CS643_Wine_Prediction
Wine Quality Prediction in ML using pyspark on AWS

GitHub Link: https://github.com/harshl08/CS643_Wine_Prediction
Docker Hub: https://hub.docker.com/repository/docker/ladharsh/cloud_programming2

Harshvardhan Lad
hsl6@njit.edu

Goal: The purpose of this individual assignment is to learn how to develop parallel machine learning (ML) applications in Amazon AWS cloud platform. Specifically, you will learn: (1) how to use Apache Spark to train an ML model in parallel on multiple EC2 instances; (2) how to use Spark's MLlib to develop and use an ML model in the cloud; (3) How to use Docker to create a container for your ML model to simplify model deployment.

1. Parallel Training Implementation:

Step1: Create an ec2 instance to run the flintrock server. Setup flintrock per the instructions Given in the document in class announcements. Make sure the config.yaml file installs both Spark and Hadoop.

Step2: Now,

1. Create a cluster with five m4-large instances where four of them are workers and one of them is the master.
   flintrock launch <cluster_name>

2. Log into spark master using:
   flintrock login < cluster_name>.

3. **Set inbound Rules After Creating the Cluster add the SSH inbound rule to the master node for port 22**

4. Add to the root of HDFS a folder called data and place in there the files to be used for training, validation, and/or testing. Should look like this:

```
ls: `data': No such file or directory
[ec2-user@ip-172-31-7-218 ~]$ hdfs dfs -ls /data
WARNING: log4j.properties is not found. HADOOP_CONF_DIR may be incomplete.
Found 2 items
-rw-r--r--   3 ec2-user supergroup      68804 2021-12-02 21:33 /data/TrainingDataset.csv
-rw-r--r--   3 ec2-user supergroup       8760 2021-12-02 21:53 /data/ValidationDataset.csv
[ec2-user@ip-172-31-7-218 ~]$
```

5. Submit the training job with the following command:
   spark-submit --master spark://ip-172-31-7-218.ec2.internal:7077 model_train.py hdfs:///data/TrainingDataset.csv hdfs:///model

```
21/12/03 23:43:42 INFO StandaloneAppClient$ClientEndpoint: Executor updated: app-20211203234342-0002/0 is now RUNNING
21/12/03 23:43:42 INFO StandaloneAppClient$ClientEndpoint: Executor updated: app-20211203234342-0002/3 is now RUNNING
21/12/03 23:43:42 INFO StandaloneSchedulerBackend: SchedulerBackend is ready for scheduling beginning after reached minRegisteredResourcesRatio: 0.0
21/12/03 23:43:42 INFO SharedState: Setting hive.metastore.warehouse.dir ('null') to the value of spark.sql.warehouse.dir ('file:/home/ec2-user/spark-warehouse').
21/12/03 23:43:42 INFO SharedState: Warehouse path is 'file:/home/ec2-user/spark-warehouse'.
Reading data from hdfs:///data/TrainingDataset.csv...
Saving file to hdfs:///model...
Model is saved ... now terminating.
[ec2-user@ip-172-31-7-218 ~]$
```

6. I have used DecisionTree algorithm to train and predict the accuracy

7. We then want to grab our model from HDFS, so we use the get command:
   hdfs dfs -get /model

8. This is our saved model that we can then use to create our prediction application.

2. Single Machine Prediction Application:

Step1:
We just need to execute Spark in local master mode to only use the master node for prediction. This can be done with the following command:
spark-submit --master local[*] quality_predict.py file:///home/ec2-user/ValidationDataset.csv file:///home/ec2-user/model

Output would look like this after all the log messages:

```
9, None)
21/12/03 23:50:38 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, ip-172-31-7-218.ec2.internal, 42509, None)
21/12/03 23:50:38 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, ip-172-31-7-218.ec2.internal, 42509, None)
21/12/03 23:50:39 INFO SharedState: Setting hive.metastore.warehouse.dir ('null') to the value of spark.sql.warehouse.dir ('file:/home/ec2-user/spark-warehouse').
21/12/03 23:50:39 INFO SharedState: Warehouse path is 'file:/home/ec2-user/spark-warehouse'.
Loading file:///home/ec2-user/model...
Evaluating the model...
Accuracy of the model is = 0.48125
F1 Score = 0.43487
Model prediction is done ... now terminating.
[ec2-user@ip-172-31-7-218 ~]$
```

Creating Docker Image for Prediction Application
1. Install most recent docker engine package: sudo amazon-linux-extras install docker or sudo yum install docker
2. Start docker : **sudo service docker start**
3. Adding ec2-user to docker group: **sudo usermod -a -G docker ec2-user**
4. Exit the flintrock cluster and login again.
5. verify ec2-user: **docker info**
6. After setting up login into your docker using : docker login. Here you have to provide you credentials for dockerhub.
7. Build the docker image using : **docker build -t ladharsh/cloud_programming2 .**

8. Run the image using: **docker run ladharsh/cloud_programming2 driver quality_predict.py ValidationDataset.csv model**
9. Pushing the image to Docker hub repository: **docker push ladharsh/cloud_programming2**

3. Docker container for prediction application

1. Launch your ec2-instance and then step-up docker using the above steps.
2. Copy the ValidationDataset.csv into location where you will pull the repository
3. Pull the image from repositroy: **docker pull ladharsh/cloud_programming2**
4. Run the image using : **docker run ladharsh/cloud_programming2 driver quality_predict.py ValidationDataset.csv model**