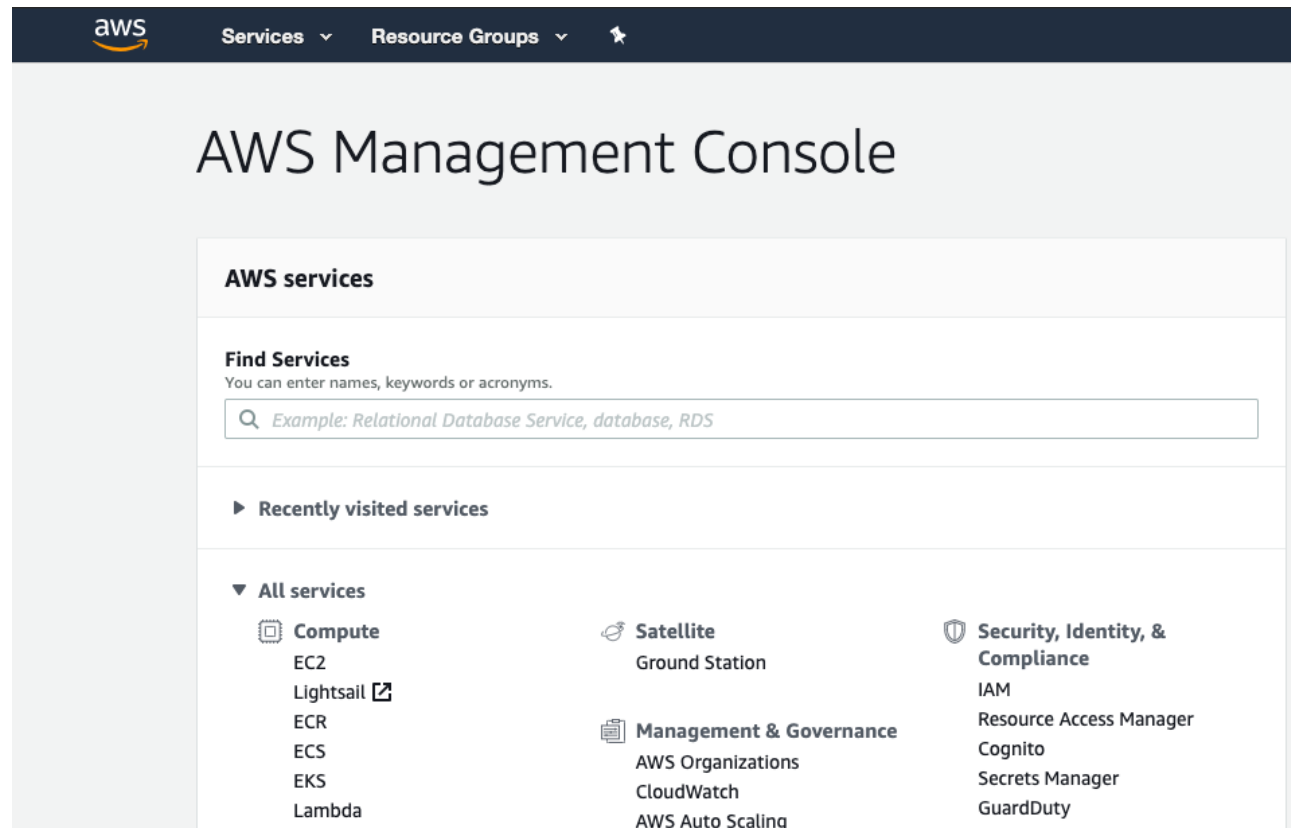


HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER



HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER

Find Services

You can enter names, keywords or acronyms.

 **sagema**

Amazon **SageMaker**

Build, Train, and Deploy Machine Learning Models

Amazon SageMaker



Dashboard

Search

▼ Ground Truth

Labeling jobs

Labeling datasets

Labeling workforces

▼ Notebook

Notebook instances

Lifecycle configurations

Amazon SageMaker > Dashboard



Amazon Elastic Inference

Amazon Elastic Inference adds GPU acceleration to any Amazon SageMaker or EC2 instance for faster inference at much lower cost, with up to 75% savings. Find out if Elastic Inference is right for you.

[Learn more](#) 

Overview



HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER

Amazon SageMaker X

- Amazon SageMaker Studio
- Dashboard
- Search
- ▼ Ground Truth
 - Labeling jobs
 - Labeling datasets
 - Labeling workforces
- ▼ Notebook
 - Notebook instances
 - Lifecycle configurations
 - Git repositories
- ▼ Training
 - Algorithms
 - Training jobs
 - Hyperparameter tuning jobs
- ▼ Inference
 - Compilation jobs
 - Model packages
 - Models
 - Endpoint configurations
 - Endpoints

Amazon SageMaker > Hyperparameter tuning jobs

Hyperparameter tuning jobs

Search hyperparameter tuning jobs

Add/Edit tags

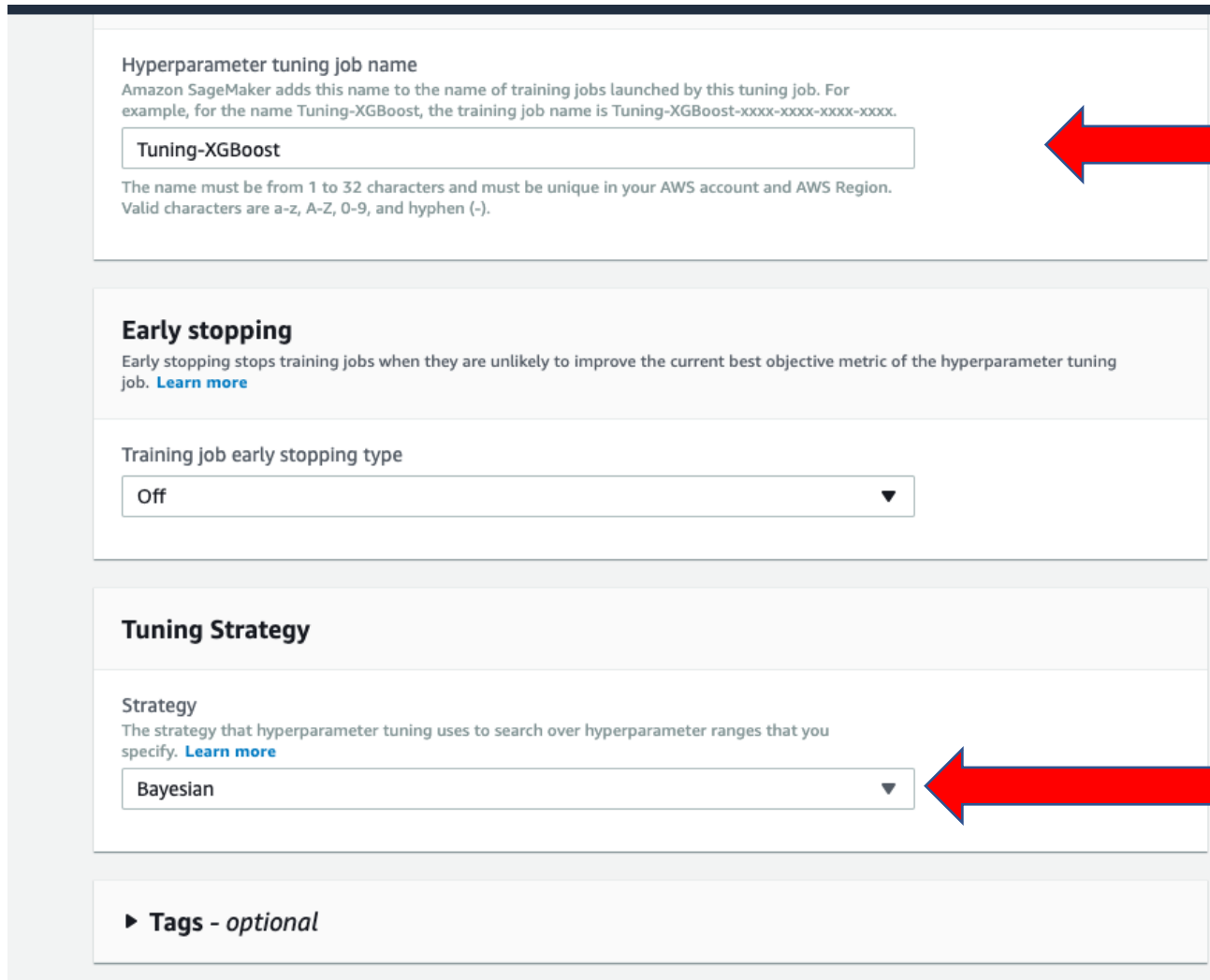
Create hyperparameter tuning job

< 1 > ⚙

Step 1: Click on this

Step 2: Click on this to create a hyperparameter tuning job

HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER



Hyperparameter tuning job name
Amazon SageMaker adds this name to the name of training jobs launched by this tuning job. For example, for the name Tuning-XGBoost, the training job name is Tuning-XGBoost-xxxx-xxxx-xxxx-xxxx.

The name must be from 1 to 32 characters and must be unique in your AWS account and AWS Region. Valid characters are a-z, A-Z, 0-9, and hyphen (-).

Early stopping
Early stopping stops training jobs when they are unlikely to improve the current best objective metric of the hyperparameter tuning job. [Learn more](#)

Training job early stopping type

Tuning Strategy

Strategy
The strategy that hyperparameter tuning uses to search over hyperparameter ranges that you specify. [Learn more](#)

► **Tags - optional**

Provide the name for tuning job name

Select Bayesian strategy

HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER

parameter tuning jobs > Create hyperparameter tuning job

parameter tuning job

Training Job Definition(s)

Choose one or more algorithms you want to use for hyperparameter tuning. [Learn more](#)

Add training job definition

No training definition specified

Choose **Add training definition** to get started.

Click on this to
create a job

HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER

Training job definition

Training job definition name

Tuning-XGBoost

The name must be from 1 to 32 characters and must be unique in your AWS account and AWS Region. Valid characters are a-z, A-Z, 0-9, and hyphen (-).

Provide the name for training job

Network

Configure your VPC to ensure that your training jobs have access to protected resources.

IAM role

Amazon SageMaker requires permissions to call other services on your behalf. Choose a role or let us create a role that has the [AmazonSageMakerFullAccess](#) IAM policy attached.

AmazonSageMaker-ExecutionRole-20191104T033920

☒ Enable network isolation

Containers that run with network isolation can't make any outbound network calls. This is required for AWS ML Marketplace products.

VPC - optional

For better security, we recommend that you use a private VPC.

No VPC

Select the IAM role that is used for your notebook instance.

HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER

Algorithm options

Use an Amazon SageMaker built-in algorithm, your own algorithm, or a third-party algorithm from AWS Marketplace.

▼ Algorithm source

- ☒ Amazon SageMaker built-in algorithm [Learn more](#)
- ☐ Your own algorithm resource
- ☐ Your own algorithm container in ECR [Learn more](#)
- ☐ An algorithm subscription from AWS Marketplace

▼ Choose an algorithm

XGBoost

i You have chosen to create a tuning job using an older 0.72 version of the XGBoost algorithm. There are newer versions of XGBoost that you can use with the Amazon SageMaker Python SDK. For more information about choosing the appropriate version, see [here](#)

Container

The registry path where the training image is stored in Amazon ECR. [Learn more](#)

825641698319.dkr.ecr.us-east-2.amazonaws.com/xgboost:1

Input mode

You can provide your training data as a file or pipe.

File

☐ Enable SageMaker metrics time series

Select the built-in algorithm

Select XGBoost from the drop-down menu

HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER

Objective metric

To find the best training job, set an objective metric and tuning type. See the hyperparameter tuning job detail page for a summary of the best training job.

Objective metric

validation:rmse

Type

minimize

Select validation:rmse from the drop-down menu and choose type minimize

HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER

Name	Type	Scaling type	Value / Range
early_stopping_rounds	Static	-	
csv_weights	Static	-	0
num_round	Integer	Linear	50 - 300
boosters	Static	-	gbtree
silent	Static	-	0
nthread	Static	-	
eta	Continuous	Linear	0.01 - 0.5
gamma	Continuous	Linear	
max_depth	Integer	Linear	5 - 25
min_child_weight	Continuous	Linear	
max_delta_step	Integer	Linear	
subsample	Continuous	Linear	
colsample_bytree	Continuous	Linear	0.1 - 0.5
colsample_bylevel	Continuous	Linear	
lambda	Continuous	Linear	
alpha	Continuous	Linear	1 - 20

Select the value range

Select the value range

Select the value range

Select the value range

Select the value range

HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER

grow_policy	Static ▼	-	depthwise ▼
max_leaves	Static ▼	-	0
max_bin	Static ▼	-	256
sample_type	Static ▼	-	uniform ▼
normalize_type	Static ▼	-	tree ▼
rate_drop	Static ▼	-	0.0
one_drop	Static ▼	-	0
skip_drop	Static ▼	-	0.0
lambda_bias	Static ▼	-	0.0
tweedie_variance_power	Static ▼	-	1.5
objective	Static ▼	-	reg:linear ▼
num_class	Static ▼	-	
base_score	Static ▼	-	0.5
eval_metric	Static ▼	-	rmse ▼
* Required			

Cancel Previous Next

Select rmse

Select next

HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER

Type in the channel name (train)

Mention the content type

Add the second channel

The screenshot displays the configuration interface for a training channel in the SageMaker console. The channel is named 'train' and is set to use 'File' as the input mode. The content type is specified as 'csv'. The compression type and record wrapper are both set to 'None'. The data source is 'S3', with 'S3Prefix' as the data type and 'FullyReplicated' as the data distribution type. The S3 location is provided as 's3://sagemaker-us-east-2-542063182511/XGBoost-Regressor/train/XGBoost-Regressor/'. A red arrow points to the 'Add channel' button at the bottom.

Channels

▼ train Remove

Channel name

Input mode - optional File

Content type - optional

Compression type None

Record wrapper None

Data source
☒ S3
☐ File system

S3 data type S3Prefix

S3 data distribution type FullyReplicated

S3 location

Add channel

Select file from the drop-down menu

Provide the path of training data that is uploaded to S3

HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER

Type in the channel name (validation)

▼ validation

Remove

Channel name

validation

Input mode - optional

File

Content type - optional

csv

Compression type

None

Record wrapper

None

Data source

☒ S3

☐ File system

S3 data type

S3Prefix

S3 data distribution type

FullyReplicated

S3 location

gemaker-us-east-2-542063182511/XGBoost-Regressor/validation/XGBoost-Regressor

Add channel

Select file from the drop-down menu

Mention the content type

Provide the path of validation data that is uploaded to S3

HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER

Checkpoint configuration - optional

The algorithm is responsible for periodically generating checkpoints. The checkpoints are saved to this location and used to resume managed spot training jobs. [Learn more](#)

S3 location

To find a path, [go to Amazon S3](#)

Local path - optional

If the algorithm provides checkpoints, this is the local location the checkpoints are written to. If you do not specify a location, the checkpoints are written to /opt/ml/checkpoints.

Output data configuration

S3 output path

To find a path, [go to Amazon S3](#)

Encryption key - optional

An encryption key protects your data. Type the key ID or key ARN that you want to use.

Cancel

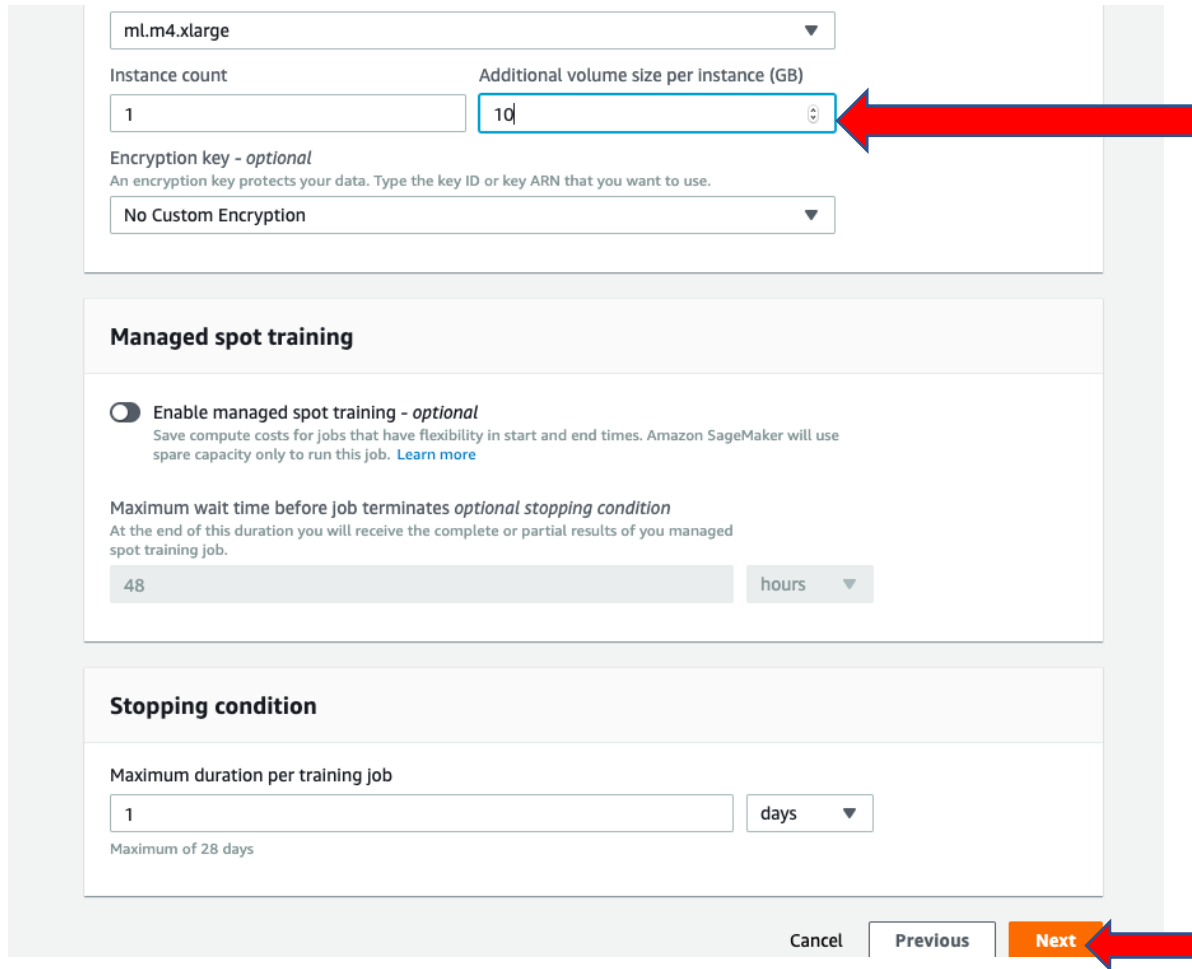
Previous

Next

Provide the S3 output path

Select next

HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER



ml.m4.xlarge

Instance count: 1

Additional volume size per instance (GB): 10

Encryption key - optional
An encryption key protects your data. Type the key ID or key ARN that you want to use.

No Custom Encryption

Managed spot training

☐ Enable managed spot training - optional
Save compute costs for jobs that have flexibility in start and end times. Amazon SageMaker will use spare capacity only to run this job. [Learn more](#)

Maximum wait time before job terminates optional stopping condition
At the end of this duration you will receive the complete or partial results of you managed spot training job.

48 hours

Stopping condition

Maximum duration per training job

1 days

Maximum of 28 days

Cancel Previous Next

Provide the volume size according to the size of dataset. We use 10 GB, as we have a small dataset.

Select Next

HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER

[Hyperparameter tuning jobs](#) > Create hyperparameter tuning job

Hyperparameter tuning job

Training Job Definition(s)

Choose one or more algorithms you want to use for hyperparameter tuning. [Learn more](#)

Add training job definition

Name	Algorithm	Objective metric	Instance type	Instance count	Action
Tuning-XGBoost	XGBoost	validation:rmse	ml.m4.xlarge	1	Edit Clone Remove

Cancel

Previous

Next

Select Next

HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER

Hyperparameter tuning job

Resource limits

Specify resource limits for hyperparameter tuning and training jobs.

Maximum parallel training jobs

The maximum number of concurrent training jobs that the hyperparameter tuning job can run.

A hyperparameter tuning job can run a maximum of 10 parallel training jobs.

Maximum training jobs

The maximum number of training jobs that the hyperparameter tuning job can run.

A hyperparameter tuning job can run a maximum of 100 training jobs.

Cancel

Previous

Next

Select Next

Select the number of jobs to be run in parallel.

Total number of training jobs to be performed

HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER

Step 2: Create training job definition(s)

Training Job Definition(s)

Choose one or more algorithms you want to use for hyperparameter tuning. [Learn more](#)

Name	Algorithm	Objective metric	Instance type	Instance count	Action
ssss	XGBoost	validation:rmse	ml.m4.xlarge	1	Edit

Step 3: Tuning job resource configuration

Resource limits

[Edit](#)

Maximum number of parallel training jobs

1

Maximum total number of training jobs


1


[Cancel](#)[Previous](#)[Create hyperparameter tuning job](#)



Finally click this to create the tuning job

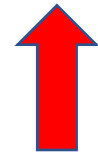
HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER

Amazon SageMaker > Hyperparameter tuning jobs

Hyperparameter tuning jobs  [Add/Edit tags](#) [Create hyperparameter tuning job](#)

< 1 > 


	Name ▾	Status ▾	Training completed/total	Creation time ▾	Duration
	Tuning-XGBoost	 InProgress	0 / 3	May 07, 2020 18:30 UTC	2 minutes




Tuning job in progress

HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER

Amazon SageMaker > Hyperparameter tuning jobs

Hyperparameter tuning jobs  [Add/Edit](#)

	Name	Status	Training completed/total
<input type="radio"/>	Tuning-XGBoost	 Completed	30 / 30



After the completion, click this.

HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER

[Best training job](#) | [Training jobs](#) | [Training job definitions](#) | [Tuning Job configuration](#) | [Tags](#)

Best training job summary

This training job is the best training job for only this hyperparameter tuning job.

Create model

Name	Status	Objective metric	Value
Tuning-XGBoost-018-0a402ba4	✔ Completed	validation:rmse	0.26034900546073914

Best training job hyperparameters

Name	▲	Type	▼	Value
alpha		-		1.0994354985124635
colsample_bytree		-		0.3913546819101119
eta		-		0.23848185159806115
max_depth		-		25
num_round		-		237

You can scroll down to see the best training job hyper-parameters

Finally, you can copy these values to train your model in your notebook instance.

HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER

```
[289]: Xgboost_regressor = sagemaker.estimator.Estimator(container,
                                                    role,
                                                    train_instance_count=1,
                                                    train_instance_type='ml.m4.xlarge',
                                                    output_path=output_location,
                                                    sagemaker_session=sagemaker_session)
Xgboost_regressor.set_hyperparameters(max_depth=25,
                                      objective='reg:linear',
                                      colsample_bytree = 0.3913546819101119,
                                      alpha = 1.0994354985124635,
                                      eta = 0.23848185159806115,
                                      num_round = 237
                                      )
```

HYPERPARAMETERS OPTIMIZATION STEPS IN SAGEMAKER

```
[294]: from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
      from math import sqrt
      k = X_test.shape[1]
      n = len(X_test)
      RMSE = float(format(np.sqrt(mean_squared_error(y_test, predicted_values)), '.3f'))
      MSE = mean_squared_error(y_test, predicted_values)
      MAE = mean_absolute_error(y_test, predicted_values)
      r2 = r2_score(y_test, predicted_values)
      adj_r2 = 1-(1-r2)*(n-1)/(n-k-1)

      print('RMSE =', RMSE, '\nMSE =', MSE, '\nMAE =', MAE, '\nR2 =', r2, '\nAdjusted R2 =', adj_r2)

RMSE = 0.182
MSE = 0.033086948
MAE = 0.07689543
R2 = 0.9668608738856486
Adjusted R2 = 0.9668482913982772
```