CS643 Programming Assignment 2

Rudolph Paulin

12/11/2024

Github Link: *https://github.com/Bonli9/CS643-Wine-Prediction.git*

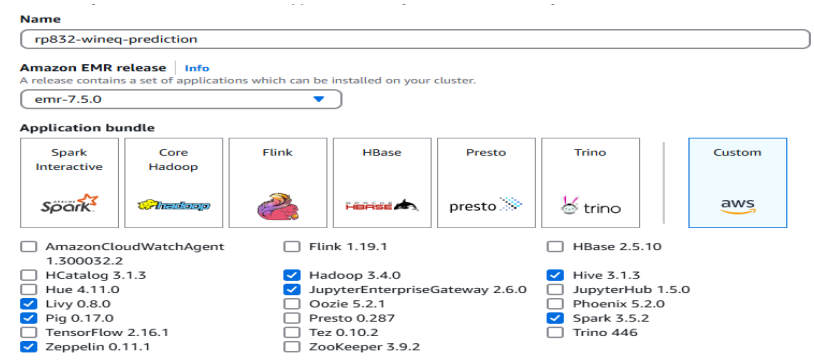
DockerHub Link: *https://hub.docker.com/repository/docker/rpaulin/wine-quality-prediction*

Training Setup

Go to Amazon Console Home, Select S3 (Scalable Storage in the cloud) and click on "Create Bucket." Name the bucket "cs643rudolphpaulin," and uncheck the "Block all public access" option, keep the default settings for all other options and create the bucket. After the bucket is created, click on it and hit ***Upload*** to select “TrainingDataset.csv” and “ValidationDataset.csv” files.

AWS EMR

Go to EMR on EC2 → Create Cluster → Give the cluster a name (“rp832-wineq-prediction”) → Choose emr-7.5.0(Amazon EMR release). Then select the following applications: Zeppelin 0.11.1 ; Pig 0.17.0 ; and leave the other applications selected by default, as shown in the image below:



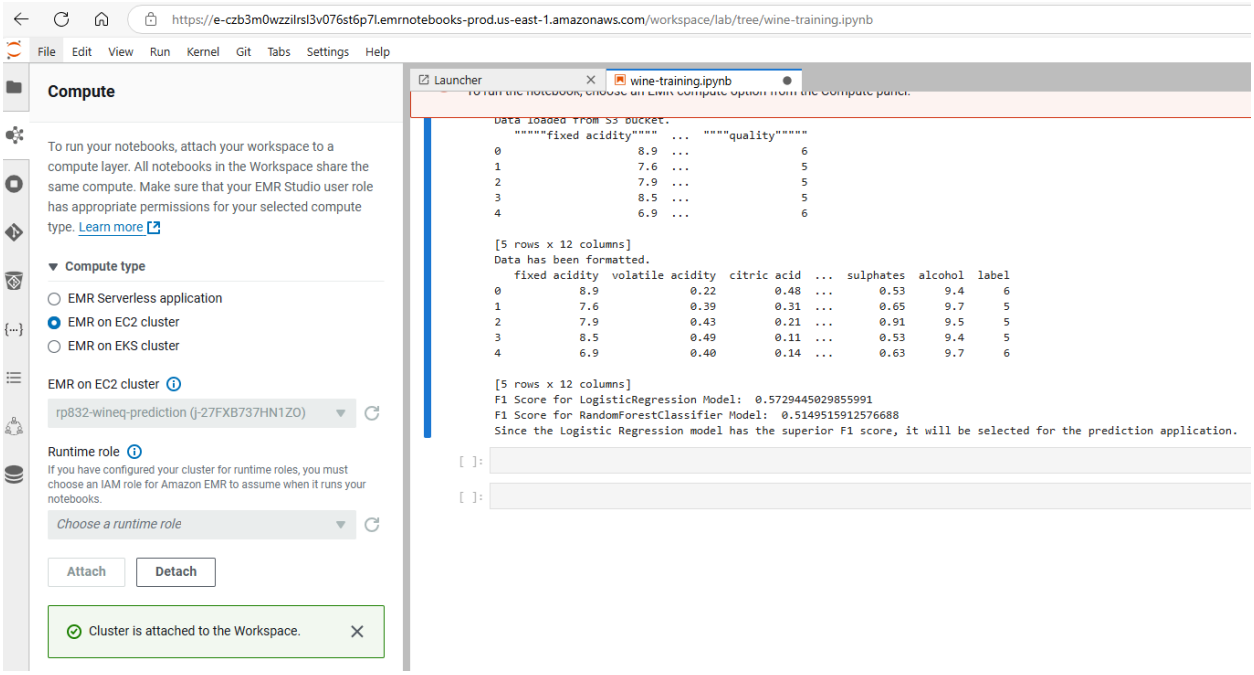
Choose a primary instance type m5.xlarge and 3 core groups. Then scroll down to select by defaults:

EMR_DefaultRole→EMR_EC2_DefaultRole→EMR_AutoScaling_DefaultRole

Then hit “create cluster” button

While waiting for the cluster to initialize, go to the left panel and click on "Workspaces (Notebooks)" then select "Create Notebook." Name the notebook "wine-training." Under "Cluster," click "Choose" and pick the cluster that was set up earlier ("CS643rudolphpaulin"). Leave the other settings as defaults and click "Create Workspace" and name it ***wine-training***. Wait for the status to update from "Starting" to "Ready." Once ready, click on the workspace and Jupyter application will open. In Jupyter, select the "wine-training.ipynb" notebook, then from the toolbar, click on Kernel > Change Kernel > PySpark.

Download the code from "training.py" (see Github repository) and paste it into the first cell of the notebook. After that, click "Run." Once executed, the results for both the LogisticRegression and RandomForestClassifier models will appear, showing the training outputs from all four nodes.



Prediction Configuration

In the AWS Console, go to Services > EC2 > Launch Instances. Choose “Amazon Linux 2 AMI..." Select the t2.large instance. Leave all other settings as default.

Generate a new key pair and name it "pa2.pem." Click "Download key pair," then select "Launch Instances." After that, click "View Instances." Initially, the EC2 instance status will likely display as "Pending." While it transitions to "Running," open a terminal and move the downloaded .pem file to your home directory. Next, execute the following command to adjust the file permissions for the .pem file:

- First, set the correct permissions for your .pem file by running the following command:

$ chmod 400 pa2.pem

- Once the EC2 instance is running, connect to it using the following SSH command:
  $ ssh -i ~/pa2.pem ec2-user@<YOUR_INSTANCE_PUBLIC_DNS>

- Next, open a terminal on your local machine to copy the data files (TrainingDataset.csv and TestDataset.csv):

$ scp -i ~/pa2.pem TrainingDataset.csv ec2user@<YOUR_INSTANCE_PUBLIC_DNS>:~/

$ scp -i ~/pa2.pem TestDataset.csv ec2-user@<YOUR_INSTANCE_PUBLIC_DNS>:~/

- Afterward, SSH into your EC2 instance again. You should see both data files in the home directory. Move them to the appropriate folder on your EC2 instance by running:

$ sudo mkdir /app

$ sudo cp TrainingDataset.csv TestDataset.csv /wineapp/

Without Docker

After connecting to your EC2 instance via SSH, use the following commands to install and configure the Java Development Kit (JDK):

$ sudo yum install java-17-amazon-corretto

$ export JAVA_HOME=/usr/lib/jvm/java-17-amazon-corretto

To prepare your environment for Spark and Python, perform these steps:

1. Fetch Apache Spark 3.5.3 and unpack it into a dedicated directory:

$ wget https://archive.apache.org/dist/spark/spark-3.5.3/spark-3.5.3-bin-hadoop3.tgz -P ~/server

$ cd ~/server

$ sudo tar -xvzf spark-3.5.3-bin-hadoop3.tgz

2. Download the Anaconda installer and execute the installation:

$ curl -O https://repo.anaconda.com/archive/Anaconda3-2023.11-Linux-x86_64.sh

$ mv Anaconda3-2023.11-Linux-x86_64.sh /tmp

$ cd /tmp

$ bash Anaconda3-2023.11-Linux-x86_64.sh

Then set up the environment variables.

Reconnect to your EC2 instance and execute these commands to complete the setup:

1. Install additional Python dependencies:

$ pip install quinn

2. Set a password for Jupyter Notebook:

$ jupyter notebook password

3. Start the Jupyter Notebook server:

$ jupyter notebook

Access Jupyter Notebook on Your Local Machine

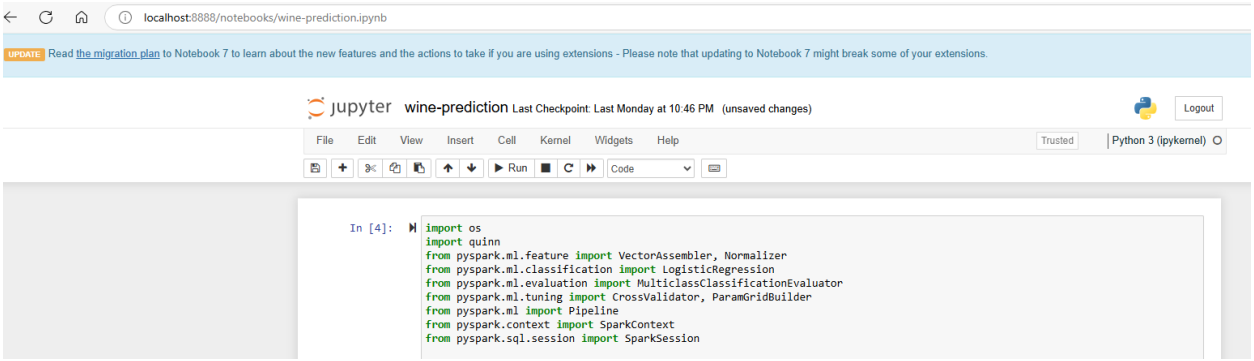
To access the notebook from your browser:

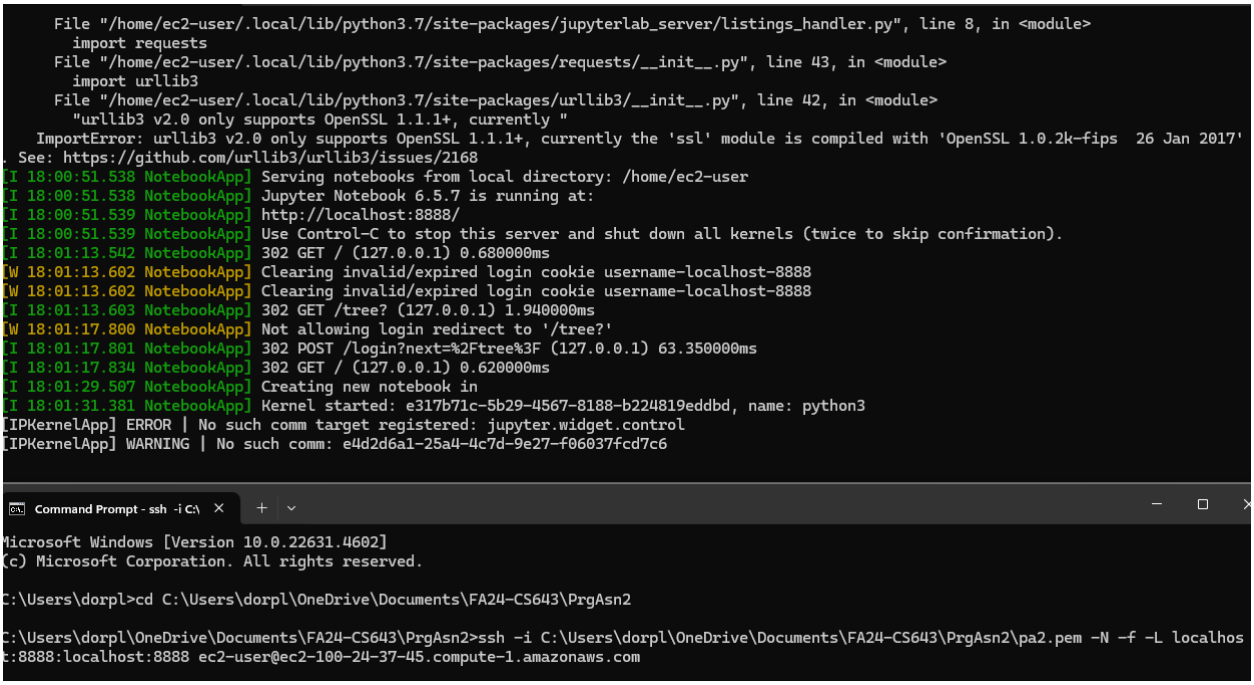
1. Open a terminal on your local machine and create an SSH tunnel to forward traffic from your EC2 instance to your local machine:

$ ssh -i ~/pa2.pem -N -f -L localhost:8888:localhost:8888 ec2-user@<YOUR_INSTANCE_PUBLIC_DNS>

2. Open your web browser and go to http://localhost:8888.
3. Log in using the password you set during the configuration process.

See the image below





Prediction with Docker on EC2

Follow these steps to set up Docker on your EC2 instance and run your prediction Docker image:

While logged into your EC2 instance via SSH, run the following commands to install *Docker: $ sudo yum install docker -y*

Start the Docker service: $ sudo service docker start