Lab Assignment-11

Working with Amazon SageMaker

Amazon SageMaker is a fully managed machine learning service. With SageMaker, data scientists and developers can quickly and easily build and train machine learning models, and then directly deploy them into a production-ready hosted environment.

 It provides an integrated Jupyter authoring notebook instance for easy access to your data sources for exploration and analysis, so you don't have to manage servers.

 It also provides common machine learning algorithms that are optimized to run efficiently against extremely large data in a distributed environment.

 With native support for bring-your-own-algorithms and frameworks, SageMaker offers flexible distributed training options that adjust to your specific workflows.

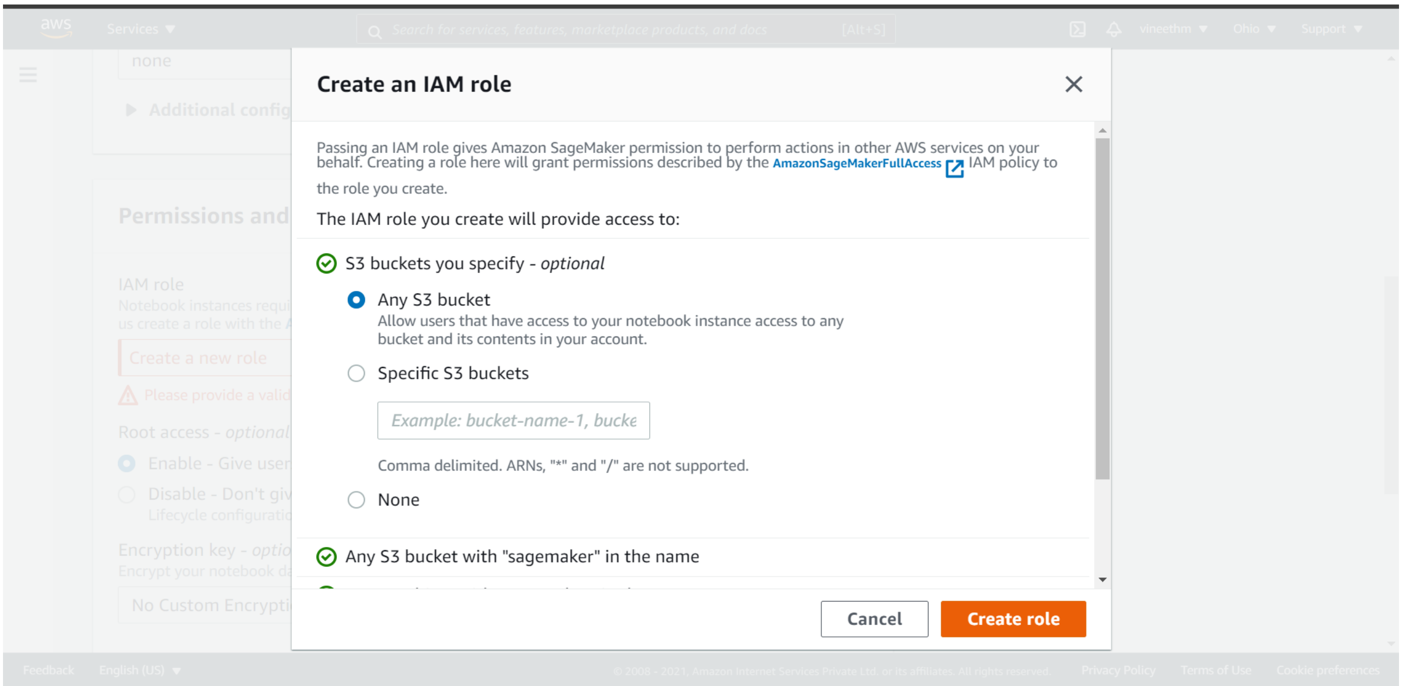
 Deploy a model into a secure and scalable environment by launching it with a few clicks from SageMaker Studio or the SageMaker console.

 Training and hosting are billed by minutes of usage, with no minimum fees and no upfront Commitments

Step-1: Build, train and deploy a Machine Learning model with Amazon SageMaker.

Task-1: Create a notebook instance.

Note: Create an IAM role to enable the SageMaker access S3 buckets.



aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
vineethm ▼
Ohio ▼
Support ▼

Create an IAM role

Passing an IAM role gives Amazon SageMaker permission to perform actions in other AWS services on your behalf. Creating a role here will grant permissions described by the AmazonSageMakerFullAccess IAM policy to the role you create.

The IAM role you create will provide access to:

✓ S3 buckets you specify - *optional*

● Any S3 bucket
Allow users that have access to your notebook instance access to any bucket and its contents in your account.

○ Specific S3 buckets

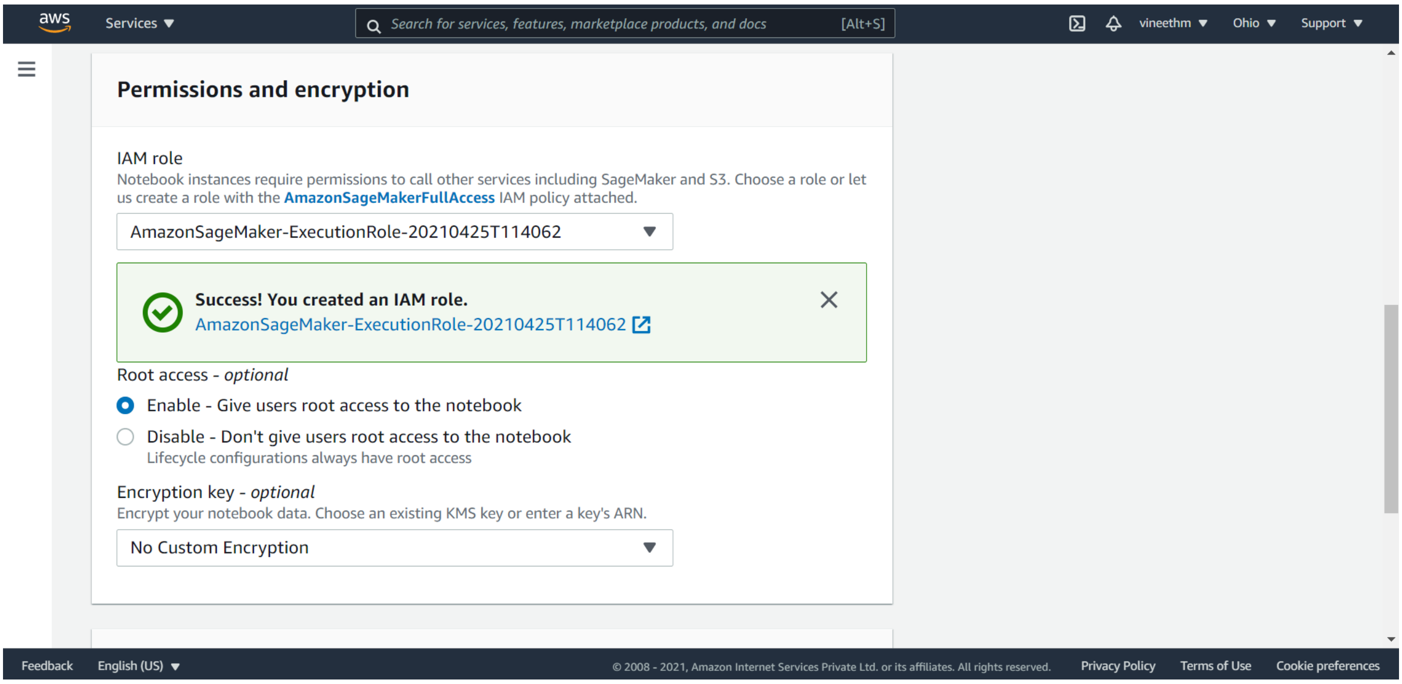
Example: bucket-name-1, bucke

Comma delimited. ARNs, "*" and "/" are not supported.

○ None

✓ Any S3 bucket with "sagemaker" in the name

Cancel
Create role
Feedback
English (US) ▼
© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences



aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
vineethm ▼
Ohio ▼
Support ▼

Permissions and encryption

IAM role
Notebook instances require permissions to call other services including SageMaker and S3. Choose a role or let us create a role with the **AmazonSageMakerFullAccess** IAM policy attached.

AmazonSageMaker-ExecutionRole-20210425T114062 ▼

Success! You created an IAM role.
AmazonSageMaker-ExecutionRole-20210425T114062

Root access - *optional*

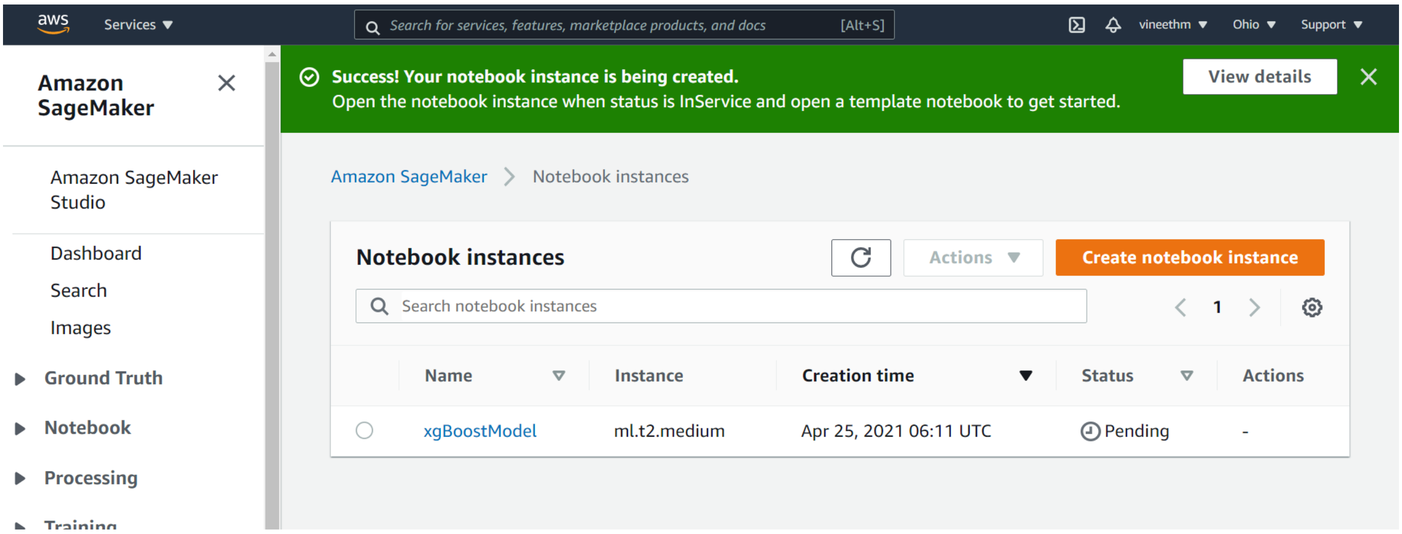
● Enable - Give users root access to the notebook

○ Disable - Don't give users root access to the notebook
Lifecycle configurations always have root access

Encryption key - *optional*
Encrypt your notebook data. Choose an existing KMS key or enter a key's ARN.

No Custom Encryption ▼

Feedback
English (US) ▼
© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences



Success! Your notebook instance is being created.
Open the notebook instance when status is InService and open a template notebook to get started.

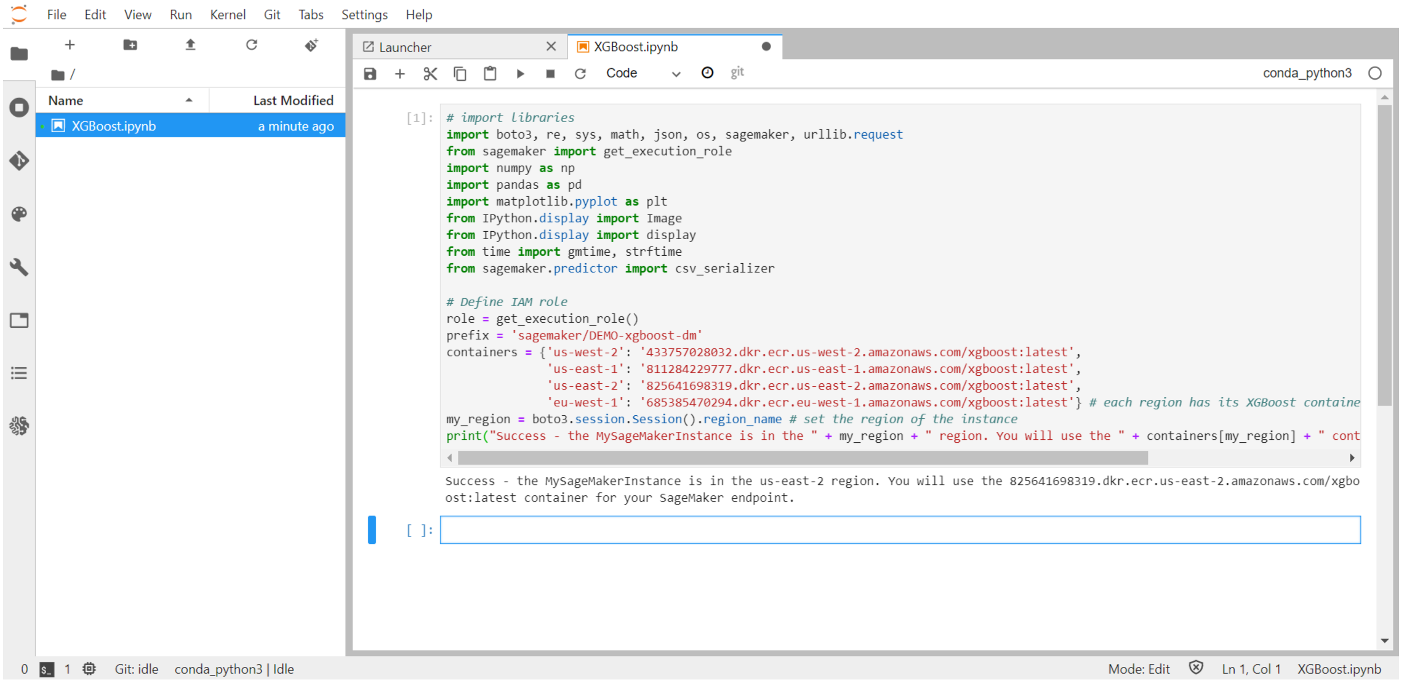
Amazon SageMaker > Notebook instances

Notebook instances

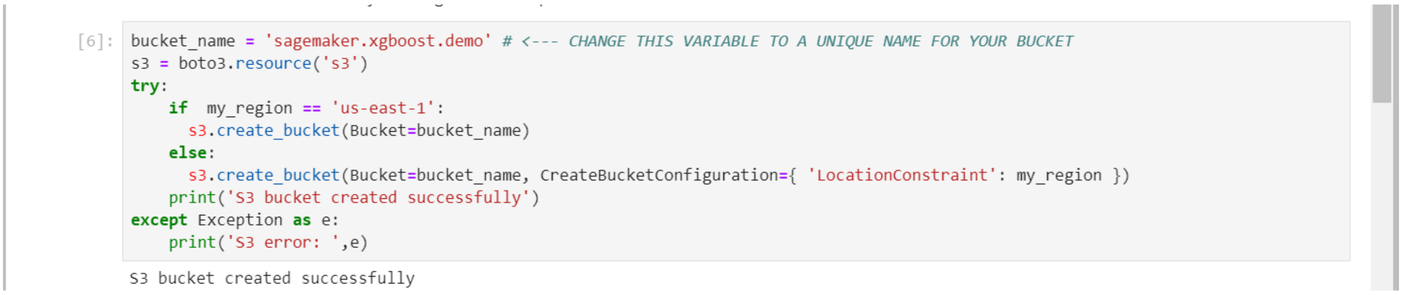
Name	Instance	Creation time	Status	Actions
○ xgBoostModel	ml.t2.medium	Apr 25, 2021 06:11 UTC	Pending	-

Step-2: Preprocess the data in the Jupyter Notebook provisioned by SageMaker.

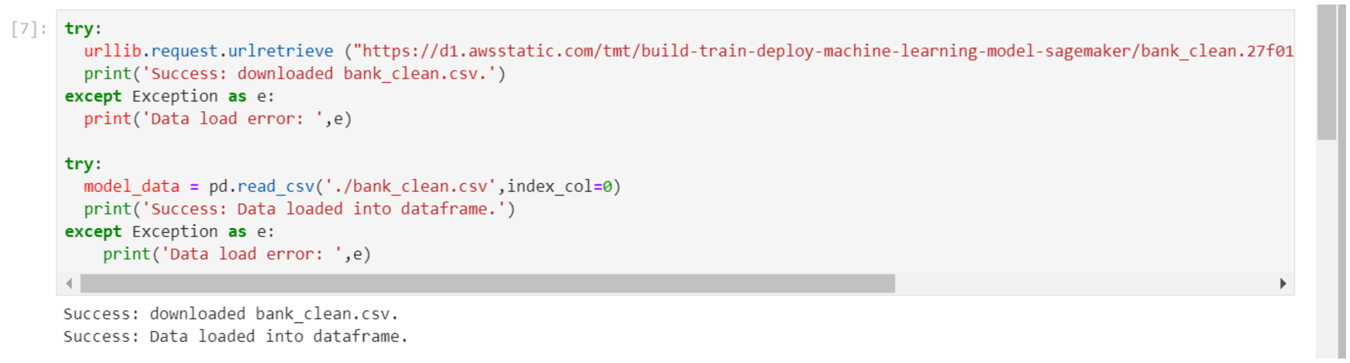
Task-1: Choose the kernel environment “conda_python3”.



```
[1]: # import libraries
     import boto3, re, sys, math, json, os, sagemaker, urllib.request
     from sagemaker import get_execution_role
     import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     from IPython.display import Image
     from IPython.display import display
     from time import gmtime, strftime
     from sagemaker.predictor import csv_serializer

     # Define IAM role
     role = get_execution_role()
     prefix = 'sagemaker/DEMO-xgboost-dm'
     containers = {'us-west-2': '433757028032.dkr.ecr.us-west-2.amazonaws.com/xgboost:latest',
                   'us-east-1': '811284229777.dkr.ecr.us-east-1.amazonaws.com/xgboost:latest',
                   'us-east-2': '825641698319.dkr.ecr.us-east-2.amazonaws.com/xgboost:latest',
                   'eu-west-1': '685385470294.dkr.ecr.eu-west-1.amazonaws.com/xgboost:latest'} # each region has its XGBoost containe
     my_region = boto3.session.Session().region_name # set the region of the instance
     print("Success - the MySageMakerInstance is in the " + my_region + " region. You will use the " + containers[my_region] + " cont

     Success - the MySageMakerInstance is in the us-east-2 region. You will use the 825641698319.dkr.ecr.us-east-2.amazonaws.com/xgbo
     ost:latest container for your SageMaker endpoint.

[ ]:
```

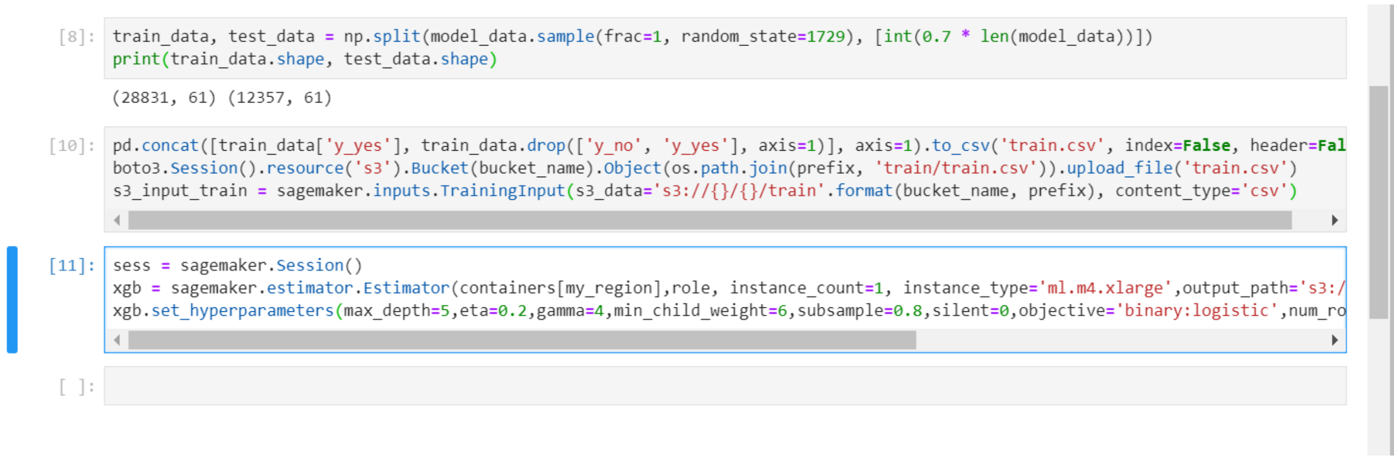


```
[6]: bucket_name = 'sagemaker.xgboost.demo' # <--- CHANGE THIS VARIABLE TO A UNIQUE NAME FOR YOUR BUCKET
     s3 = boto3.resource('s3')
     try:
         if  my_region == 'us-east-1':
           s3.create_bucket(Bucket=bucket_name)
         else:
           s3.create_bucket(Bucket=bucket_name, CreateBucketConfiguration={ 'LocationConstraint': my_region })
         print('S3 bucket created successfully')
     except Exception as e:
         print('S3 error: ',e)

     S3 bucket created successfully
```

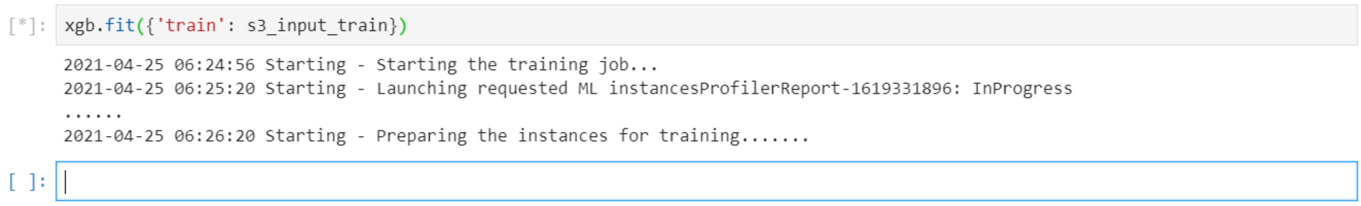


```
[7]: try:
       urllib.request.urlretrieve ("https://d1.awsstatic.com/tmt/build-train-deploy-machine-learning-model-sagemaker/bank_clean.27f01
       print('Success: downloaded bank_clean.csv.')
     except Exception as e:
       print('Data load error: ',e)

     try:
       model_data = pd.read_csv('./bank_clean.csv',index_col=0)
       print('Success: Data loaded into dataframe.')
     except Exception as e:
         print('Data load error: ',e)

     Success: downloaded bank_clean.csv.
     Success: Data loaded into dataframe.
```

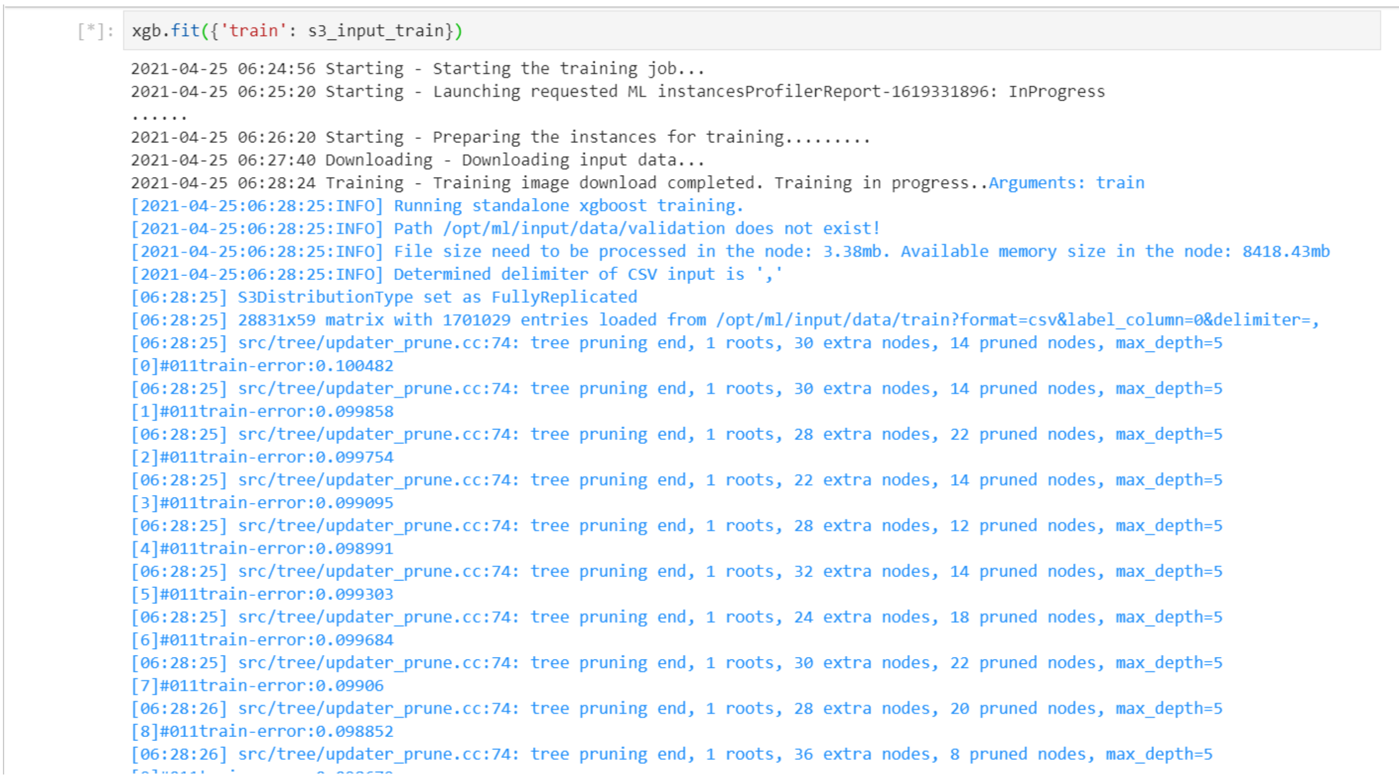
Step-3: Train the Machine Learning Model



```
[8]: train_data, test_data = np.split(model_data.sample(frac=1, random_state=1729), [int(0.7 * len(model_data))])
     print(train_data.shape, test_data.shape)

     (28831, 61) (12357, 61)

[10]: pd.concat([train_data['y_yes'], train_data.drop(['y_no', 'y_yes'], axis=1)], axis=1).to_csv('train.csv', index=False, header=Fal
      boto3.Session().resource('s3').Bucket(bucket_name).Object(os.path.join(prefix, 'train/train.csv')).upload_file('train.csv')
      s3_input_train = sagemaker.inputs.TrainingInput(s3_data='s3://{}/{}/train'.format(bucket_name, prefix), content_type='csv')

[11]: sess = sagemaker.Session()
      xgb = sagemaker.estimator.Estimator(containers[my_region],role, instance_count=1, instance_type='ml.m4.xlarge',output_path='s3:/
      xgb.set_hyperparameters(max_depth=5,eta=0.2,gamma=4,min_child_weight=6,subsample=0.8,silent=0,objective='binary:logistic',num_ro

[ ]:
```



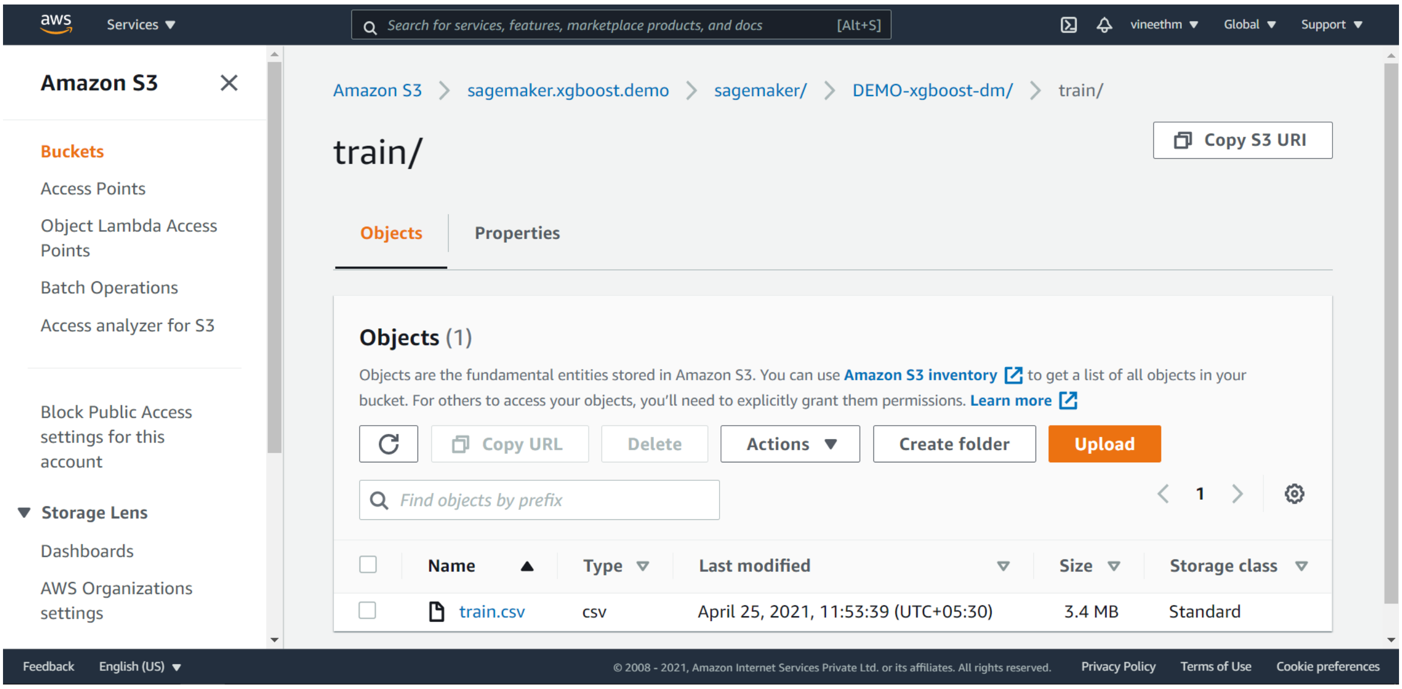
```
[*]: xgb.fit({'train': s3_input_train})

     2021-04-25 06:24:56 Starting - Starting the training job...
     2021-04-25 06:25:20 Starting - Launching requested ML instancesProfilerReport-1619331896: InProgress
     ......
     2021-04-25 06:26:20 Starting - Preparing the instances for training.......

[ ]:
```



```
[*]: xgb.fit({'train': s3_input_train})
```

```
2021-04-25 06:24:56 Starting - Starting the training job...
2021-04-25 06:25:20 Starting - Launching requested ML instancesProfilerReport-1619331896: InProgress
......
2021-04-25 06:26:20 Starting - Preparing the instances for training.........
2021-04-25 06:27:40 Downloading - Downloading input data...
2021-04-25 06:28:24 Training - Training image download completed. Training in progress..Arguments: train
[2021-04-25:06:28:25:INFO] Running standalone xgboost training.
[2021-04-25:06:28:25:INFO] Path /opt/ml/input/data/validation does not exist!
[2021-04-25:06:28:25:INFO] File size need to be processed in the node: 3.38mb. Available memory size in the node: 8418.43mb
[2021-04-25:06:28:25:INFO] Determined delimiter of CSV input is ','
[06:28:25] S3DistributionType set as FullyReplicated
[06:28:25] 28831x59 matrix with 1701029 entries loaded from /opt/ml/input/data/train?format=csv&label_column=0&delimiter=,
[06:28:25] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 30 extra nodes, 14 pruned nodes, max_depth=5
[0]#011train-error:0.100482
[06:28:25] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 30 extra nodes, 14 pruned nodes, max_depth=5
[1]#011train-error:0.099858
[06:28:25] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 28 extra nodes, 22 pruned nodes, max_depth=5
[2]#011train-error:0.099754
[06:28:25] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 22 extra nodes, 14 pruned nodes, max_depth=5
[3]#011train-error:0.099095
[06:28:25] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 28 extra nodes, 12 pruned nodes, max_depth=5
[4]#011train-error:0.098991
[06:28:25] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 32 extra nodes, 14 pruned nodes, max_depth=5
[5]#011train-error:0.099303
[06:28:25] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 24 extra nodes, 18 pruned nodes, max_depth=5
[6]#011train-error:0.099684
[06:28:25] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 30 extra nodes, 22 pruned nodes, max_depth=5
[7]#011train-error:0.09906
[06:28:26] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 28 extra nodes, 20 pruned nodes, max_depth=5
[8]#011train-error:0.098852
[06:28:26] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 36 extra nodes, 8 pruned nodes, max_depth=5
```

Note: The required files are created in the S3 bucket.



Amazon S3 > sagemaker.xgboost.demo > sagemaker/ > DEMO-xgboost-dm/ > train/

train/

Objects | Properties

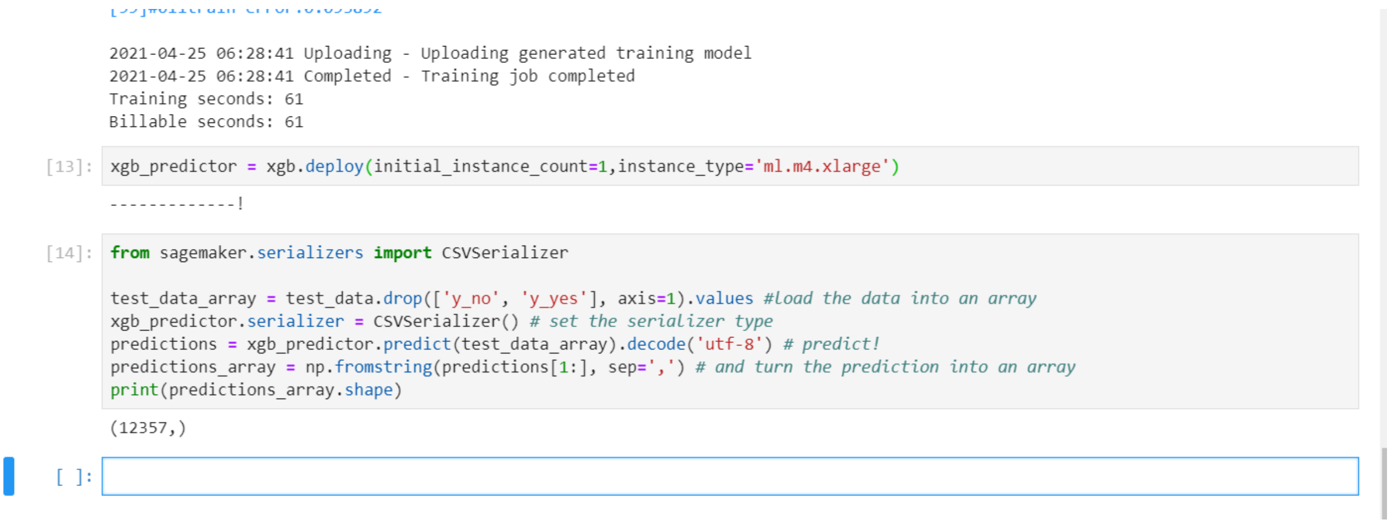
Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name ▲	Type ▽	Last modified ▽	Size ▽	Storage class ▽
□	train.csv	csv	April 25, 2021, 11:53:39 (UTC+05:30)	3.4 MB	Standard

© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

Step-4: Deploy Model



```
[99]#011train-error:0.093892

2021-04-25 06:28:41 Uploading - Uploading generated training model
2021-04-25 06:28:41 Completed - Training job completed
Training seconds: 61
Billable seconds: 61
```

```
[13]: xgb_predictor = xgb.deploy(initial_instance_count=1,instance_type='ml.m4.xlarge')
```

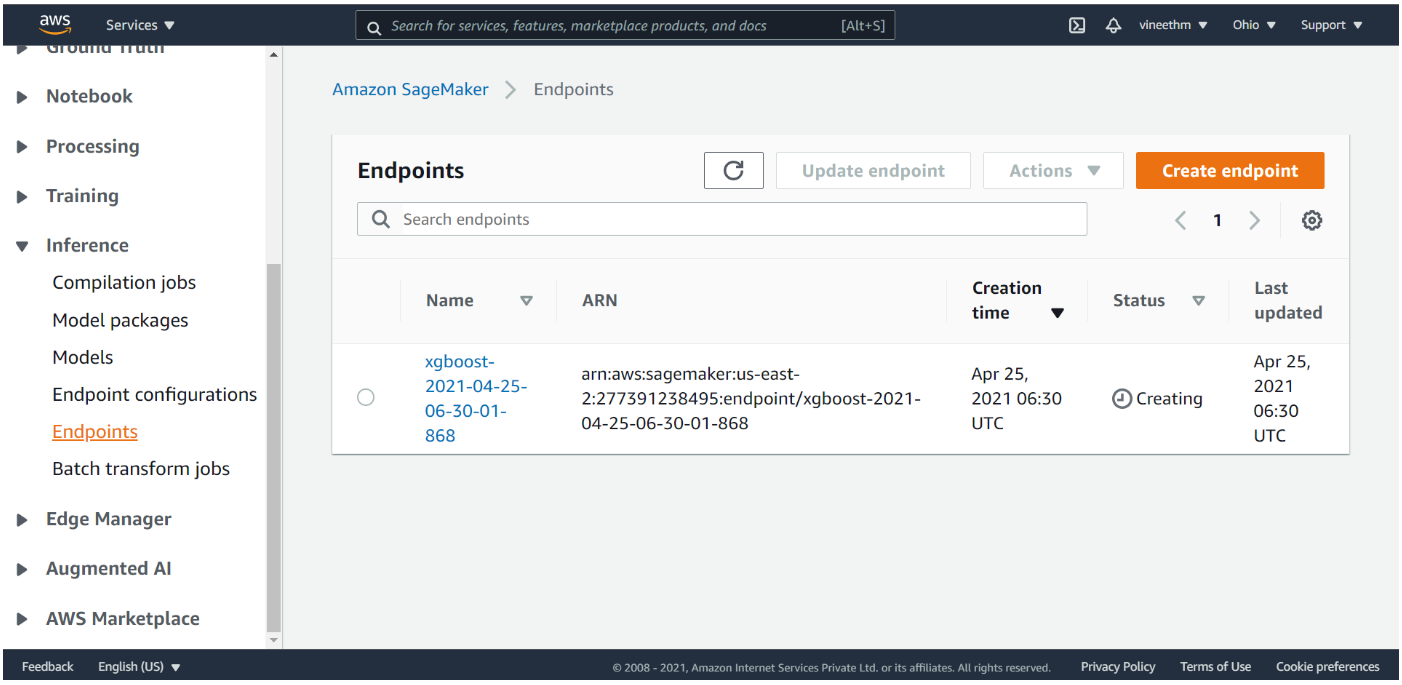
```
-------------!
```

```
[14]: from sagemaker.serializers import CSVSerializer

test_data_array = test_data.drop(['y_no', 'y_yes'], axis=1).values #load the data into an array
xgb_predictor.serializer = CSVSerializer() # set the serializer type
predictions = xgb_predictor.predict(test_data_array).decode('utf-8') # predict!
predictions_array = np.fromstring(predictions[1:], sep=',') # and turn the prediction into an array
print(predictions_array.shape)
```

```
(12357,)
```

```
[ ]:
```

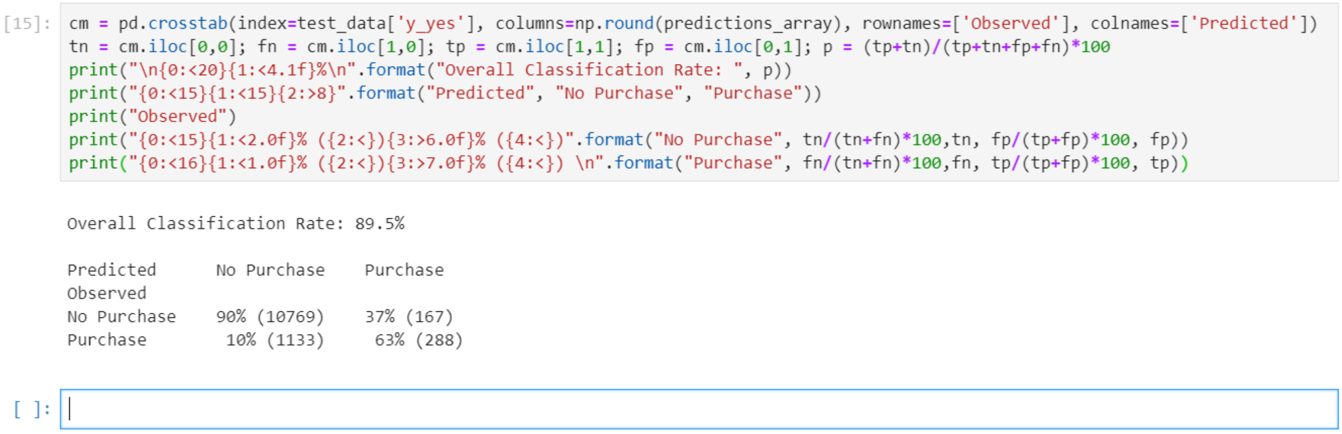
We can notice the endpoint being created.



Endpoints

	Name	ARN	Creation time	Status	Last updated
○	xgboost-2021-04-25-06-30-01-868	arn:aws:sagemaker:us-east-2:277391238495:endpoint/xgboost-2021-04-25-06-30-01-868	Apr 25, 2021 06:30 UTC	Creating	Apr 25, 2021 06:30 UTC

Step-5: Evaluate Model performance.

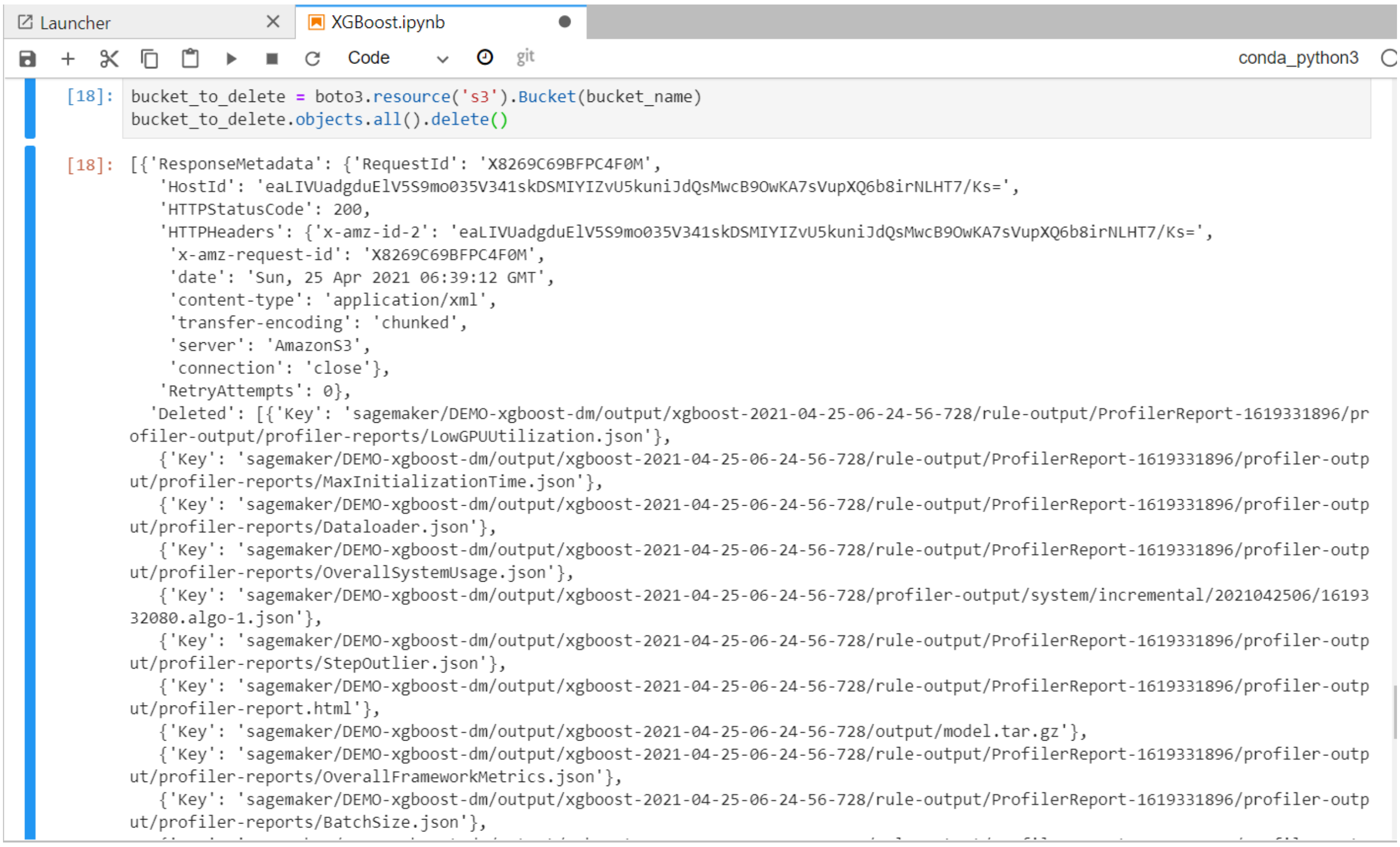


```
[15]: cm = pd.crosstab(index=test_data['y_yes'], columns=np.round(predictions_array), rownames=['Observed'], colnames=['Predicted'])
      tn = cm.iloc[0,0]; fn = cm.iloc[1,0]; tp = cm.iloc[1,1]; fp = cm.iloc[0,1]; p = (tp+tn)/(tp+tn+fp+fn)*100
      print("\n{0:<20}{1:<4.1f}%\n".format("Overall Classification Rate: ", p))
      print("{0:<15}{1:<15}{2:>8}".format("Predicted", "No Purchase", "Purchase"))
      print("Observed")
      print("{0:<15}{1:<2.0f}% ({2:<}){3:>6.0f}% ({4:<})".format("No Purchase", tn/(tn+fn)*100,tn, fp/(tp+fp)*100, fp))
      print("{0:<16}{1:<1.0f}% ({2:<}){3:>7.0f}% ({4:<}) \n".format("Purchase", fn/(tn+fn)*100,fn, tp/(tp+fp)*100, tp))
```

```
Overall Classification Rate: 89.5%

Predicted      No Purchase    Purchase
Observed
No Purchase    90% (10769)    37% (167)
Purchase        10% (1133)     63% (288)
```

```
[ ]: |
```

Step-6: Clean up the resources.



```
[18]: bucket_to_delete = boto3.resource('s3').Bucket(bucket_name)
      bucket_to_delete.objects.all().delete()
```

```
[18]: [{'ResponseMetadata': {'RequestId': 'X8269C69BFPC4F0M',
         'HostId': 'eaLIVUadgduElV5S9mo035V341skDSMIYIZvU5kuniJdQsMwcB9OwKA7sVupXQ6b8irNLHT7/Ks=',
         'HTTPStatusCode': 200,
         'HTTPHeaders': {'x-amz-id-2': 'eaLIVUadgduElV5S9mo035V341skDSMIYIZvU5kuniJdQsMwcB9OwKA7sVupXQ6b8irNLHT7/Ks=',
          'x-amz-request-id': 'X8269C69BFPC4F0M',
          'date': 'Sun, 25 Apr 2021 06:39:12 GMT',
          'content-type': 'application/xml',
          'transfer-encoding': 'chunked',
          'server': 'AmazonS3',
          'connection': 'close'},
         'RetryAttempts': 0},
        'Deleted': [{'Key': 'sagemaker/DEMO-xgboost-dm/output/xgboost-2021-04-25-06-24-56-728/rule-output/ProfilerReport-1619331896/pr
      ofiler-output/profiler-reports/LowGPUUtilization.json'},
         {'Key': 'sagemaker/DEMO-xgboost-dm/output/xgboost-2021-04-25-06-24-56-728/rule-output/ProfilerReport-1619331896/profiler-outp
      ut/profiler-reports/MaxInitializationTime.json'},
         {'Key': 'sagemaker/DEMO-xgboost-dm/output/xgboost-2021-04-25-06-24-56-728/rule-output/ProfilerReport-1619331896/profiler-outp
      ut/profiler-reports/Dataloader.json'},
         {'Key': 'sagemaker/DEMO-xgboost-dm/output/xgboost-2021-04-25-06-24-56-728/rule-output/ProfilerReport-1619331896/profiler-outp
      ut/profiler-reports/OverallSystemUsage.json'},
         {'Key': 'sagemaker/DEMO-xgboost-dm/output/xgboost-2021-04-25-06-24-56-728/profiler-output/system/incremental/2021042506/16193
      32080.algo-1.json'},
         {'Key': 'sagemaker/DEMO-xgboost-dm/output/xgboost-2021-04-25-06-24-56-728/rule-output/ProfilerReport-1619331896/profiler-outp
      ut/profiler-reports/StepOutlier.json'},
         {'Key': 'sagemaker/DEMO-xgboost-dm/output/xgboost-2021-04-25-06-24-56-728/rule-output/ProfilerReport-1619331896/profiler-outp
      ut/profiler-report.html'},
         {'Key': 'sagemaker/DEMO-xgboost-dm/output/xgboost-2021-04-25-06-24-56-728/output/model.tar.gz'},
         {'Key': 'sagemaker/DEMO-xgboost-dm/output/xgboost-2021-04-25-06-24-56-728/rule-output/ProfilerReport-1619331896/profiler-outp
      ut/profiler-reports/OverallFrameworkMetrics.json'},
         {'Key': 'sagemaker/DEMO-xgboost-dm/output/xgboost-2021-04-25-06-24-56-728/rule-output/ProfilerReport-1619331896/profiler-outp
      ut/profiler-reports/BatchSize.json'},
```