CS643 Programming Assignment 2

Sathvik Reddy Thogaru

My GitHub repository link: https://github.com/sathvikreddy25/winePredict

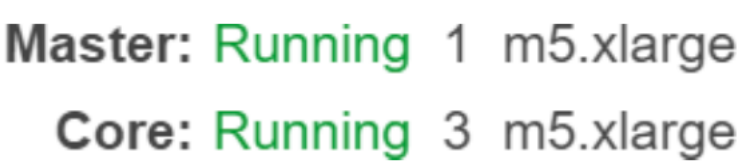
Docker link: https://hub.docker.com/repository/docker/sathvikreddy968/predictingwine/general

Objectives

- to use Apache Spark to train an ML model in parallel on multiple EC2 instances
- to use Spark's MLlib to develop and use an ML model in the cloud
- to use Docker to create a container for your ML model to simplify model deployment

Setup

First we have to create EMR cluster using AWS management console We should have 4 EC2 instances and using Spark for the ML application
In the 4 Ec2 instances we have one master node and 3 core nodes
After setting up our cluster we need to create S3 bucket which acts as storage and helps to fetch the required data faster and send output to that bucket
One bucket is created for the logs of the cluster and the other for storing our data



Model training and Application Prediction

Now connect to the master instance using any SSH client . I am using Putty to connect to the master node. Then using WinSCP copy the spark application to run on the cluster

General purpose buckets (2) Info All AWS Regions

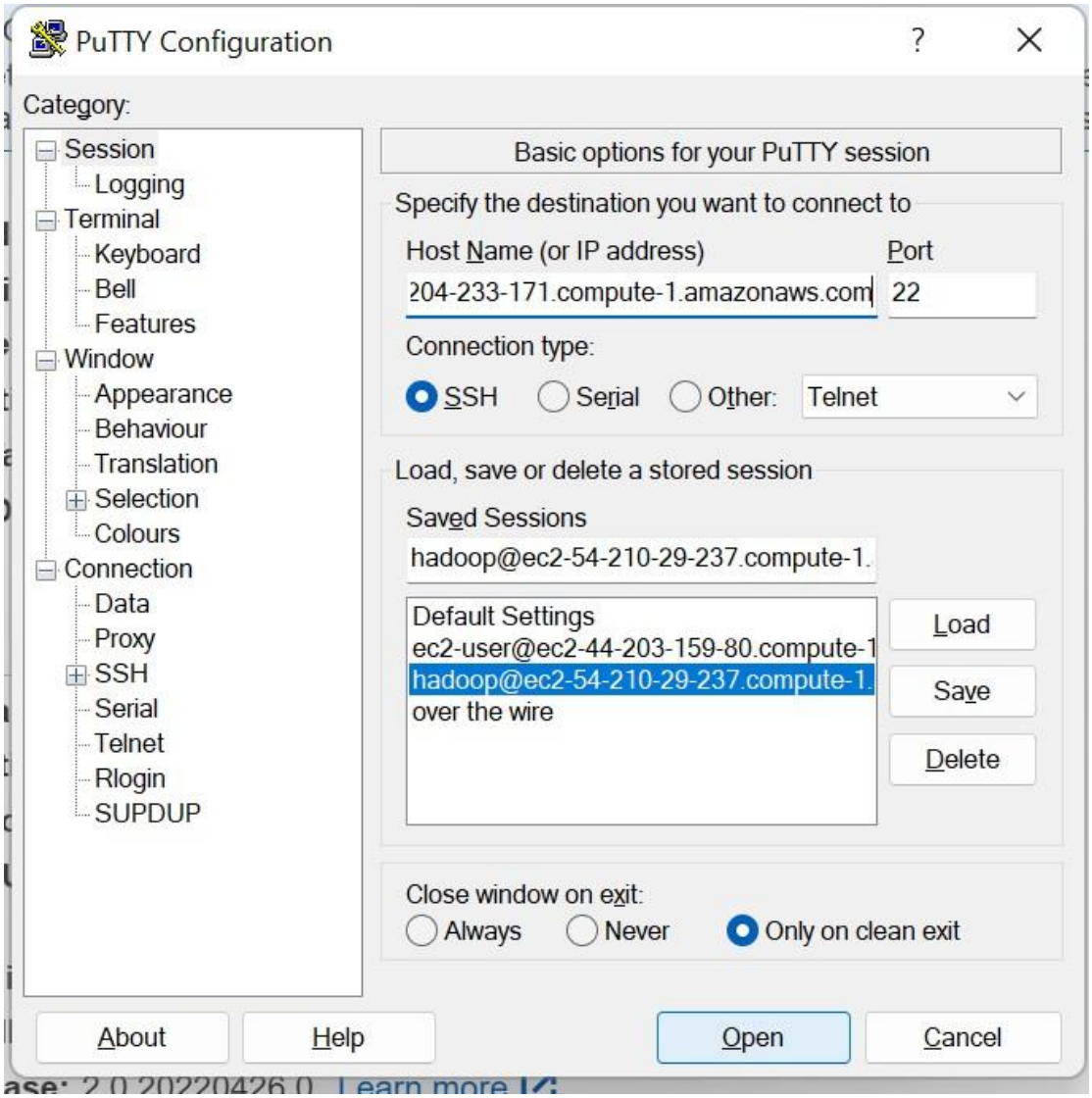
Copy ARN | Empty | Delete | **Create bucket**

Buckets are containers for data stored in S3.

Find buckets by name

1

Name	AWS Region	IAM Access Analyzer	Creation date
aws-logs-us-east-1-cloud-computing	US East (N. Virginia) us-east-1	View analyzer for us-east-1	April 28, 2024, 23:33:57 (UTC+05:30)
my-bucket-source-cs643	US East (N. Virginia) us-east-1	View analyzer for us-east-1	April 28, 2024, 23:35:03 (UTC+05:30)

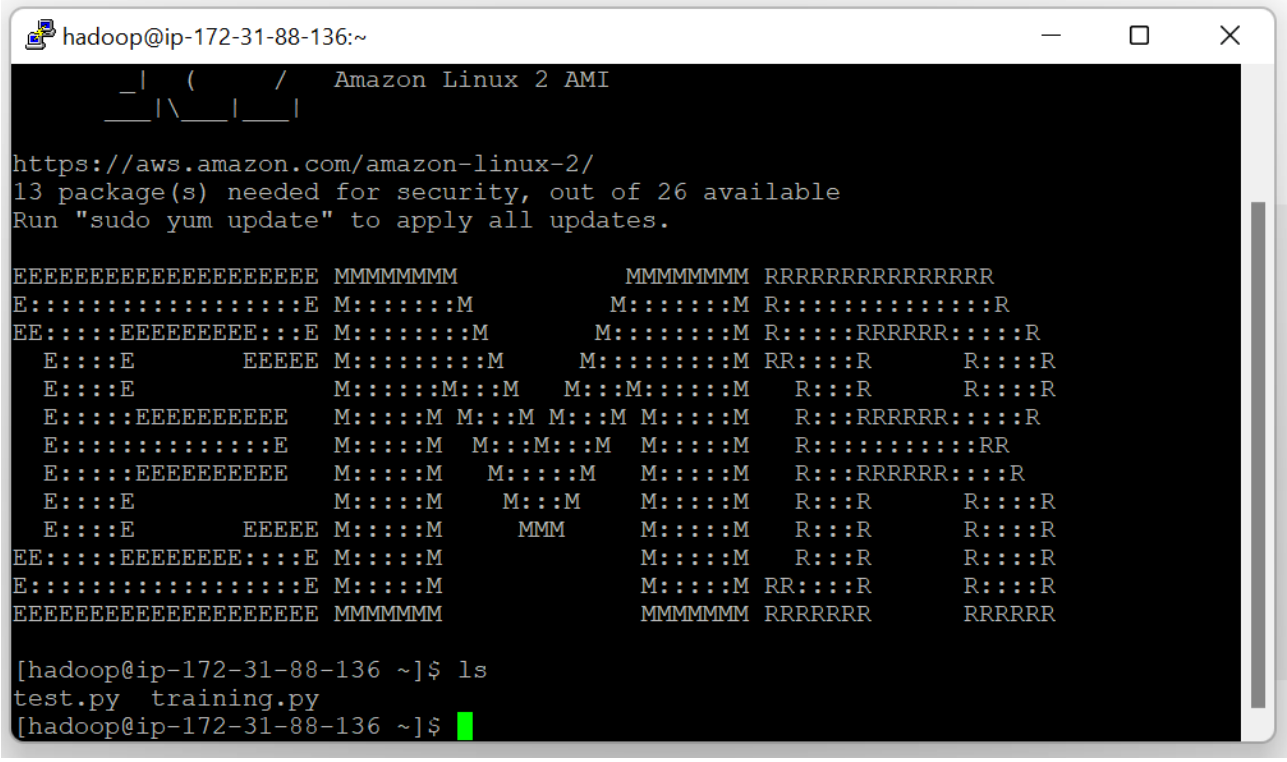


In the home directory i have my training.py file which i am using to train my model using the TrainingDataset.csv

By using the command ‘spark-submit training.py’ we can run the application on multiple clusters with the help of serialize

The s3 bucket URl is given in the document to fetch the files from the S3 bucket and for saving the model

I have also used ValidationDataset.csv to adjust my hyperparameters and tune it to highest score possible



Results

Code tested ValidationDataset.csv and the scores are as below

The F1- score we are seeing is the highest score achieved using RandomForestTrees We can also see scores of other models like LogisticRegression, NaiveBayes, gradientBoostedTrees which are comparatively lowers than RandomForestTrees

```
model accuracy 0.40625
LR F1 Score = 0.53125
F1- score:  0.75
[[10  2  0  0]
 [ 2 10  1  0]
 [ 1  2  3  0]
 [ 0  0  0  1]]
              precision    recall  f1-score   support

         5.0       0.77      0.83      0.80        12
         6.0       0.71      0.77      0.74        13
         7.0       0.75      0.50      0.60         6
         8.0       1.00      1.00      1.00         1

    accuracy                           0.75        32
   macro avg       0.81      0.78      0.79        32
weighted avg       0.75      0.75      0.74        32

Accuracy 0.75
```

Now we have to create Docker container and run our application on it using the docker file created

We use following commands to run docker container using the docker image
docker build -t docker-ml-model -f Dockerfile .
docker run docker-ml-model

After creating our docker successfully we can run our prediction test

Screenshots for pushing images to docker hub

```
C:\Users\pranavtalanki\Desktop\predictingwine>docker build -t predictingwine .
[+] Building 0.0s (0/0)  docker:default
[+] Building 2.0s (13/13) FINISHED                                                                                                  docker:default
 => [internal] load build definition from Dockerfile                                                                                          0.0s
 => => transferring dockerfile: 2.19kB                                                                                                        0.0s
 => [internal] load metadata for docker.io/library/openjdk:8-jre-slim                                                                         1.8s
 => [auth] library/openjdk:pull token for registry-1.docker.io                                                                                0.0s
 => [internal] load .dockerignore                                                                                                             0.0s
 => => transferring context: 2B                                                                                                               0.0s
 => [1/7] FROM docker.io/library/openjdk:8-jre-slim@sha256:53186129237fbb8bc0a12dd36da6761f4c7a2a20233c20d4eb0d497e4045a4f5                    0.0s
 => [internal] load build context                                                                                                             0.0s
 => => transferring context: 435B                                                                                                             0.0s
 => CACHED [2/7] RUN set -ex &&         apt-get update &&     apt-get install -y curl bzip2 --no-install-recommends &&     curl -s -L --url "https://  0.0s
 => CACHED [3/7] COPY test.py /winepredict/                                                                                                   0.0s
 => CACHED [4/7] COPY ValidationDataset.csv /winepredict/                                                                                     0.0s
 => CACHED [5/7] COPY trainingmodel /winepredict/                                                                                             0.0s
 => CACHED [6/7] COPY trainingmodel/ /winepredict/trainingmodel                                                                               0.0s
 => CACHED [7/7] ADD test.py .                                                                                                                0.0s
 => exporting to image                                                                                                                        0.0s
 => => exporting layers                                                                                                                       0.0s
 => => writing image sha256:fb3d5b1d2a6a63503e3a3aace82ef8183d5565a436fb39bef7a6baa8b319ee48                                                  0.0s
 => => naming to docker.io/library/predictingwine                                                                                             0.0s

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview

C:\Users\pranavtalanki\Desktop\predictingwine>docker images
REPOSITORY       TAG       IMAGE ID       CREATED         SIZE
quelpred         latest    fb3d5b1d2a6a   3 minutes ago   1.41GB
predictingwine   latest    fb3d5b1d2a6a   3 minutes ago   1.41GB

C:\Users\pranavtalanki\Desktop\predictingwine>docker delete rmi quelpred
docker: 'delete' is not a docker command.
See 'docker --help'

C:\Users\pranavtalanki\Desktop\predictingwine>docker rmi quelpred:latest
Untagged: quelpred:latest
```

```
C:\Users\pranavtalanki\Desktop\predictingwine>docker images
REPOSITORY       TAG       IMAGE ID       CREATED         SIZE
quelpred         latest    fb3d5b1d2a6a   3 minutes ago   1.41GB
predictingwine   latest    fb3d5b1d2a6a   3 minutes ago   1.41GB

C:\Users\pranavtalanki\Desktop\predictingwine>docker delete rmi quelpred
docker: 'delete' is not a docker command.
See 'docker --help'

C:\Users\pranavtalanki\Desktop\predictingwine>docker rmi quelpred:latest
Untagged: quelpred:latest

C:\Users\pranavtalanki\Desktop\predictingwine>docker images
REPOSITORY       TAG       IMAGE ID       CREATED         SIZE
predictingwine   latest    fb3d5b1d2a6a   4 minutes ago   1.41GB

C:\Users\pranavtalanki\Desktop\predictingwine>docker push sathvikreddy968/predictingwine:latest
The push refers to repository [docker.io/sathvikreddy968/predictingwine]
An image does not exist locally with the tag: sathvikreddy968/predictingwine

C:\Users\pranavtalanki\Desktop\predictingwine>docker push sathvikreddy968/predictingwine:tagname
The push refers to repository [docker.io/sathvikreddy968/predictingwine]
An image does not exist locally with the tag: sathvikreddy968/predictingwine

C:\Users\pranavtalanki\Desktop\predictingwine>docker tag predictingwine:latest

C:\Users\pranavtalanki\Desktop\predictingwine>docker tag predictingwine:latest sathvikreddy968/predictingwine:latest

C:\Users\pranavtalanki\Desktop\predictingwine>docker push sathvikreddy968/predictingwine:tagname
The push refers to repository [docker.io/sathvikreddy968/predictingwine]
tag does not exist: sathvikreddy968/predictingwine:tagname
```

```
C:\Users\pranavtalanki\Desktop\predictingwine>docker push sathvikreddy968/predictingwine:tagname
The push refers to repository [docker.io/sathvikreddy968/predictingwine]
tag does not exist: sathvikreddy968/predictingwine:tagname

C:\Users\pranavtalanki\Desktop\predictingwine>docker push sathvikreddy968/predictingwine:latest
The push refers to repository [docker.io/sathvikreddy968/predictingwine]
5cbe602fbdbb: Pushed
04a7a8f27da1: Pushed
3a62265466ea: Pushed
728c8a35de40: Pushed
0b2604e31ab9: Pushed
4e8e8ab1de30: Pushed
b66078cf4b41: Mounted from library/openjdk
cd5a0a9f1e01: Mounted from library/openjdk
eafe6e032dbd: Mounted from library/openjdk
92a4e8a3140f: Mounted from library/openjdk
latest: digest: sha256:03c90c527207238da66f0a4dd0b31a273dbe69f5c8da4adb396c811729f8e42e size: 2418
```