

CLOUD COMPUTING

PRACTICAL 8:AMAZON SAGEMAKER

Name: Ashwibi Shetty

Roll No: A0 MSc00005

SDS Batch 2

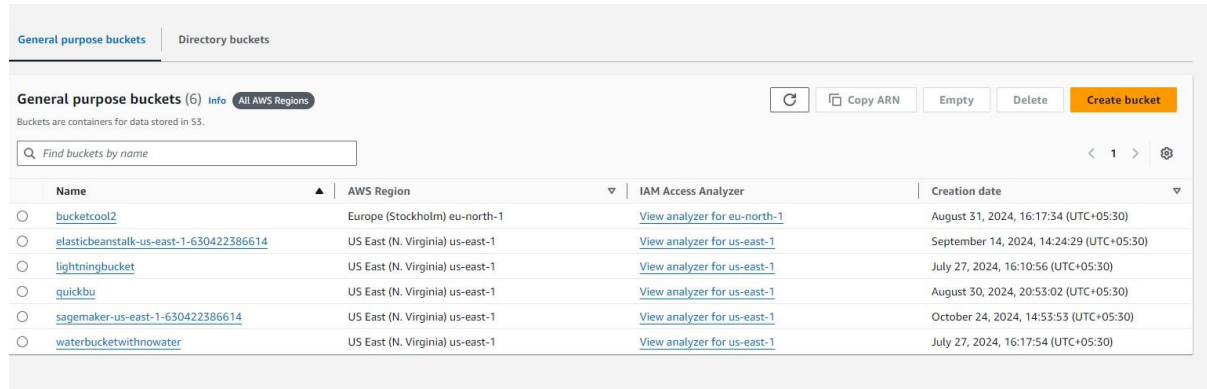
1)Creating IAM ROLE and assigning sagemaker permission

The screenshot shows the 'Create role' wizard in the AWS IAM console. The left sidebar indicates the current step is 'Step 1: Select trusted entity'. The main content area is titled 'Select trusted entity' and includes an 'info' link. Under 'Trusted entity type', there are five radio button options: 'AWS service' (selected), 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. Each option has a brief description. Below this, the 'Use case' section shows 'Service or use case' set to 'SageMaker'. Under 'Choose a use case for the specified service', there are two options: 'SageMaker - Execution' (selected) and 'SageMaker - HyperPod Clusters'. At the bottom right, there are 'Cancel' and 'Next' buttons.

IAM Role is created.

The screenshot shows the 'Add permissions' step of the IAM role creation process. The title is 'Add permissions' with an 'info' link. Below, the 'Permissions policies (1)' section shows a table with one policy: 'AmazonSageMakerFullAccess', which is 'AWS managed'. At the bottom, there is a section for 'Set permissions boundary - optional' with a right-pointing arrow. At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons.

2) creating s3 bucket named mygroup2

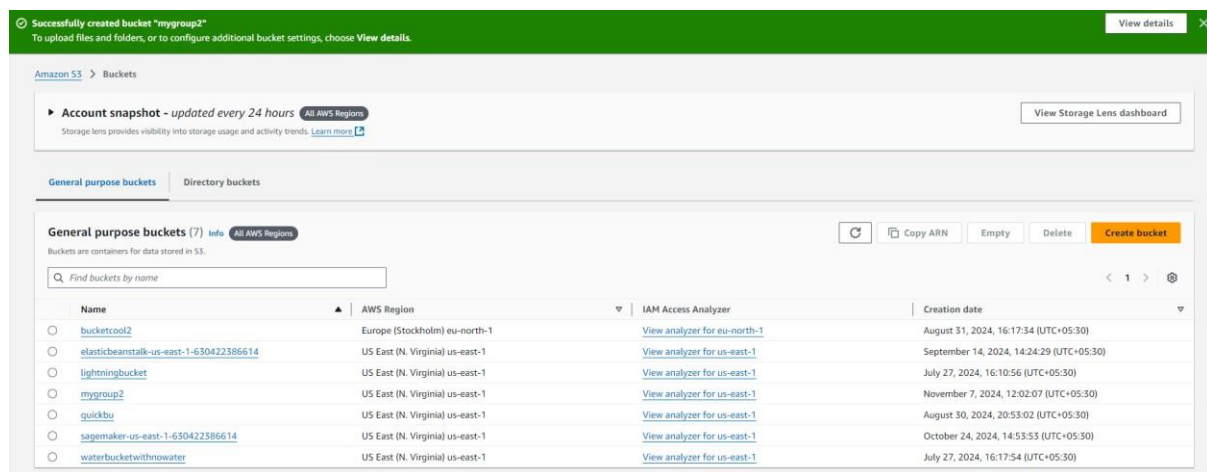


General purpose buckets (6) [Info](#) [All AWS Regions](#)

Buckets are containers for data stored in S3.

Find buckets by name

Name	AWS Region	IAM Access Analyzer	Creation date
bucketcool2	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	August 31, 2024, 16:17:34 (UTC+05:30)
elasticbeanstalk-us-east-1-630422386614	US East (N. Virginia) us-east-1	View analyzer for us-east-1	September 14, 2024, 14:24:29 (UTC+05:30)
lightningbucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	July 27, 2024, 16:10:56 (UTC+05:30)
quickbu	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 30, 2024, 20:53:02 (UTC+05:30)
sagemaker-us-east-1-630422386614	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 24, 2024, 14:53:53 (UTC+05:30)
waterbucketwithnowater	US East (N. Virginia) us-east-1	View analyzer for us-east-1	July 27, 2024, 16:17:54 (UTC+05:30)



✔ Successfully created bucket "mygroup2"
To upload files and folders, or to configure additional bucket settings, choose [View details](#).

Amazon S3 > Buckets

Account snapshot - updated every 24 hours [All AWS Regions](#) [View Storage Lens dashboard](#)

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

General purpose buckets | Directory buckets

General purpose buckets (7) [Info](#) [All AWS Regions](#)

Buckets are containers for data stored in S3.

Find buckets by name

Name	AWS Region	IAM Access Analyzer	Creation date
bucketcool2	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	August 31, 2024, 16:17:34 (UTC+05:30)
elasticbeanstalk-us-east-1-630422386614	US East (N. Virginia) us-east-1	View analyzer for us-east-1	September 14, 2024, 14:24:29 (UTC+05:30)
lightningbucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	July 27, 2024, 16:10:56 (UTC+05:30)
mygroup2	US East (N. Virginia) us-east-1	View analyzer for us-east-1	November 7, 2024, 12:02:07 (UTC+05:30)
quickbu	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 30, 2024, 20:53:02 (UTC+05:30)
sagemaker-us-east-1-630422386614	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 24, 2024, 14:53:53 (UTC+05:30)
waterbucketwithnowater	US East (N. Virginia) us-east-1	View analyzer for us-east-1	July 27, 2024, 16:17:54 (UTC+05:30)

3)open Amazon SageMaker console


Select Notebook instances and click create notebook instances

Here we will assign the IAM role created earlier i.e fraud_detection

[Amazon SageMaker](#) > [Notebooks and Git Repos](#)

Notebooks and Git repos

▼ Try the new JupyterLab in SageMaker Studio



Try the new

JupyterLab in SageMaker Studio

- Launch notebooks in seconds and start coding instantly
- Use the similar underlying compute and storage as your notebook instances to enable more features at the same cost
- Seamlessly perform comprehensive ML and analytics workflows, all in one notebook
- Leverage GenAI-powered coding assistance from Amazon CodeWhisperer and JupyterAI to accelerate development
- Collaborate with your peers in real-time on the same notebook for seamless ideation

Get Started


► [How to access JupyterLab in Studio?](#)

Notebook instances

Git repositories

Notebook instances


Info



Actions ▼

Create notebook instance

< 1 > ⚙

	Name ▼	Instance	Creation time ▼	Status ▼	Actions
<input type="radio"/>	franklin	ml.t3.medium	10/24/2024, 2:40:30 PM	 InService	Open Jupyter Open JupyterLab

4) CREATE A JUPYTER NOTEBOOK

[Amazon SageMaker](#) > [Notebook instances](#) > Create notebook instance

Create notebook instance

Amazon SageMaker provides pre-built fully managed notebook instances that run Jupyter notebooks. The notebook instances include example code for common model training and hosting exercises. [Learn more](#)

Notebook instance settings

Notebook instance name

Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.

Notebook instance type

Platform identifier [Learn more](#)

► Additional configuration

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.

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.

Notebook is created

Success! Your notebook instance is being created.
Open the notebook instance when status is InService and open a template notebook to get started. [View details](#)

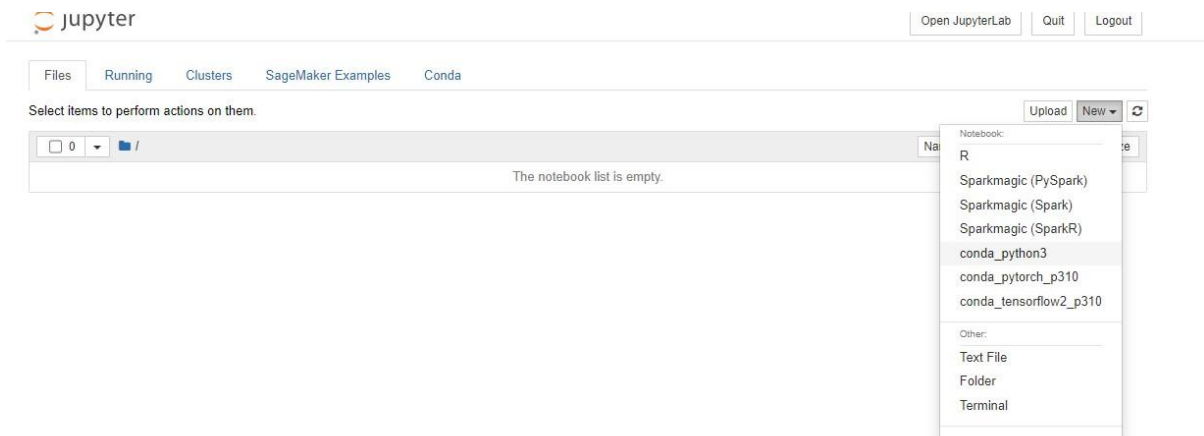
[Amazon SageMaker](#) > [Notebook instances](#)

Notebook instances info

	Name	Instance	Creation time	Status	Actions
<input type="radio"/>	frauddetection	ml.t3.medium	11/7/2024, 12:05:56 PM	Pending	-
<input type="radio"/>	franklin	ml.t3.medium	10/24/2024, 2:40:30 PM	InService	Open Jupyter Open JupyterLab

1. Open Jupyter or JupyterLab according to the interface needed.

2. Go to File menu->Choose New-> Notebook.
3. Select Kernel as 'conda_python3'



Deploying the model (Here it is stored in s3 bucket that we had created)

```
In [1]: import shap
X, y = shap.datasets.adult()
X_display, y_display = shap.datasets.adult(display=True)
feature_names = list(X.columns)
feature_names

Matplotlib is building the font cache; this may take a moment.
```

```
Out[1]: ['Age',
'Workclass',
'Education-Num',
'Marital Status',
'Occupation',
'Relationship',
'Race',
'Sex',
'Capital Gain',
'Capital Loss',
'Hours per week',
'Country']
```

```
In [7]: import sagemaker, boto3, os
bucket = sagemaker.Session().default_bucket()
prefix = "demo-sagemaker-xgboost-adult-income-prediction"

boto3.Session().resource('s3').Bucket(bucket).Object(
    os.path.join(prefix, 'data/train.csv')).upload_file('train.csv')
boto3.Session().resource('s3').Bucket(bucket).Object(
    os.path.join(prefix, 'data/validation.csv')).upload_file('validation.csv')

sagemaker.config INFO - Not applying SDK defaults from location: /etc/xdg/sagemaker/config.yaml
sagemaker.config INFO - Not applying SDK defaults from location: /home/ec2-user/.config/sagemaker/config.yaml
```

```
In [8]: import sagemaker

region = sagemaker.Session().boto_region_name
print("AWS Region: {}".format(region))

role = sagemaker.get_execution_role()
print("RoleArn: {}".format(role))

AWS Region: us-east-1
RoleArn: arn:aws:iam::975050009706:role/lucifer007
```

```
! aws s3 cp {rule_output_path} ./ --recursive
```

```
from IPython.display import FileLink, FileLinks
display("Click link below to view the XGBoost Training report", FileLink("CreateXgboostReport/xgboost_report.html"))
```

download: s3://sagemaker-us-east-1-975050009706/demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-10-24-09-29-24-130/rule-output/CreateXgboostReport/xgboost-reports/EvaluationMetrics.json to CreateXgboostReport/xgboost-reports/EvaluationMetrics.json
download: s3://sagemaker-us-east-1-975050009706/demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-10-24-09-29-24-130/rule-output/CreateXgboostReport/xgboost-reports/FeatureImportance.json to CreateXgboostReport/xgboost-reports/FeatureImportance.json
download: s3://sagemaker-us-east-1-975050009706/demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-10-24-09-29-24-130/rule-output/ProfilerReport/profiler-output/profiler-report.ipynb to ProfilerReport/profiler-output/profiler-report.ipynb
download: s3://sagemaker-us-east-1-975050009706/demo-sagemaker-xgboost-adult-income-prediction/xgboost_model/sagemaker-xgboost-2024-10-24-09-29-24-130/rule-output/CreateXgboostReport/xgboost-reports/ConfusionMatrix.json to CreateXgboostReport/xgboost-reports/ConfusionMatrix.json

```
from sagemaker.debugger import Rule, ProfilerRule, rule_configs
from sagemaker.session import TrainingInput

s3_output_location='s3://{}/{}/{}/'.format(bucket, prefix, 'xgboost_model')

container=sagemaker.image_uris.retrieve("xgboost", region, "1.2-1")
print(container)

xgb_model=sagemaker.estimator.Estimator(
    image_uri=container,
    role=role,
    instance_count=1,
    instance_type='ml.m4.xlarge',
    volume_size=5,
    output_path=s3_output_location,
    sagemaker_session=sagemaker.Session(),
    rules=[
        Rule.sagemaker(rule_configs.create_xgboost_report()),
        ProfilerRule.sagemaker(rule_configs.ProfilerReport())
    ]
)
```

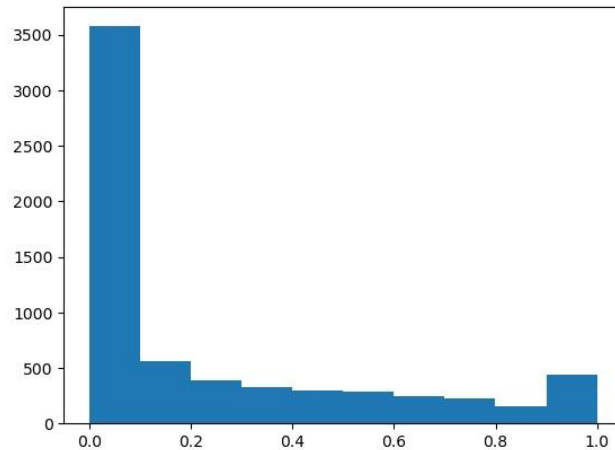
```
In [18]: xgb_predictor.endpoint_name
```

```
Out[18]: 'sagemaker-xgboost-2024-10-24-09-34-02-816'
```

```
In [19]: import numpy as np
def predict(data, rows=1000):
    split_array = np.array_split(data, int(data.shape[0] / float(rows) + 1))
    predictions = ''
    for array in split_array:
        predictions = ''.join([predictions, xgb_predictor.predict(array).decode('utf-8')])
    return np.fromstring(predictions[1:], sep=',')
```

```
In [20]: import matplotlib.pyplot as plt

predictions=predict(test.to_numpy()[1:,1:])
plt.hist(predictions)
plt.show()
```



```
In [21]: import sklearn
```

```
cutoff=0.5
print(sklearn.metrics.confusion_matrix(test.iloc[:, 0], np.where(predictions > cutoff, 1, 0)))
print(sklearn.metrics.classification_report(test.iloc[:, 0], np.where(predictions > cutoff, 1, 0)))
```

```
[[4670 356]
 [ 480 1007]]
      precision    recall  f1-score   support

     0       0.91      0.93      0.92     5026
     1       0.74      0.68      0.71     1487

 accuracy          0.82
 macro avg          0.80      0.81     6513
 weighted avg          0.87      0.87     6513
```

Upload [Info](#)

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (1 Total, 98.2 MB)

Remove

Add files

Add folder

All files and folders in this table will be uploaded.

<input type="text" value="creditcard"/>	X	1 match	< 1 >
<input type="checkbox"/> Name	▼	Folder	
<input type="checkbox"/> creditcard.csv		-	

Destination [Info](#)

Destination

[s3://mygroup2](#)

► Destination details

Bucket settings that impact new objects stored in the specified destination.

► Permissions

Grant public access and access to other AWS accounts.

► Properties

Specify storage class, encryption settings, tags, and more.

Cancel

Upload