CS643 Cloud Computing

Programming Assignment 2

Name: Milankumar Patel

NJIT UCID: mkp6

Email: mkp6@njit.edu

- GitHub link to code and ReadMe file:

  https://github.com/Milan-36/winequalityprediction

- Docker container link:

  https://hub.docker.com/repository/docker/milankumarpatel/winequalityprediction

- Please follow the steps from the following README.md file to set-up the cloud environment and run the wine quality prediction ML model in Spark over Amazon AWS cloud platform and the application prediction:

  https://github.com/Milan-36/winequalityprediction/blob/main/README.md

1. General Info

Name: Milankumar Patel

NJIT UCID: MKP6

CS643 - Cloud Computing - Fall 2022

2. Description

The purpose of Programming Assignment 2 is to build a wine quality prediction ML model in Spark over Amazon AWS cloud platform. This project is done in Java on Amazon EMR (Amazon Elastic MapReduce).

2.1) AWS Services and other resours that I have used for this project:

- Amazon EMR (Amazon Elastic MapReduce) : one Master EC2 Instance and three Slave EC2 Instances.
- One EC2 instances (`EC2_Validation` for testing the wine quality prediction ML model on Validation dataset): AMI: Ubuntu 22.04 LTS (HVM)
- TrainingDataset.csv
- ValidationDataset.csv
- Docker container

2.2) General Workflow:

- Trained wine quality prediction ML model in paralle using AWS EMR's four EC2 instances. For training ML model, I'm using TrainingDataset.csv in parallel on four EC2 instances.
- Created another EC2 instace to save the trained model and load it in the spark to perform predication on it. To validate of ML model and to check the performace of the trained model, I'm using ValidationDataset.csv.
- Build a Docker container for prediction model. So, the prediction model can be quickly deployed across many different environments.

3. Instructions

3.1) Create a AWS EMR Cluster

To create and setup AWS EMR Cluster follow below steps:

1. From AWS Learner Lab Start the lab. It will redirect you to AWS Management Console.

2. Search for EMR in search box Or you can open the Amazon EC2 console at EMR - AWS Console.

3. Click on `Create cluster` to create a EMR cluster.

4. Use following details to create Cluster on EMR:

General Configuration:

Cluster name: winequalityprediction

Logging: Enable

Launch mode: Cluster

Software configuration:

Release: emr-5.36.0

Applications: Spark: Spark 2.4.8 on Hadoop 2.10.1 YARN and Zeppelin 0.10.0

Hardware configuration:

Instance type: m5.xlarge

Number of instances: 4 (1 master and 3 core nodes)

Auto-termination: Diable auto-termination

Security and access:

EC2 key pair: milan_aws_key_pair

Permissions: Default

I'm using .pem so it can be used with MacOS terminal.

5. Edit inbound rules for `ElasticMapReduce-master` and `ElasticMapReduce-slave`: add `SSH - Anywhere IPv4`

3.2) Transfer data into AWS EMR local file system

To transfer the data into local AWS EMR file system we're going to need AWS security Key pair.

Steps:

1. Open a terminal window. On Mac OS X, choose Applications > Utilities > Terminal.

2. Move to the location where security key pair is stored using `cd` command.

3. AWS security Key file must not be publicly viewable for SSH to work. Use this command if needed: chmod 400 milan_aws_key_pair.pem

400 protects it by making it read only and only for the owner.

4. Perform Secure copy for the CSV datasets and .jar file from your local machine:

Output of the above code:

```
milanpatel@milans-mbp PA2 % scp -i milan_aws_key_pair.pem
/Users/milanpatel/eclipse-workspace/winequalityprediction/data/TrainingDataset.csv
hadoop@ec2-54-198-81-15.compute-1.amazonaws.com:/home/hadoop

TrainingDataset.csv                   100%   66KB  88.7KB/s   00:00
```

```
milanpatel@milans-mbp PA2 % scp -i milan_aws_key_pair.pem
/Users/milanpatel/eclipse-workspace/winequalityprediction/data/ValidationDataset.csv
hadoop@ec2-54-198-81-15.compute-1.amazonaws.com:/home/hadoop

ValidationDataset.csv                 100% 8503    82.7KB/s   00:00
```

```
milanpatel@milans-mbp PA2 % scp -i milan_aws_key_pair.pem
/Users/milanpatel/eclipse-
workspace/winequalityprediction/target/winequalityprediction.jar hadoop@ec2-54-
198-81-15.compute-1.amazonaws.com:/home/hadoop

Wine-Quality-Prediction-1.0.ja        100% 6397    87.1KB/s   00:00
```

3.3) Connect to the Master Node Using **SSH**

1. Open a terminal window. On Mac OS X, choose Applications > Utilities > Terminal.

2. Move to the location where EC2 key pair is stored using `cd` command.

3. To establish a connection to the master node, type the following command in terminal to launch the cluster.

```
ssh -i milan_aws_key_pair.pem hadoop@ec2-54-198-81-15.compute-1.amazonaws.com
```

4. Type yes to dismiss the security warning.

3.4) Put Files in Hadoop Distributed file system

Run following command on Master Node CLI:

```
hadoop fs -put * .
```

3.5) Train wine quality prediction ML model:

Launching Applications with spark-submit. It will train wine quality prediction ML model in paralle using AWS EMR's four EC2 instances.

Application: winequalityprediction.jar

Java Class: PredictionModelTrainer.java

Input: The ML model will get input files (TrainingDataset.csv, ValidationDataset.csv) from Hadoop FS.

Output: It wil write the output on predictionModelPath on Hadoop FS

Output file Location: /user/hadoop/model/

Command:

spark-submit --deploy-mode cluster --class winequalityprediction.PredictionModelTrainer winequalityprediction.jar hdfs://ip-172-31-16-28.ec2.internal:8020/user/hadoop/TrainingDataset.csv hdfs://ip-172-31-16-

28.ec2.internal:8020/user/hadoop/ValidationDataset.csv hdfs://ip-172-31-16-28.ec2.internal:8020/user/hadoop/model/

Output of above commad:

22/12/08 20:19:46 INFO Client: Application report for application_1670529827505_0001 (state: RUNNING)

22/12/08 20:19:47 INFO Client: Application report for application_1670529827505_0001 (state: FINISHED)

22/12/08 20:19:47 INFO Client:

client token: N/A

diagnostics: N/A

ApplicationMaster host: ip-172-31-17-249.ec2.internal

ApplicationMaster RPC port: 42225

queue: default

start time: 1670530759026

final status: SUCCEEDED

tracking URL: http://ip-172-31-16-28.ec2.internal:20888/proxy/application_1670529827505_0001/

user: hadoop

22/12/08 20:19:47 INFO ShutdownHookManager: Shutdown hook called

22/12/08 20:19:47 INFO ShutdownHookManager: Deleting directory /mnt/tmp/spark-ba494960-e815-4bdc-bc2a-d073e6a6cccf

22/12/08 20:19:47 INFO ShutdownHookManager: Deleting directory /mnt/tmp/spark-70cc87c6-2ee5-4b55-b160-229cdadb648f

3.6) Check the performance of trained wine quality prediction ML model:

Launching Applications again with spark-submit. It wil validate the ML model and check the performance of trained wine quality prediction ML model using AWS one EC2 instance.

Application: winequalityprediction.jar

Java Class: TestPrediction.java

Input: The ML model will get input files (ValidationDataset.csv) from Hadoop FS.

Output: Output contains the wine prediction table and Absolute mean error of ValidationDataset.

Output file Location: /user/hadoop/model/

Command:

spark-submit --deploy-mode cluster --class winequalityprediction.PredictionModelTrainer winequalityprediction.jar hdfs://ip-172-31-16-28.ec2.internal:8020/user/hadoop/ValidationDataset.csv hdfs://ip-172-31-16-28.ec2.internal:8020/user/hadoop/model/

Output of above commad:

[hadoop@ip-172-31-16-28 ~]$ spark-submit --deploy-mode cluster --class winequalityprediction.PredictionModelTrainer winequalityprediction.jar hdfs://ip-172-31-16-28.ec2.internal:8020/user/hadoop/ValidationDataset.csv hdfs://ip-172-31-16-28.ec2.internal:8020/user/hadoop/model/

22/12/08 20:27:16 INFO GPLNativeCodeLoader: Loaded native gpl library

22/12/08 20:27:16 INFO LzoCodec: Successfully loaded & initialized native-lzo library [hadoop-lzo rev 049362b7cf53ff5f739d6b1532457f2c6cd495e8]

+-------------------+-------+------------------+

| features|quality| prediction|

+-------------------+-------+------------------+

||[4.7,0.6,0.17,2.3...| 6| 5.775033061578215|

```
|[8.3,0.42,0.38,2....|     6| 5.826299201762993|
|[10.0,0.48,0.24,2...|     6|5.4825807223501215|
|[10.9,0.53,0.49,4...|     6|6.0322388644384315|
|[5.0,1.02,0.04,1....|     4| 4.677494723964648|
|[9.3,0.715,0.24,2...|     5|  5.40913518074032|
|[7.8,0.46,0.26,1....|     6|5.3186771136967295|
|[5.6,0.615,0.0,1....|     5| 5.042239679239383|
|[9.3,0.39,0.44,2....|     5| 6.029804263392517|
|[7.6,0.9,0.06,2.5...|     5| 4.994329279478681|
|[10.9,0.21,0.49,2...|     6| 6.460044027268264|
|[8.1,0.545,0.18,1...|     6| 5.158607763014938|
|[7.4,0.66,0.0,1.8...|     5| 5.015281677184131|
|[7.0,0.62,0.08,1....|     5| 4.907881419042209|
|[8.5,0.28,0.56,1....|     7|  5.82950519942851|
|[11.6,0.32,0.55,2...|     7| 6.238443699117255|
|[10.5,0.51,0.64,2...|     7| 6.117838445890076|
|[13.7,0.415,0.68,...|     6|  5.93062783232325|
|[11.9,0.38,0.49,2...|     5| 5.762192588038326|
|[8.0,0.33,0.53,2....|     6| 5.542901695484648|
+--------------------+-------+------------------+
only showing top 20 rows


---------------------------------------------------

------ Absolute mean error:0.5294072418851784 ------

---------------------------------------------------
```

3.7) Run the wine quality prediction ML model with Docker:

Following following steps to run the wine quality prediction ML model with Docke for Mac

1. Get Docker for Mac

2. OR - Install from the command line using following command:

    sudo hdiutil attach Docker.dmg

    sudo /Volumes/Docker/Docker.app/Contents/MacOS/install

    sudo hdiutil detach /Volumes/Docker

3. Run Following commnad to pull the new docker image of wine quality prediction model:

    docker push milankumarpatel/winequalityprediction:tagname

4. Run Follwoing command to run using Docker:

    docker run --network winequalityprediction /Users/milanpatel/eclipse-workspace/winequalityprediction/data/ValidationDataset.csv:/usr/src/app/data/ValidationDataset.csv --link spark-master:spark-master milankumarpatel/winequalityprediction --name winequalityprediction-docker

4. References

- Canvas Link for Programming Assignment 2
- AWS Learner Lab
- Apache - Spark
- Spark – Mllib
- Docker
- spark-maven-template