CS643 - AWS Spark Wine Quality Prediction Application

Name - Abhiraj Gupta
UCID - ag2672
Section – 861

- I developed a machine learning model to predict wine quality using publicly available data.
- I trained the model in parallel on EC2 instances to improve efficiency.
- I leveraged Docker to create a container image of the trained model, simplifying the deployment process.

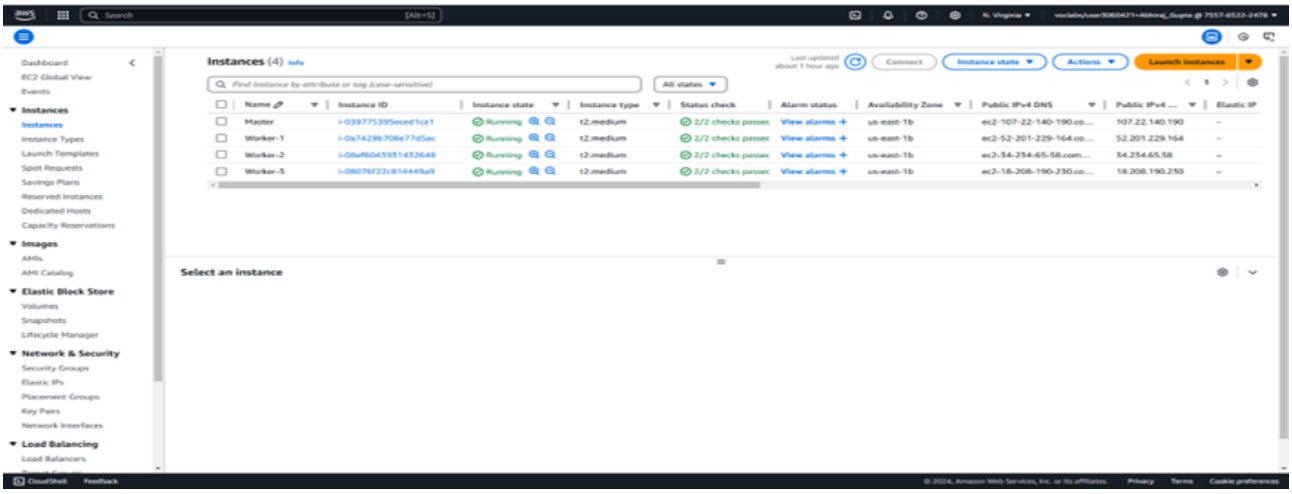
Link to github code:
https://github.com/Abhiraj16/CS643_ProgrammingAssignment2

Link to Docker:
https://hub.docker.com/r/abhiraj1625/wine-eval

Docker Pull Command:
docker pull abhiraj1625/wine-eval

Instructions:

1) **Accessing the four instances:**
   Connect to each of the four instances via SSH using the public IP address and the .pem key file.



2) **Generate SSH Keys on All Instances**

- Create SSH keys on each instance and record the public keys for sharing.

3) **Enable Passwordless SSH**

- Add the public keys from all instances to the authorized_keys file on each instance to allow passwordless SSH communication.

4) **Update /etc/hosts on All Instances**

- Add the IP addresses and hostnames of all instances to the /etc/hosts file to facilitate easier hostname resolution.

5) **Install Java, Maven, and Spark**

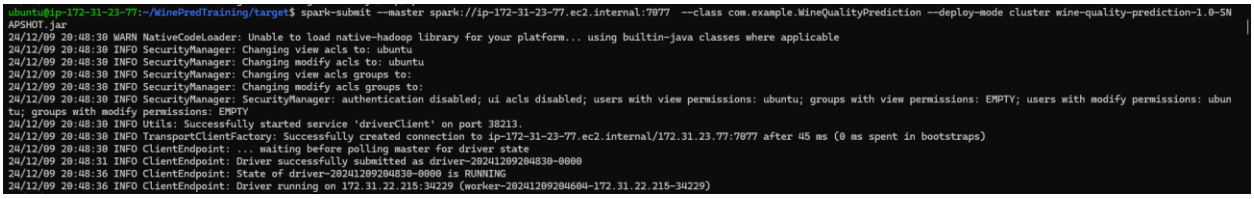
- Install Java (OpenJDK 8), Maven, and Spark 3.4.1 on each instance.
- Configure the necessary environment variables to ensure Spark is globally accessible.

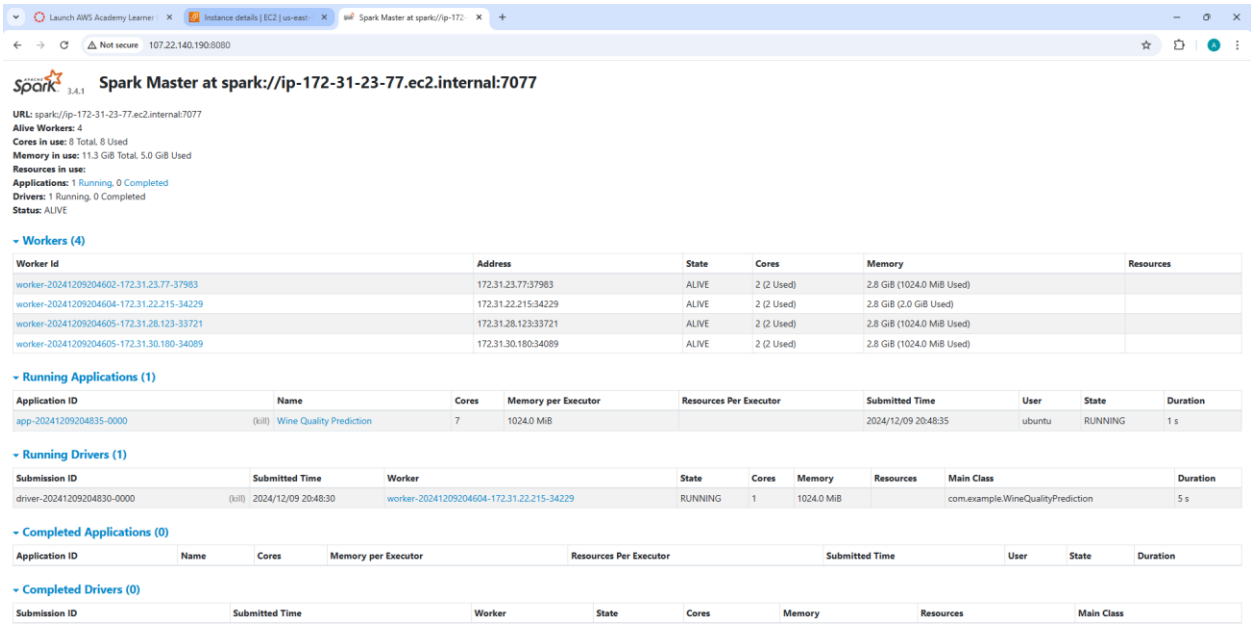
6) **Configure Spark Workers**

- Update the workers file in the Spark configuration directory by adding the hostnames or IP addresses of all worker instances.

7) Set up directories for Training and Evaluation

8) To execute the training code in parallel on all instances, use the spark-submit command. This command enables distributed processing by leveraging the Spark cluster setup, allowing the training tasks to run concurrently across all instances.





Spark Master at spark://ip-172-31-23-77.ec2.internal:7077

URL: spark://ip-172-31-23-77.ec2.internal:7077
Alive Workers: 4
Cores in use: 8 Total, 8 Used
Memory in use: 11.3 GiB Total, 5.0 GiB Used
Resources in use:
Applications: 1 Running, 0 Completed
Drivers: 1 Running, 0 Completed
Status: ALIVE

Workers (4)

Worker Id	Address	State	Cores	Memory	Resources
worker-20241209204602-172.31.23.77-37983	172.31.23.77:37983	ALIVE	2 (2 Used)	2.8 GiB (1024.0 MiB Used)	
worker-20241209204604-172.31.22.215-34229	172.31.22.215:34229	ALIVE	2 (2 Used)	2.8 GiB (2.0 GiB Used)	
worker-20241209204605-172.31.28.123-33721	172.31.28.123:33721	ALIVE	2 (2 Used)	2.8 GiB (1024.0 MiB Used)	
worker-20241209204605-172.31.30.180-34089	172.31.30.180:34089	ALIVE	2 (2 Used)	2.8 GiB (1024.0 MiB Used)	

Running Applications (1)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20241209204835-0000	(kill) Wine Quality Prediction	7	1024.0 MiB		2024/12/09 20:48:35	ubuntu	RUNNING	1 s

Running Drivers (1)

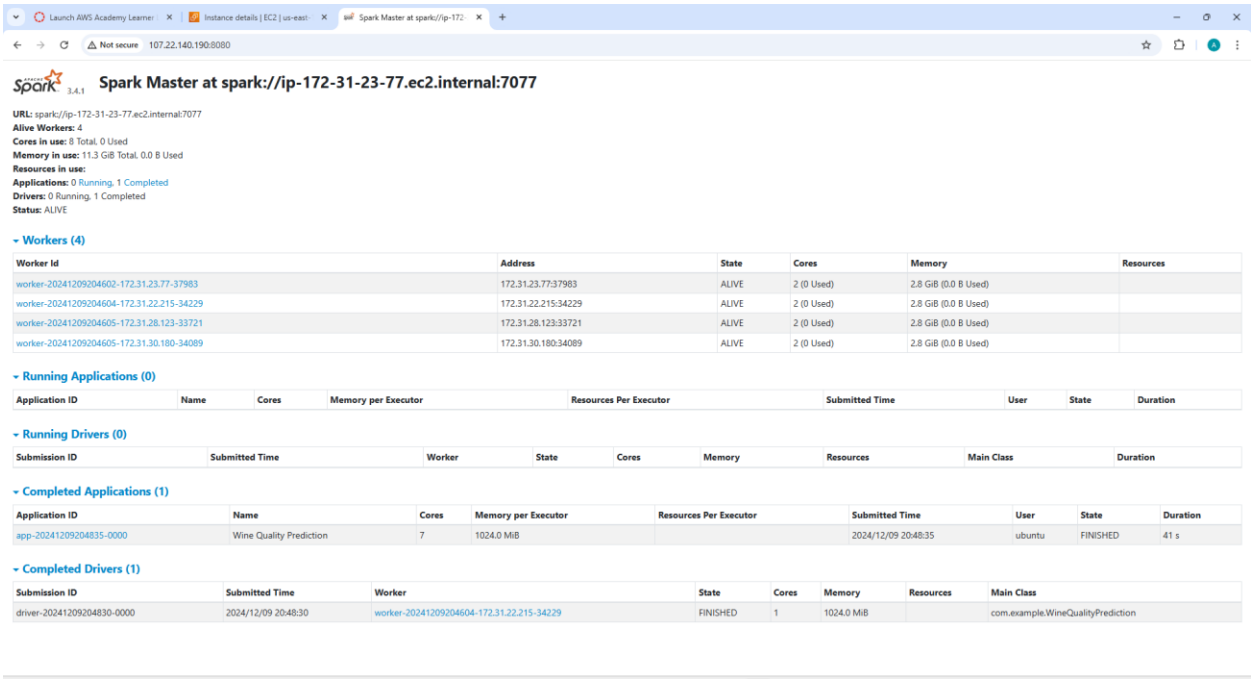
Submission ID	Submitted Time	Worker	State	Cores	Memory	Resources	Main Class	Duration
driver-20241209204830-0000 (kill)	2024/12/09 20:48:30	worker-20241209204604-172.31.22.215-34229	RUNNING	1	1024.0 MiB		com.example.WineQualityPrediction	5 s

Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

Completed Drivers (0)

Submission ID	Submitted Time	Worker	State	Cores	Memory	Resources	Main Class



Spark Master at spark://ip-172-31-23-77.ec2.internal:7077

URL: spark://ip-172-31-23-77.ec2.internal:7077
Alive Workers: 4
Cores in use: 8 Total, 0 Used
Memory in use: 11.3 GiB Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 1 Completed
Drivers: 0 Running, 1 Completed
Status: ALIVE

Workers (4)

Worker Id	Address	State	Cores	Memory	Resources
worker-20241209204602-172.31.23.77-37983	172.31.23.77:37983	ALIVE	2 (0 Used)	2.8 GiB (0.0 B Used)	
worker-20241209204604-172.31.22.215-34229	172.31.22.215:34229	ALIVE	2 (0 Used)	2.8 GiB (0.0 B Used)	
worker-20241209204605-172.31.28.123-33721	172.31.28.123:33721	ALIVE	2 (0 Used)	2.8 GiB (0.0 B Used)	
worker-20241209204605-172.31.30.180-34089	172.31.30.180:34089	ALIVE	2 (0 Used)	2.8 GiB (0.0 B Used)	

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

Running Drivers (0)

Submission ID	Submitted Time	Worker	State	Cores	Memory	Resources	Main Class	Duration

Completed Applications (1)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20241209204835-0000	Wine Quality Prediction	7	1024.0 MiB		2024/12/09 20:48:35	ubuntu	FINISHED	41 s

Completed Drivers (1)

Submission ID	Submitted Time	Worker	State	Cores	Memory	Resources	Main Class
driver-20241209204830-0000	2024/12/09 20:48:30	worker-20241209204604-172.31.22.215-34229	FINISHED	1	1024.0 MiB		com.example.WineQualityPrediction

Step 9: Build and Push Docker Image

Use the provided Dockerfile to build a Docker image that includes the trained model, validation dataset, and evaluation code. Then, push the Docker image to Docker Hub for

easy access across instances.
Commands:

sudo docker build -t abhiraj1625/wine-quality-eval:latest .

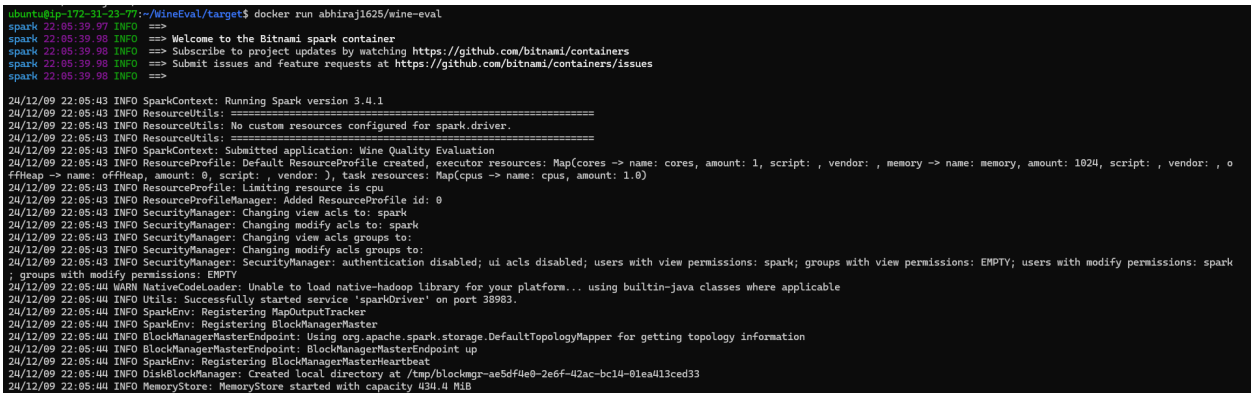
sudo docker push abhiraj1625/wine-quality-eval:latest

Step 10: Pull Docker Image on Target Instances

On the desired instances, pull the Docker image from Docker Hub to deploy the evaluation setup.
Command:

sudo docker pull abhiraj1625/wine-quality-eval:latest

Step 11) Start the Docker container on the desired instances to execute the evaluation setup.



Finally, I got an F1 Score of 0.8730357142857142, and I have used model as Random Forest.