CS643 Cloud Computing
Programming Assignment 2

Dhruvi Makadia

- The purpose of this assignment is gain experience in developing a parallel machine (ML) applications in AWS cloud. This project is done to train a machine learning model parallelly on ec2 instances for predicting wine quality on a available data and then use the trained model to predict the wine quality.
- The link to GitHub repository(https://github.com/dhruvi1996/wine-quality-prediction.git) shows the source codes which reads the training dataset from s3 and trains model in parallel on EMR spark cluster.
- Dockerfile is mainly used to create a docker image and for running the container present. The link to docker image in docker hub is : (https://hub.docker.com/repository/docker/dhruvi11/aws-winepredictionmodel).
  [Docker image : dhruvi11/aws-winepredictionmodel]

Requirements :

- Input files must be copied under data/ and with same name TestDataset.csv
- Input file must have save format as of TrainingDataset.csv and ValidationDataset.csv.
- Column names in double quotes with exact name number of column with same names and separator ';'

Following are the steps to be followed inorder to get the output for wine quality prediction.

TASK 1 : How to create Spark cluster in AWS

As given in our guidelines, for performance of this project we must use Spark data frames api and MLib libraries. So, running the Spark dataframes along with MLib libraries on AWS cluster (EMR) hence as a matter of course it will parallelize and will distribute job execution accordingly.

How to create Spark cluster in AWS?

1) After logging into AWS console, user can create cluster using EMR console provided by AWS. User has go to the tool bar and choose EMR; create a cluster by entering name.

2) Following are the steps to create one with 4 ec2 instances

EC2-> Network & Security -> Key-pairs

First create a Key-pair for EMR cluster and download .pem key. We will need it later.

3) Second we go to EMR console there we Create cluster

- Enter cluster name
- Vendor Amazon
- Release emr-5.3.10
- Hardware Configurations:
    1. select the instance type
    2. 1 master 3 slaves total 4 number of instances
- Security Access -> select the ec2 key pair or generate one to access the master node.
- click on create cluster.

4) Uploading files to EMR . Cluster status should be 'Waiting' on successful cluster creation. Now, copy the master node dns address(eg: hadoop@ec2*****us-east-2.@amazon.com) and on local machine terminal paste it.

- Open sftp connection to master node:

  ```
  sftp -i cluster-keypair.pem hadoop@ec2-3-127-26-180.us-east-2.compute.amazonaws.com
  ```
- Upload TrainingDataSet , ValidationDataset to master node.(remember to make it appropriate in csv format).

Task 2 : How to train ML model in Spark cluster with 4 ec2 instances in parallel

1) Now when cluster is ready to accept jobs, to submit one you can either use step button to add steps or submit manually. To submit manually, Perform SSH to Master of cluster using below command:

- ssh -i "ec2key.pem" <<User>>@<<Public IPv4 DNS>>

2) We will copy both of our validation and training files now to HDFS, we will be doing this so that the 3 slave nodes can gain access to these files with the commands:

- hadoop fs -putTrainingDataset.csv/user/hadoop/TrainingDataset.csv
- hadoop fs -put ValidationDataset.csv/user/hadoop/ValidationDataset.csv

3) Using the following command we will be able to launch modeltrainer application on EMR cluster.

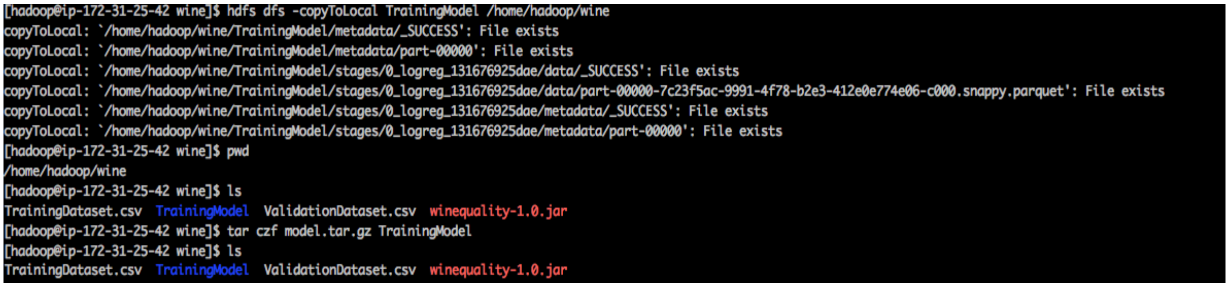
- spark-submit winequality-1.0.jar

Move ahead, by copying this folder to back its master node with the given command:

- hdfs dfs -copyToLocal TrainingModel /home/hadoop/wine

4) In order to use these training files all of them should be in a zip and transferred to local machine environment which we can use to predict on ec2. Make a tar.gz zip of folder by shown command:

- tar czf model.tar.gz TrainingModel



```
[hadoop@ip-172-31-25-42 wine]$ hdfs dfs -copyToLocal TrainingModel /home/hadoop/wine
copyToLocal: `/home/hadoop/wine/TrainingModel/metadata/_SUCCESS': File exists
copyToLocal: `/home/hadoop/wine/TrainingModel/metadata/part-00000': File exists
copyToLocal: `/home/hadoop/wine/TrainingModel/stages/0_logreg_131676925dae/data/_SUCCESS': File exists
copyToLocal: `/home/hadoop/wine/TrainingModel/stages/0_logreg_131676925dae/data/part-00000-7c23f5ac-9991-4f78-b2e3-412e0e774e06-c000.snappy.parquet': File exists
copyToLocal: `/home/hadoop/wine/TrainingModel/stages/0_logreg_131676925dae/metadata/_SUCCESS': File exists
copyToLocal: `/home/hadoop/wine/TrainingModel/stages/0_logreg_131676925dae/metadata/part-00000': File exists
[hadoop@ip-172-31-25-42 wine]$ pwd
/home/hadoop/wine
[hadoop@ip-172-31-25-42 wine]$ ls
TrainingDataset.csv  TrainingModel  ValidationDataset.csv  winequality-1.0.jar
[hadoop@ip-172-31-25-42 wine]$ tar czf model.tar.gz TrainingModel
[hadoop@ip-172-31-25-42 wine]$ ls
TrainingDataset.csv  TrainingModel  ValidationDataset.csv  winequality-1.0.jar
```

Task 3: How to run trained ML model locally without docker.

We will try to execute the code for prediction on EC2 instances: 1.) So, common step is to create Ec2 instance:

- On AWS console, we go to EC2 ---> Launch instances
- Select key-pair and launch it.

2.) After doing EC2 instance configuration we install Scala with the help of commands like:

- wget http://downloads.typesafe.com/scala/2.11.6/scala-2.11.6.tgz
- tar -xzvf scala-2.11.6.tgz
- With doing vim ~/.bashrc , source ~/.bash_profile. we will update our PATH environment

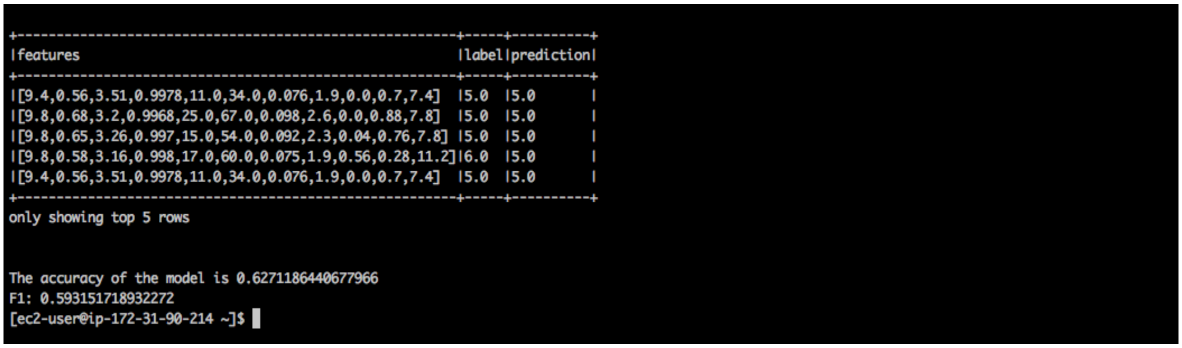
3) We also install Spark and setup with the help of given link in the homework description. (https://spark.apache.org/docs/latest/)

- We have to make sure that spark environment is setup locally so here also we will make changes in vim ~/.bash_profile file , source ~/.bash_profile file.

4.) with the use of EC2 instances I upload jar file and TestDataset.csv.

Run wine-predict application:

- spark-submit wine-quality-predict-1.0.jar
- you will see the output score on ec2 as shown below:



```
+--------------------------------------------------------+-----+----------+
|features                                                |label|prediction|
+--------------------------------------------------------+-----+----------+
|[9.4,0.56,3.51,0.9978,11.0,34.0,0.076,1.9,0.0,0.7,7.4]  |5.0  |5.0       |
|[9.8,0.68,3.2,0.9968,25.0,67.0,0.098,2.6,0.0,0.88,7.8]  |5.0  |5.0       |
|[9.8,0.65,3.26,0.997,15.0,54.0,0.092,2.3,0.04,0.76,7.8] |5.0  |5.0       |
|[9.8,0.58,3.16,0.998,17.0,60.0,0.075,1.9,0.56,0.28,11.2]|6.0  |5.0       |
|[9.4,0.56,3.51,0.9978,11.0,34.0,0.076,1.9,0.0,0.7,7.4]  |5.0  |5.0       |
+--------------------------------------------------------+-----+----------+
only showing top 5 rows


The accuracy of the model is 0.6271186440677966
F1: 0.593151718932272
[ec2-user@ip-172-31-90-214 ~]$ 
```

Task 4 : Predict wine quality using docker

1.) Using docker to predict wine quality we have to local path to the file as TestDataset.csv which can be put as input while running a docker command. Place your TestDataset.csv file in a folder (let's call it directory data), which you will mount with docker container.

2.) Doing this the file named TestDataset.csv will get copied to local docker environment.

3.) Now, all we must do is to run command.

Execute following command:

- docker pull dhruvi11/aws-winepredictionmodel

- docker run --platform=linux/amd64 dhruvi11/aws-winepredictionmodel testFilePath
- The above docker code mentions specific platform to run my docker container image.
- Sample command :
  docker run -v [local_testfile_directory:/data] dhruvi11/aws-winepredictionmodel/TestDataset.csv
- For some reason sometimes mac with M1 chip doesn't allow to pull images so we can go online to 'docker playground' pull the images and then run it.

4.) We can use docker playground link to docker playground is : (https://labs.play-with-docker.com/) We sign in with docker hub username and password.

- Next is we start a instance and there in instance we can see terminal pop up with root user.
- Run command :

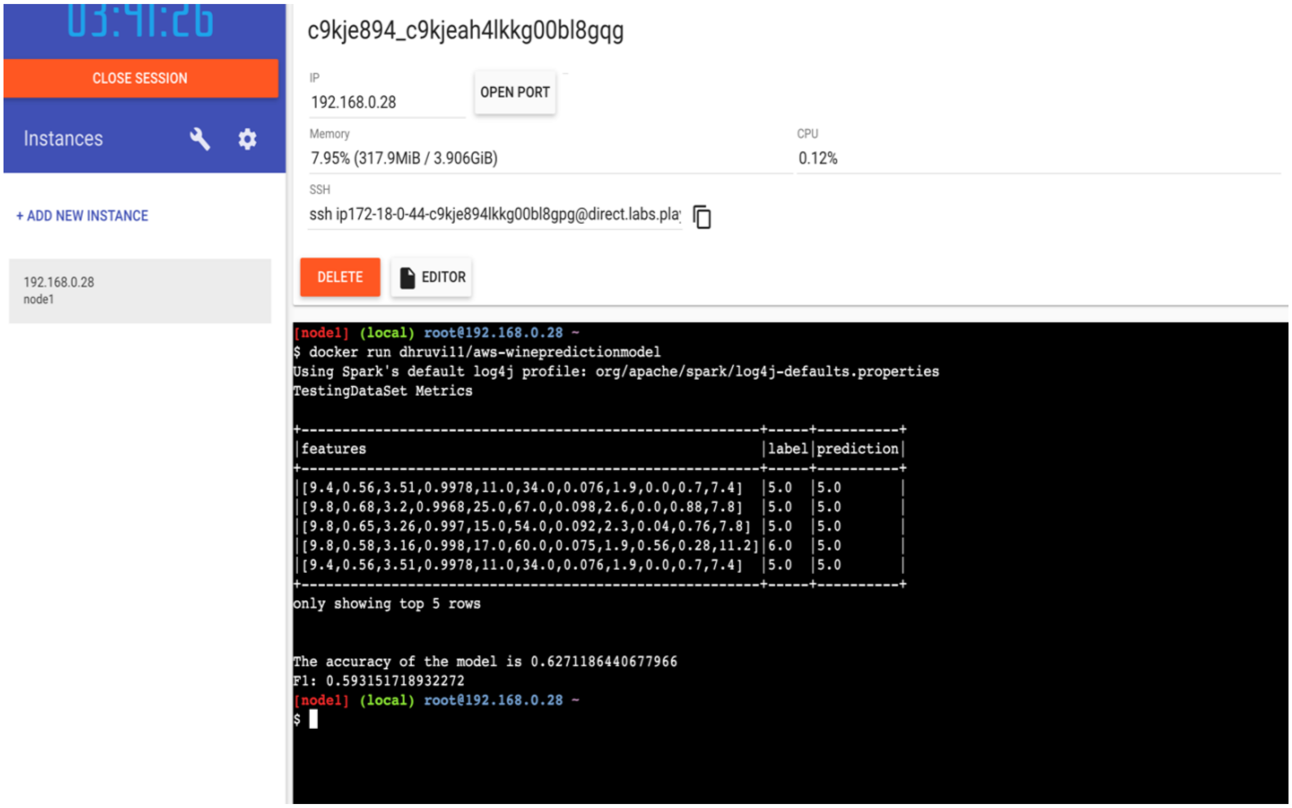
  docker run dhruvi11/aws-winepredictionmodel

  or

  docker run --platform=linux/amd64 dhruvi11/aws-winepredictionmodel testFilePath

- And instantly you can see the results.

- As we can see in below screenshot, Scores are printed on local machine environment with the help of docker.



c9kje894_c9kjeah4lkkg00bl8gqg

IP 192.168.0.28

Memory 7.95% (317.9MiB / 3.906GiB)

CPU 0.12%

SSH ssh ip172-18-0-44-c9kje894lkkg00bl8gpg@direct.labs.pla

```
[node1] (local) root@192.168.0.28 ~
$ docker run dhruvill/aws-winepredictionmodel
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
TestingDataSet Metrics

+--------------------------------------------------------+-----+----------+
|features                                                |label|prediction|
+--------------------------------------------------------+-----+----------+
|[9.4,0.56,3.51,0.9978,11.0,34.0,0.076,1.9,0.0,0.7,7.4]  |5.0  |5.0       |
|[9.8,0.68,3.2,0.9968,25.0,67.0,0.098,2.6,0.0,0.88,7.8]  |5.0  |5.0       |
|[9.8,0.65,3.26,0.997,15.0,54.0,0.092,2.3,0.04,0.76,7.8] |5.0  |5.0       |
|[9.8,0.58,3.16,0.998,17.0,60.0,0.075,1.9,0.56,0.28,11.2]|6.0  |5.0       |
|[9.4,0.56,3.51,0.9978,11.0,34.0,0.076,1.9,0.0,0.7,7.4]  |5.0  |5.0       |
+--------------------------------------------------------+-----+----------+
only showing top 5 rows


The accuracy of the model is 0.6271186440677966
F1: 0.593151718932272
[node1] (local) root@192.168.0.28 ~
$
```

-------------------------END---------------------