Angela Murphy

04/30/2023

**CS 643 Programming Assignment 2**

**Task 1**: Parallel Training on four ec2 Instances

**First:**

* Log into Aws Console
  + Find EMR Service & Create Cluster
  + Ensure that you are setting the proper security settings along with vendor amazon, Release emr 5.3.10 and select spark application (whichever is the updated version). Hardware configuration should be set to four instances and in this case this would be 1 master and 3 slaves. Creating a name for these instances will only help you keep track of things.
  + Select the proper ec2 key pair.
  + Create cluster.

A screenshot of a computer

Description automatically generated

**Upload files to the EMR Cluster Master Node**

* Ensure that you download the ppk key from aws.
* Upload the downloaded ppk key and start a ssh session for the master node.
* Create a directory titled data under /home/hadoop and upload both csv files provided to the Data folder. Upload your jar file in /home/hadoop.
* In order for our slave nodes to access these files, move the files to HDFS. The following command will be applicable here:

$ hadoop fs -put Data/ValidationDataset.csv /user/hadoop/ValidationDataset.csv

$ hadoop fs -put Data/TrainingDataset.csv /user/hadoop/TrainingDataset.csv

**The following command will verify these files are copied to HDFS:**

$ hdfs dfs -ls /user/hadoop

**Launching Apache-Spark Application on EMR Cluster**

Execute following command:

Sudo spark-submit –class CS643\_Programming\_Assignment2.Wine\_Quality\_Training app.jar

**Task 2**

Here we will execute the prediction code on a single ec2 instance.

* Log into AWS Console.
* Click EC2 and then launch instance as well as select AMI.
* Here we would select the keypair and launch it.
* After this installing java and scala within the ec2 is necessary.
* Update path environment variables as well for scala.
* Installation of spark is necessary as well.
* Once these steps are completed, we can run the wine-prediction application through the ec2.

$ spark-submit –class cs643.predictionapp.Wine\_Quality\_Prediction prediction-app.jar

**Task 3**

Here we will be using docker. We need to build a docker container for the prediction application so that the prediction model can be quickly deployed across many different environments.

1. Create your Dockerfile and requirements.txt
2. Create a docker repository on the docker profile. Mine is named -cs643\_program\_2
3. Run the following commands to create and publish the docker image to the docker hub.

* Docker build -t cs643program2
* Tag the docker image as such: docker tag cs643program2 am3228/cs643\_program\_2
* Push the docker image to docker hub: docker push am3228/cs643\_program\_2