Cloud Computing - Programming Assignment 2

Chaitanya Karnati **UCID: ck338**

Wine Quality Prediction AWS Spark Application:

pa2assignment:

This project entails creating a Python application that leverages the PySpark interface. The application is deployed on an Amazon Web Services (AWS) Elastic MapReduce (EMR) cluster. The main goal is to parallelize the training of a machine learning model on EC2 instances to predict wine quality using publicly accessible data. Following the training phase, the model is then used to predict the quality of wine. Docker is employed to produce a container image for the trained machine learning model, simplifying the deployment process.

GitHub Link: https://github.com/ChaitanyaKarnati/Cloud-Computing-PA2

Docker Link: https://hub.docker.com/repository/docker/chaitanyakarnati/winequlpred/general

Execution Steps for the AWS Spark Application for Wine Quality Prediction:

1. Generate an EMR Cluster Key Pair:
- Navigate to EC2/Network/Key-pairs in the AWS console.
- Use the .pem format and download the key pair.
- Example key pair: pa2assignment.pem

2. Set Up an S3 Bucket:
- Create an S3 bucket in AWS, e.g., pa2assignwineprediction

3. EMR Cluster Creation:
- Access the EMR console.
- Create a new EMR cluster.

4. Spark Cluster Configuration:
- Configure the Spark cluster using the EMR console.
- Specify the cluster details, including:
- Name: chaitanyapa2
- Amazon EMR release: emr-5.33.0
- Application bundle: Hadoop 2.10.1, Spark 2.4.7, Zippeline 0.9.0, and Yarn.

5. Instance Configuration:
- Create four instances for the Spark cluster.

These steps collectively establish the necessary infrastructure and configurations for running the AWS Spark application dedicated to predicting wine quality.

6. Currently, we are concurrently training an ML model on a Spark cluster using EC2 instances. Subsequently, the cluster is prepared to execute tasks related to running the ML model. To establish a connection with the Master instance through the Terminal:

- **ssh -i "pa2assignment.pem" ec2-user@ec2-54-242-27-160.compute-1.amazonaws.com**

7. Upon successful login, After accessing the Master instance, switch to the root user by executing:

- **sudo su**

8. Submit the task by the command:

- **spark-submit s3://pa2assignwineprediction/wine_prediction.py**

9. Subsequently, check the trace status for the aforementioned tasks. If the status indicates success, it implies the successful creation of the "test.model" in the S3 bucket at the location s3://**pa2assignwineprediction**.

Docker steps:

1. Presently, we are executing the ML model using Docker:

- Begin by creating a Docker account and completing the signup process.
- Following a successful login, proceed to download and set up Docker on your local system.
- Install Docker on your system.
- In the PowerShell, log in to Docker using the following command:

  docker login
  docker login -u chaitanyakarnati
  pwd

- After login you need to build the image: **docker build -t winequlpre .**

2. The push and pull into the docker hub repository:

- PUSH:
  docker tag winequlpre chaitanyakarnati/winequlpre
  docker push chaitanyakarnati/winequlpre
- PULL:
  docker pull chaitanyakarnati/winequlpre

3. Place your test data file in a specified folder named "dir." Mount this directory to the Docker container and run the container using the following command:

- **docker run -v /Users/chaitanyakarnati/Documents/Wineprediction/data/csv winequlpred testdata.csv**

Running the machine learning model without utilizing Docker:

1. Clone the Repository:
   Begin by cloning this repository onto your local machine.

2. Set Up Local Spark Environment:
   Ensure that you have a local Spark environment ready for running this application. If you haven't set up Spark yet, you can follow the instructions provided in the [official Spark documentation](https://spark.apache.org/docs/latest).

3. Navigate to the Python File Folder:
   Access the 'python file' directory within the cloned repository.

4. Prepare Test Data:
   Store your test data in the '/Users/chaitanyakarnati/Documents/Wineprediction/data/csv' folder.

Conclusion:

As depicted in the image below, an accuracy of approximately 98% was achieved while predicting wine quality.