

# SVKM'S NMIM'S Nilkamal School of Mathematics, Applied Statistics & Analytics Master of Science (Data Science)

## Practical-9 Sagemaker

Step 1 :login to the amazon console and go on the amazon sagemaker.



Step 2 : Then select domain and create a new domain and click on the set up.

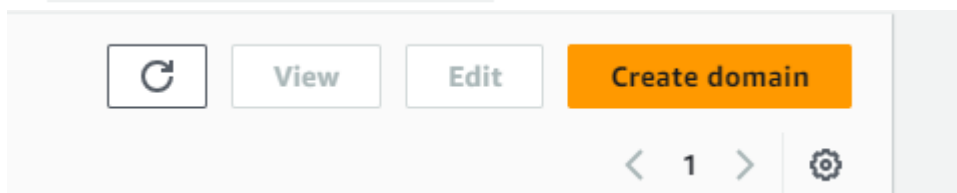
### ▼ Admin configurations

#### Domains

Role manager

Images

Lifecycle configurations



# Set up SageMaker Domain

Use SageMaker Domain as the central store to manage the configuration of SageMaker for your organization.

## Set up for single user (Quick setup)

Let Amazon SageMaker configure your account, and set up permissions for your SageMaker Domain.

- ✓ New IAM role with AmazonSageMakerFullAccess policy
- ✓ Public internet access, and standard encryption
- ✓ SageMaker Studio - New, and SageMaker Studio Classic integrations
- ✓ Sharable SageMaker Studio Notebooks
- ✓ SageMaker Canvas
- ✓ IAM Authentication

*Perfect for single user domains and first time users looking to get started with SageMaker.*

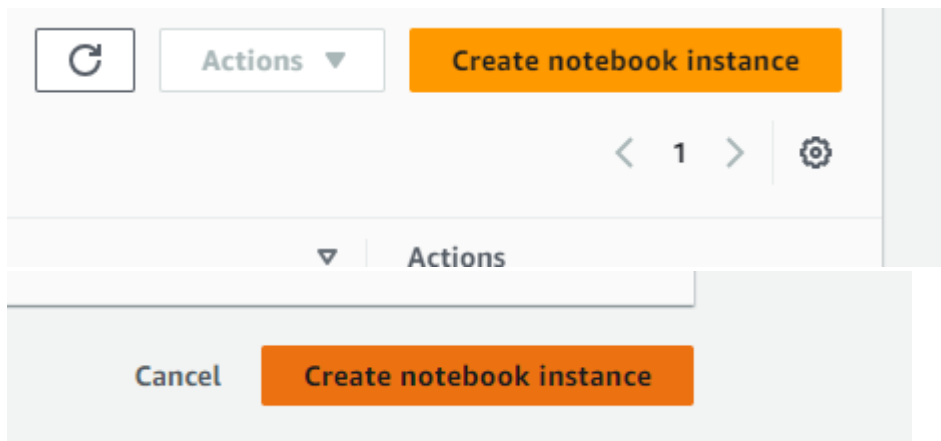
**Set up**

Step 3 : click on the notebook and create a new notebook instance.

### ▼ Notebook

Notebook instances

Git repositories



Notebook is successfully created.

Amazon SageMaker > Notebook instances

Notebook instances <a href="#">Info</a>					<a href="#">Refresh</a>	<a href="#">Actions</a>	<a href="#">Create notebook instance</a>
<input type="text" value="Search notebook instances"/>					<a href="#">&lt;</a> <a href="#">1</a> <a href="#">&gt;</a> <a href="#">⚙️</a>		
Name	Instance	Creation time	Status	Actions			
MLAlgorithm	ml.t3.medium	4/18/2024, 7:45:45 AM	InService	<a href="#">Open Jupyter</a>   <a href="#">Open JupyterLab</a>			

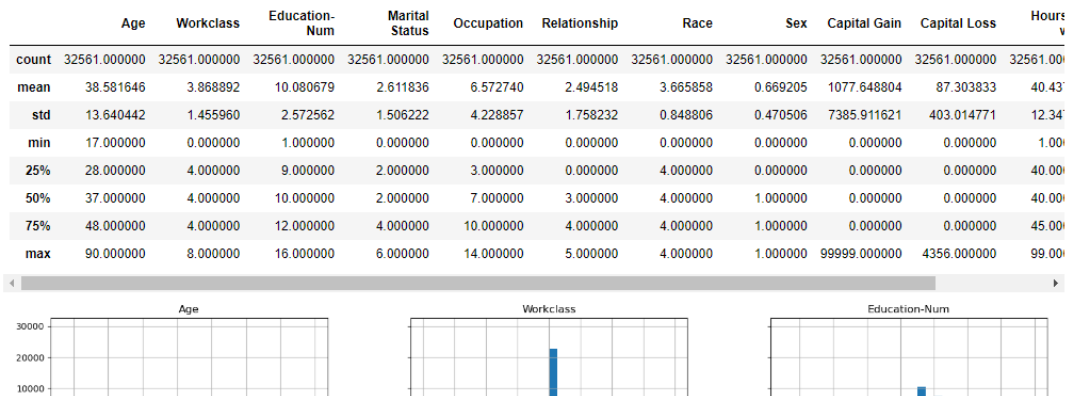
Step 4 : after succesfully creating a notebook instance open the python and train the model.

```
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 [2]: display(X.describe())
hist = X.hist(bins=30, sharey=True, figsize=(20, 10))
```



Step 5 : Deploy the model on amazon ec2.

```
In [12]: 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::167376188154:role/service-role/AmazonSageMakerServiceCatalogProductsUseRole
```

```
In [13]: 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())
    ]
)
```

Step 6 : after the model is successfully deployed.

```
In [22]: import sagemaker
from sagemaker.serializers import CSVSerializer
xgb_predictor=xgb_model.deploy(
    initial_instance_count=1,
    instance_type='ml.t2.medium',
    serializer=CSVSerializer()
)
```

```
INFO:sagemaker:Creating model with name: sagemaker-xgboost-2024-04-18-02-39-19-852
INFO:sagemaker:Creating endpoint-config with name sagemaker-xgboost-2024-04-18-02-39-19-852
INFO:sagemaker:Creating endpoint with name sagemaker-xgboost-2024-04-18-02-39-19-852
```

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

```
In [23]: xgb_predictor.endpoint_name
```

```
Out[23]: 'sagemaker-xgboost-2024-04-18-02-39-19-852'
```