Distributed and Parallel Computing Lab

CS461 Lab10

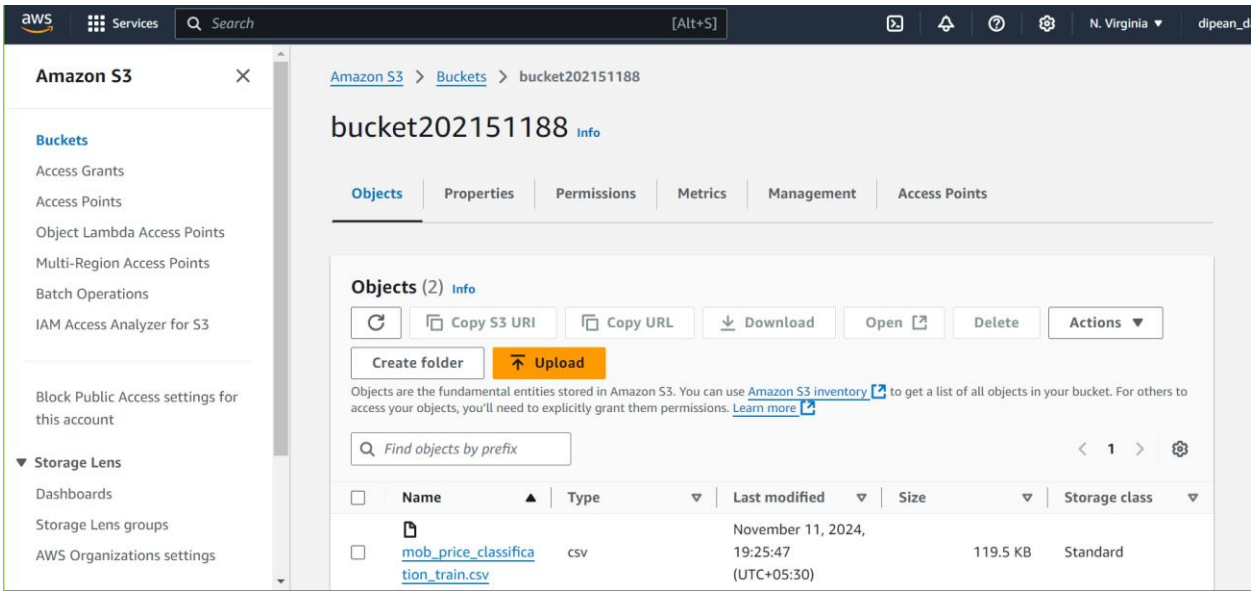
Name: Dipean Dasgupta **ID:202151188**

Task: Working with AWS and Google Colab Architectures

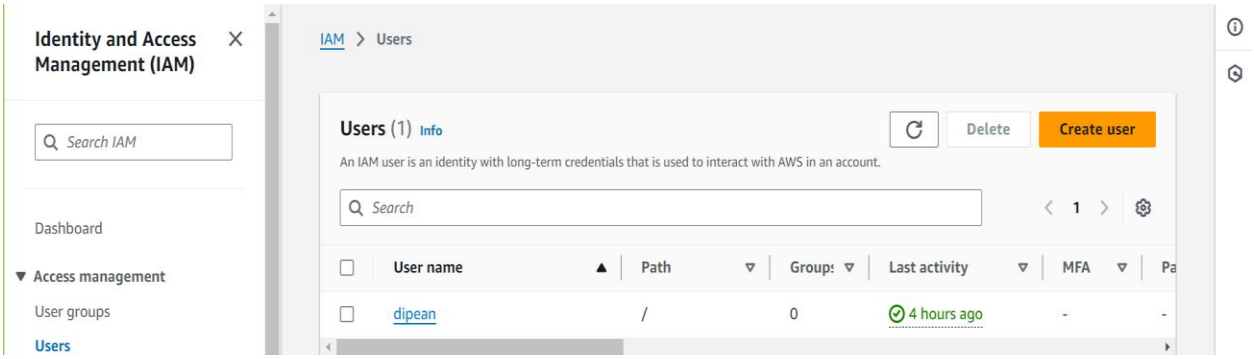
IDE: Jupyter Notebook[anaconda]

Program Language: Python **Cloud Platform:** AWS

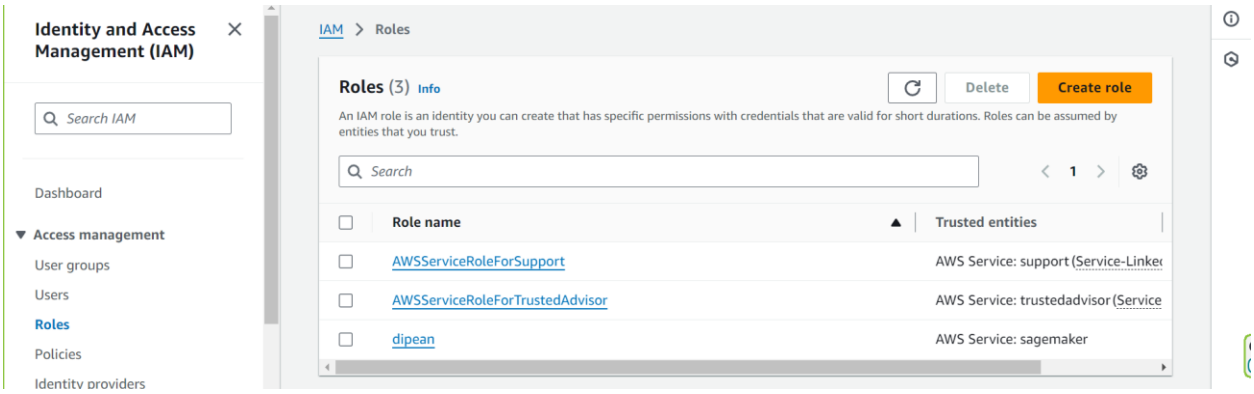
Setting Up AWS:



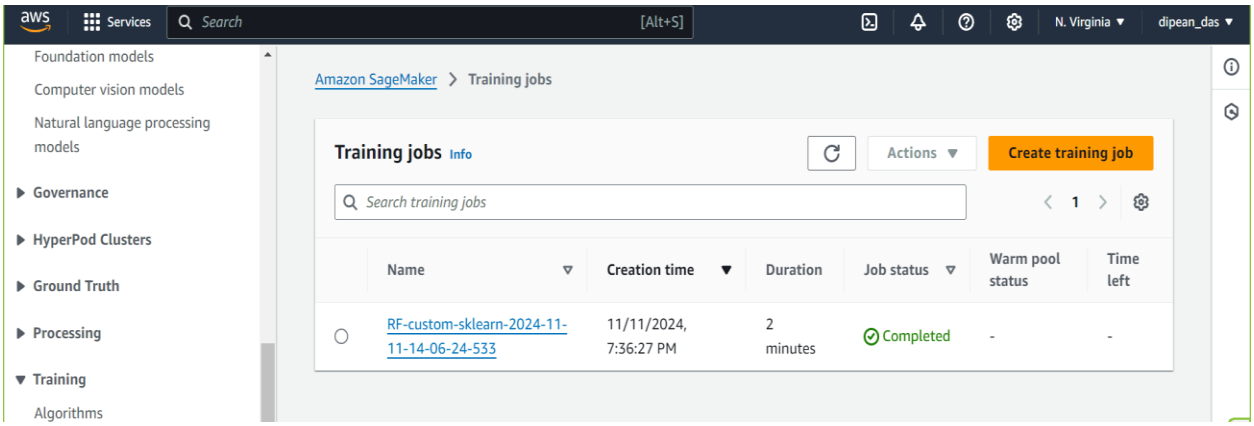
A bucket is created in amazon S3 named bucket202151188. The csv file for ML model training purpose is uploaded on the bucket.



User setup done in IAM (Identity and Access Management)



Roles in IAM are provided.



The ML model training job is completed through sagemaker in AWS successfully.

ML Model Train Codes execution:

Setting up the Environment for SageMaker

In [1]:
```
import sagemaker
from sklearn.model_selection import train_test_split
import boto3
import pandas as pd
import io
```

```
sagemaker.config INFO - Not applying SDK defaults from location: C:\ProgramData\sagemaker\sagemaker\config.yaml
sagemaker.config INFO - Not applying SDK defaults from location: C:\Users\Asus\AppData\Local\sagemaker\sagemaker\config.yaml
```

In [2]:
```
sm_boto3 = boto3.client("sagemaker")
sess = sagemaker.Session()
region = sess.boto_session.region_name
bucket = 'bucket202151188' # Created S3 bucket name here
file_key = 'mob_price_classification_train.csv'  # Path in your S3 bucket
print("Using bucket " + bucket)
```

```
Using bucket bucket202151188
```

Reading the Data from S3 Bucket

In [3]:
```
# Create an S3 object URI
s3_uri = f's3://bucket202151188/mob_price_classification_train.csv'
print("Using S3 URI:", s3_uri)
```

```
Using S3 URI: s3://bucket202151188/mob_price_classification_train.csv
```

Importing necessary python libraries and accessing model training file from bucket.

Performing the Train Test Split

In [10]:
```
X_train, X_test, y_train, y_test = train_test_split(x,y, test_size=0.15, random_state=0)

# print shapes of train and test dataframes
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(1700, 20)
(300, 20)
(1700,)
(300,)
```

In [11]:
```
# Create Train and Test dataframes to be stored for further use
trainX = pd.DataFrame(X_train)
trainX[label] = y_train

testX = pd.DataFrame(X_test)
testX[label] = y_test

# Print shape of new dataframes
print("Shape of Train dataframe",trainX.shape)
print("Shape of Test dataframe",testX.shape)
```

```
Shape of Train dataframe (1700, 21)
Shape of Test dataframe (300, 21)
```

In [12]:
```
# Save the train and test dataframes
trainX.to_csv("train-V-1.csv",index = False)
testX.to_csv("test-V-1.csv", index = False)
```

Performing the train test split.

Data Ingestion: Send the Train and Test CSV files to S3 bucket

In [13]:
```
# Print the bucket name
bucket
```

Out[13]:
```
'bucket202151188'
```

In [14]:
```
# Send data to S3. SageMaker will take training data from s3
sk_prefix = "sagemaker/mobile_price_classification/sklearncontainer"
trainpath = sess.upload_data(
    path="train-V-1.csv", bucket=bucket, key_prefix=sk_prefix
)

testpath = sess.upload_data(
    path="test-V-1.csv", bucket=bucket, key_prefix=sk_prefix
)
print(trainpath)
print(testpath)
```

```
s3://bucket202151188/sagemaker/mobile_price_classification/sklearncontainer/train-V-1.csv
s3://bucket202151188/sagemaker/mobile_price_classification/sklearncontainer/test-V-1.csv
```

Sending the train and test files to the S3 bucket successfully.

Training the model on Sagemaker

In [17]:
```
sklearn_estimator.fit({"train": trainpath, "test": testpath}, wait=True)
```

Model training in sagemaker.

OUTPUTS:

```
2024-11-11 14:06:31 Starting - Starting the training job...
2024-11-11 14:06:45 Starting - Preparing the instances for training...
2024-11-11 14:07:18 Downloading - Downloading input data...
2024-11-11 14:07:49 Downloading - Downloading the training image...
2024-11-11 14:08:35 Training - Training image download completed. Training in
progress.
```

```
2024-11-11 14:08:35 Uploading - Uploading generated training model2024-11-11
14:08:29,689 sagemaker-containers INFO     Imported framework
sagemaker_sklearn_container.training
2024-11-11 14:08:29,692 sagemaker-training-toolkit INFO     No GPUs detected
(normal if no gpus installed)
2024-11-11 14:08:29,735 sagemaker_sklearn_container.training INFO     Invoking
user training script.
2024-11-11 14:08:29,912 sagemaker-training-toolkit INFO     No GPUs detected
(normal if no gpus installed)
2024-11-11 14:08:29,925 sagemaker-training-toolkit INFO     No GPUs detected
(normal if no gpus installed)
2024-11-11 14:08:29,936 sagemaker-training-toolkit INFO     No GPUs detected
(normal if no gpus installed)
2024-11-11 14:08:29,945 sagemaker-training-toolkit INFO     Invoking user script
Training Env:
{
    "additional_framework_parameters": {},
    "channel_input_dirs": {
        "test": "/opt/ml/input/data/test",
        "train": "/opt/ml/input/data/train"
    },
    "current_host": "algo-1",
    "framework_module": "sagemaker_sklearn_container.training:main",
    "hosts": [
        "algo-1"
    ],
    "hyperparameters": {
        "n_estimators": 100,
        "random_state": 0
    },
    "input_config_dir": "/opt/ml/input/config",
    "input_data_config": {
        "test": {
            "TrainingInputMode": "File",
            "S3DistributionType": "FullyReplicated",
            "RecordWrapperType": "None"
        },
        "train": {
            "TrainingInputMode": "File",
            "S3DistributionType": "FullyReplicated",
            "RecordWrapperType": "None"
        }
    },
    "input_dir": "/opt/ml/input",
    "is_master": true,
```

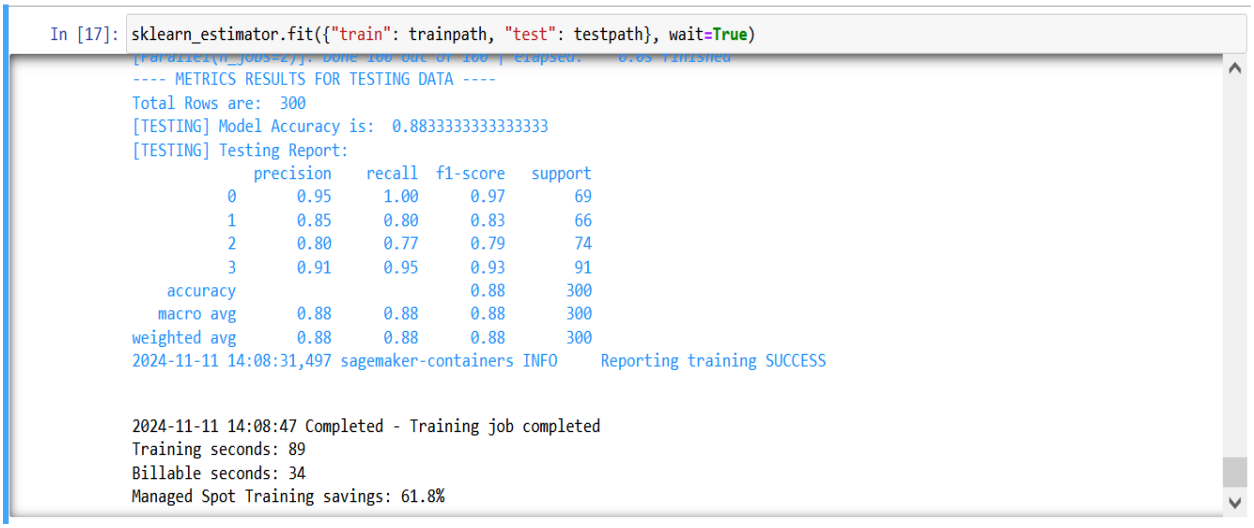
```
    "job_name": "RF-custom-sklearn-2024-11-11-14-06-24-533",
    "log_level": 20,
    "master_hostname": "algo-1",
    "model_dir": "/opt/ml/model",
    "module_dir": "",
    "module_name": "script",
    "network_interface_name": "eth0",
    "num_cpus": 2,
    "num_gpus": 0,
    "output_data_dir": "/opt/ml/output/data",
    "output_dir": "/opt/ml/output",
    "output_intermediate_dir": "/opt/ml/output/intermediate",
    "resource_config": {
        "current_host": "algo-1",
        "current_instance_type": "ml.m5.large",
        "current_group_name": "homogeneousCluster",
        "hosts": [
            "algo-1"
        ],
        "instance_groups": [
            {
                "instance_group_name": "homogeneousCluster",
                "instance_type": "ml.m5.large",
                "hosts": [
                    "algo-1"
                ]
            }
        ],
        "network_interface_name": "eth0"
    },
    "user_entry_point": "script.py"
}
Environment variables:
SM_HOSTS=["algo-1"]
SM_NETWORK_INTERFACE_NAME=eth0
SM_HPS={"n_estimators":100,"random_state":0}
SM_USER_ENTRY_POINT=script.py
SM_FRAMEWORK_PARAMS={}
SM_RESOURCE_CONFIG={"current_group_name":"homogeneousCluster","current_host":"alg
o-1","current_instance_type":"ml.m5.large","hosts":["algo-
1"],"instance_groups":[{"hosts":["algo-
1"],"instance_group_name":"homogeneousCluster","instance_type":"ml.m5.large"}],"n
etwork_interface_name":"eth0"}
```

```
SM_INPUT_DATA_CONFIG={"test":{"RecordWrapperType":"None","S3DistributionType":"FullyReplicated","TrainingInputMode":"File"},"train":{"RecordWrapperType":"None","S3DistributionType":"FullyReplicated","TrainingInputMode":"File"}}
SM_OUTPUT_DATA_DIR=/opt/ml/output/data
SM_CHANNELS=["test","train"]
SM_CURRENT_HOST=algo-1
SM_MODULE_NAME=script
SM_LOG_LEVEL=20
SM_FRAMEWORK_MODULE=sagemaker_sklearn_container.training:main
SM_INPUT_DIR=/opt/ml/input
SM_INPUT_CONFIG_DIR=/opt/ml/input/config
SM_OUTPUT_DIR=/opt/ml/output
SM_NUM_CPUS=2
SM_NUM_GPUS=0
SM_MODEL_DIR=/opt/ml/model
SM_MODULE_DIR=s3://sagemaker-us-east-1-266735828925/RF-custom-sklearn-2024-11-11-14-06-24-533/source/sourcedir.tar.gz
SM_TRAINING_ENV={"additional_framework_parameters":{},"channel_input_dirs":{"test":"/opt/ml/input/data/test","train":"/opt/ml/input/data/train"},"current_host":"algo-
1","framework_module":"sagemaker_sklearn_container.training:main","hosts":["algo-1"],"hyperparameters":{"n_estimators":100,"random_state":0},"input_config_dir":"/opt/ml/input/config","input_data_config":{"test":{"RecordWrapperType":"None","S3DistributionType":"FullyReplicated","TrainingInputMode":"File"},"train":{"RecordWrapperType":"None","S3DistributionType":"FullyReplicated","TrainingInputMode":"File"}},"input_dir":"/opt/ml/input","is_master":true,"job_name":"RF-custom-sklearn-2024-11-11-14-06-24-533","log_level":20,"master_hostname":"algo-
1","model_dir":"/opt/ml/model","module_dir":"s3://sagemaker-us-east-1-
266735828925/RF-custom-sklearn-2024-11-11-14-06-24-
533/source/sourcedir.tar.gz","module_name":"script","network_interface_name":"eth0","num_cpus":2,"num_gpus":0,"output_data_dir":"/opt/ml/output/data","output_dir":"/opt/ml/output","output_intermediate_dir":"/opt/ml/output/intermediate","resource_config":{"current_group_name":"homogeneousCluster","current_host":"algo-
1","current_instance_type":"ml.m5.large","hosts":["algo-
1"],"instance_groups":[{"hosts":["algo-
1"],"instance_group_name":"homogeneousCluster","instance_type":"ml.m5.large"}],"network_interface_name":"eth0"},"user_entry_point":"script.py"}
SM_USER_ARGS=["--n_estimators","100","--random_state","0"]
SM_OUTPUT_INTERMEDIATE_DIR=/opt/ml/output/intermediate
SM_CHANNEL_TEST=/opt/ml/input/data/test
SM_CHANNEL_TRAIN=/opt/ml/input/data/train
SM_HP_N_ESTIMATORS=100
SM_HP_RANDOM_STATE=0
```

```
PYTHONPATH=/opt/ml/code:/miniconda3/bin:/miniconda3/lib/python37.zip:/miniconda3/
lib/python3.7:/miniconda3/lib/python3.7/lib-
dynload:/miniconda3/lib/python3.7/site-packages
Invoking script with the following command:
/miniconda3/bin/python script.py --n_estimators 100 --random_state 0
[INFO] Extracting arguments
SKLearn Version:  0.23.2
Joblib Version:  1.2.0
[INFO] Reading data
Building training and testing datasets
Column order:
['battery_power', 'blue', 'clock_speed', 'dual_sim', 'fc', 'four_g',
'int_memory', 'm_dep', 'mobile_wt', 'n_cores', 'pc', 'px_height', 'px_width',
'ram', 'sc_h', 'sc_w', 'talk_time', 'three_g', 'touch_screen', 'wifi']
Label column is:  price_range
Data Shape:
---- SHAPE OF TRAINING DATA (85%) ----
(1700, 20)
(1700,)
---- SHAPE OF TESTING DATA (15%) ----
(300, 20)
(300,)
Training RandomForest Model.....
[Parallel(n_jobs=-1)]: Using backend ThreadingBackend with 2 concurrent workers.
building tree 1 of 100building tree 2 of 100
building tree 3 of 100
building tree 4 of 100
building tree 5 of 100
building tree 6 of 100
building tree 7 of 100
building tree 8 of 100
building tree 9 of 100
building tree 10 of 100
building tree 11 of 100
building tree 12 of 100
building tree 13 of 100
building tree 14 of 100
building tree 15 of 100
building tree 16 of 100
building tree 17 of 100
building tree 18 of 100
building tree 19 of 100
building tree 20 of 100
building tree 21 of 100
building tree 22 of 100
```

```
building tree 23 of 100
building tree 24 of 100
building tree 25 of 100
building tree 26 of 100
building tree 27 of 100
building tree 28 of 100
building tree 29 of 100
building tree 30 of 100
[Parallel(n_jobs=-1)]: Done  28 tasks      | elapsed:    0.1s
building tree 31 of 100
      ---------
building tree 100 of 100
[Parallel(n_jobs=-1)]: Done 100 out of 100 | elapsed:    0.3s finished
Model persisted at /opt/ml/model/model.joblib
[Parallel(n_jobs=2)]: Using backend ThreadingBackend with 2 concurrent workers.
[Parallel(n_jobs=2)]: Done  28 tasks      | elapsed:    0.0s
[Parallel(n_jobs=2)]: Done 100 out of 100 | elapsed:    0.0s finished
```

Model Training Report:



Model is trained and its having accuracy of 88.3%. Other parameters like precision, recall, f1-score and support values are also shared in the testing report.

----------END OF ASSIGNMENT----------