicy(*trainingJobName*, *modelName*, *endpointConfigName*, *endpointName*)
Bases: sagemaker_pyspark.NamePolicy.NamePolicy

Provides custom SageMaker entity names.

Parameters

- **trainingJobName** (*str*) – The name of the SageMaker entity
- **modelName** (*str*) – The model name of the SageMaker entity
- **endpointConfigName** (*str*) – The endpoint config name of the SageMaker entity
- **endpointName** (*str*) – The endpoint name of the SageMaker entity

class CustomNamePolicyFactory(*trainingJobName*, *modelName*, *endpointConfigName*, *endpointName*)
Bases: sagemaker_pyspark.NamePolicy.NamePolicyFactory

Creates a CustomNamePolicy upon a call to createNamePolicy

Parameters

- **trainingJobName** (*str*) – The job name of the SageMaker entity with this NamePolicy.
- **modelName** (*str*) – The job name of the SageMaker entity with this NamePolicy.
- **endpointConfigName** (*str*) – The job name of the SageMaker entity with this NamePolicy.

- **endpointName** (*str*) – The job name of the SageMaker entity with this NamePolicy.

class CustomNamePolicyWithTimeStampSuffix(*trainingJobName*, *modelName*, *endpointConfigName*, *endpointName*)

Bases: sagemaker_pyspark.NamePolicy.NamePolicy

Provides custom SageMaker entity names with timestamp suffix.

Parameters

- **trainingJobName** (*str*) – The job name of the SageMaker entity
- **modelName** (*str*) – The model name of the SageMaker entity
- **endpointConfigName** (*str*) – The endpoint config name of the SageMaker entity
- **endpointName** (*str*) – The endpoint name of the SageMaker entity

class CustomNamePolicyWithTimeStampSuffixFactory(*trainingJobName*, *modelName*, *endpointConfigName*, *endpointName*)

Bases: sagemaker_pyspark.NamePolicy.NamePolicyFactory

Creates a CustomNamePolicyFactoryWithTimeStampSuffix upon a call to createNamePolicy

Parameters

- **trainingJobName** (*str*) – The job name of the SageMaker entity with this NamePolicy.
- **modelName** (*str*) – The job name of the SageMaker entity with this NamePolicy.
- **endpointConfigName** (*str*) – The job name of the SageMaker entity with this NamePolicy.
- **endpointName** (*str*) – The job name of the SageMaker entity with this NamePolicy.

classpath_jars()

Returns a list with the paths to the required jar files.

The sagemakerpyspark library is mostly a wrapper of the scala sagemakerspark sdk and it depends on a set of jar files to work correctly. This function retrieves the location of these jars in the local installation.

Returns List of absolute paths.

class SageMakerResourceCleanup(*sagemakerClient*, *java_object=None*)

Bases: sagemaker_pyspark.wrapper.SageMakerJavaWrapper

class CreatedResources(*model_name=None*, *endpoint_config_name=None*, *endpoint_name=None*, *java_object=None*)

Bases: sagemaker_pyspark.wrapper.SageMakerJavaWrapper

Resources that may have been created during operation of the SageMaker Estimator and Model.

Parameters

- **model_name** (*str*) – Name of the SageMaker Model that was created, or None if it wasn't created.
- **endpoint_config_name** (*str*) – Name of the SageMaker EndpointConfig that was created, or None if it wasn't created.
- **endpoint_name** (*str*) – Name of the SageMaker Endpoint that was created, or None if it wasn't created.
- **(** (*java_object*) – obj: *py4j.java_gateway.JavaObject*, optional): an existing CreatedResources java instance. If provided the other arguments are ignored.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Python Module Index

S

Index

G

H

I

K

L

O

P

R

S

T

U

X