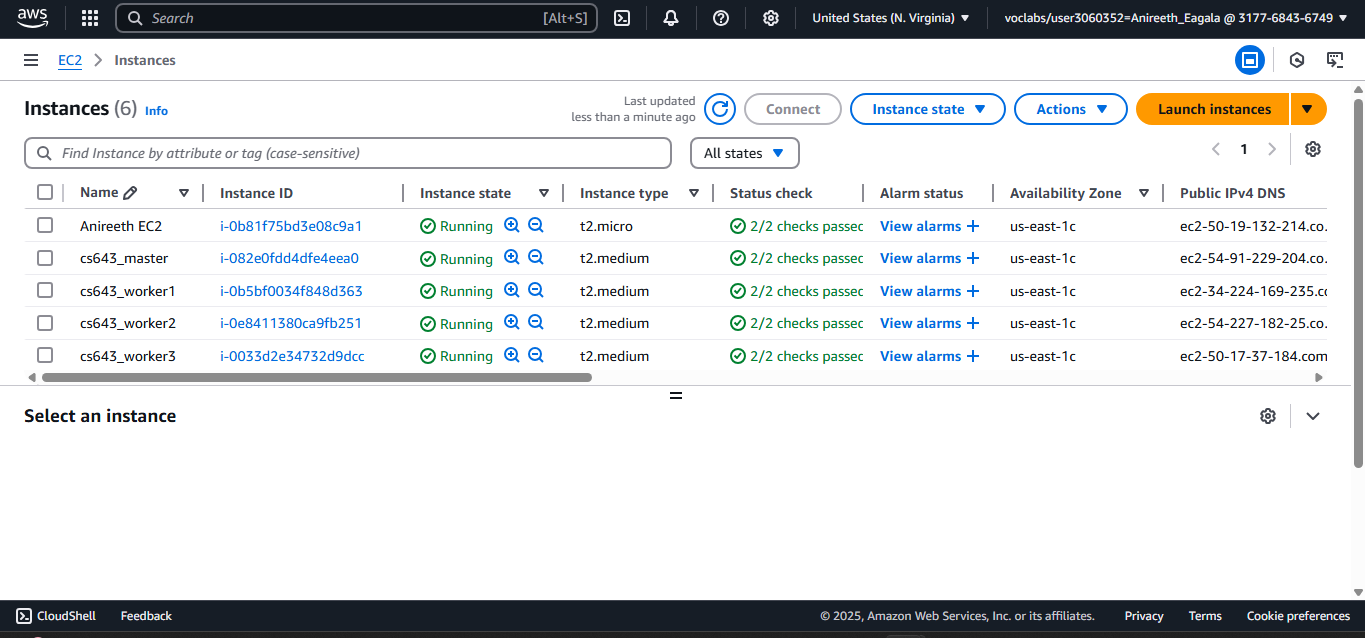
**CS643: Wine Quality Prediction – Final Report**

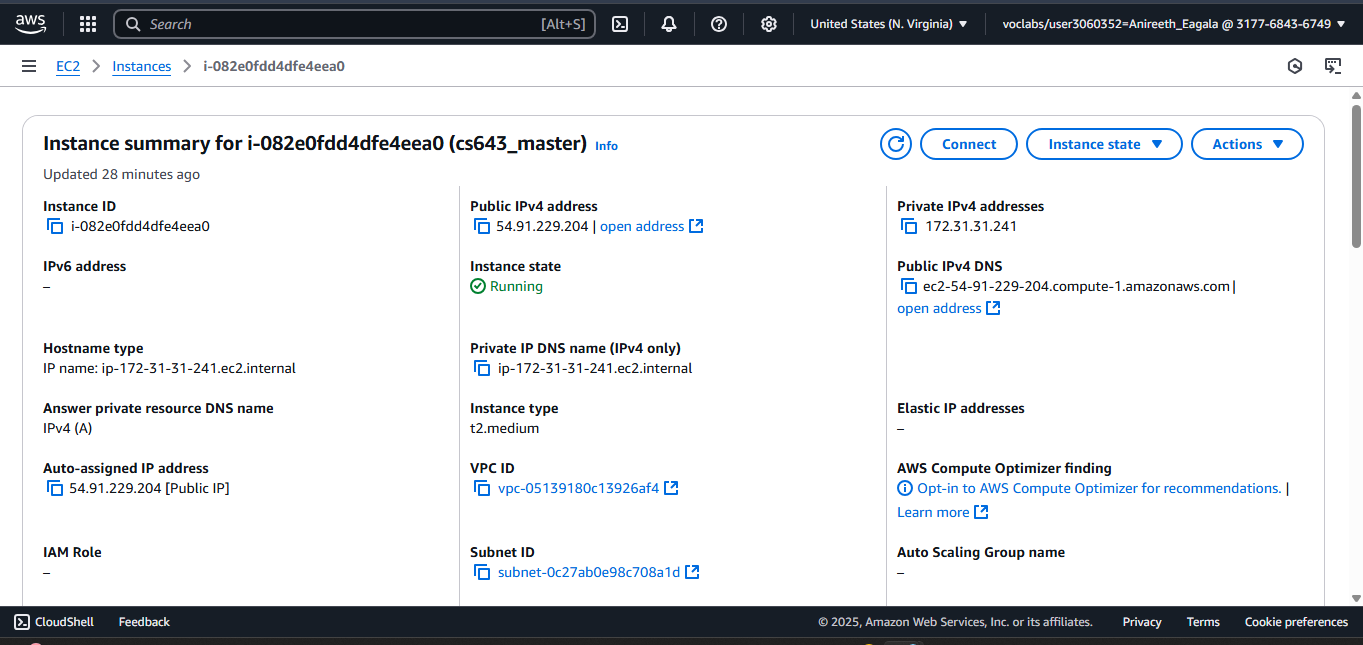
**1. AWS EC2 Instance Setup and Connection**

We used an **Amazon EC2 instance** to build, test, and run our Docker containerized Spark MLlib application.

**Steps:**

1. **Instance Creation**:
   * Go to AWS EC2 Dashboard.
   * Launch an EC2 instance using the following:
     + **AMI**: Ubuntu Server 22.04 LTS
     + **Instance Type**: t2.medium (or higher, depending on resources)
     + **Key Pair**: cs643\_keypair.pem
     + **Security Group**: Allowed SSH (port 22) from our IP address.
     + **Master 1, workers 3**



****

1. **Connect to EC2**:
   * Run the following command from your local terminal:
   * chmod 400 cs643\_keypair.pem
   * ssh -i "cs643\_keypair.pem" ubuntu@ec2-54-91-229-204.compute-1.amazonaws.com
2. **Setup on EC2**:
   * Install Docker:
   * sudo apt update
   * sudo apt install docker.io -y
   * sudo usermod -aG docker ubuntu
   * Logout and log back in to apply Docker permissions.
3. **Copy Files to EC2**: From your local machine:

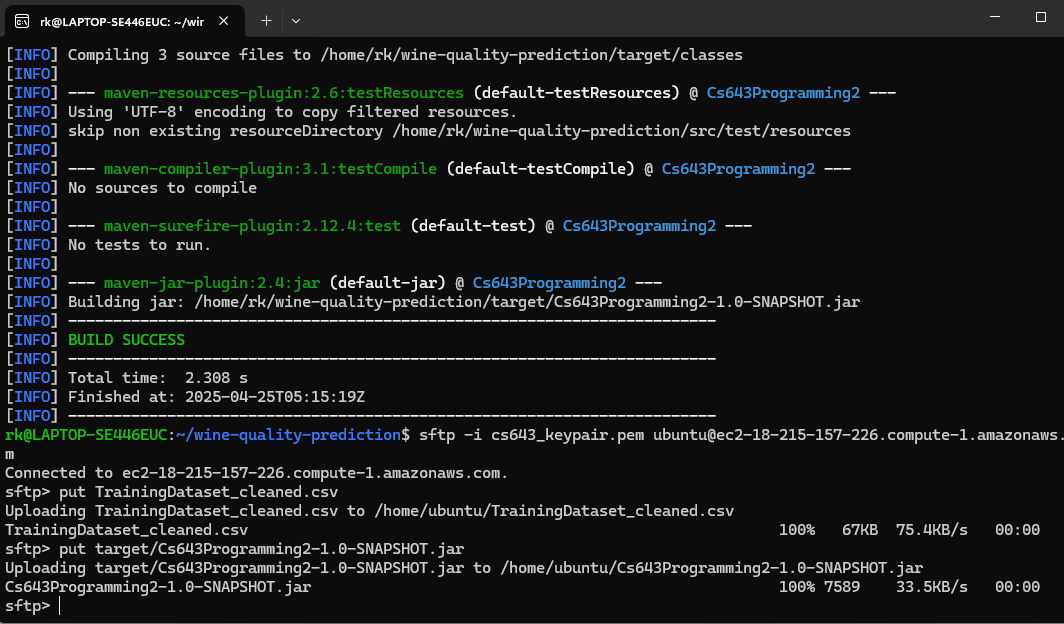
sftp -i cs643\_keypair.pem Cs643Programming2-1.0-SNAPSHOT.jar [ubuntu@ec2-54-91-229-204.compute-1.amazonaws.com](mailto:ubuntu@ec2-54-91-229-204.compute-1.amazonaws.com)

put TrainingDataset\_cleaned.csv

put TestDataset.csv

put ValidationDataset.csv

put target/Cs643Programming2-1.0-SNAPSHOT.jar



1. **Build and Run Docker**: On the EC2 instance:

sudo docker build -t wine-quality-prediction .

sudo docker run -v /home/ubuntu:/app wine-quality-prediction

**1. Dataset Description**

**Source**: UCI Machine Learning Repository  
**Files Used**:

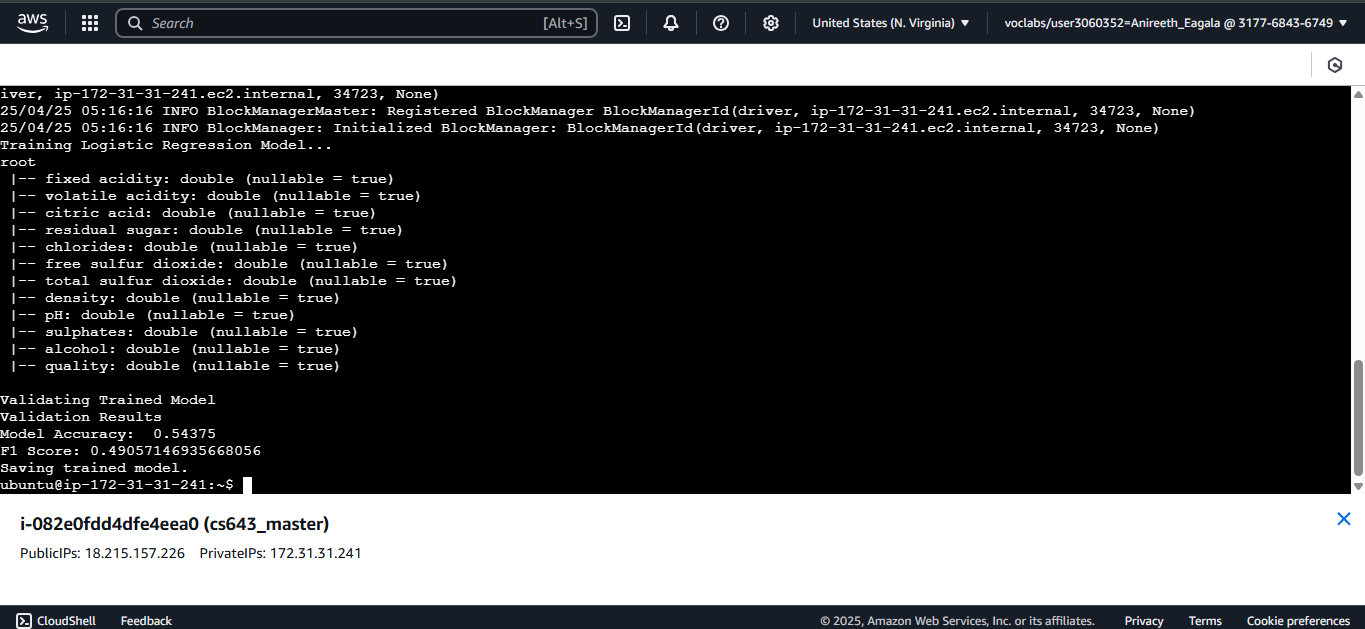
* TrainingDataset.csv
* ValidationDataset.csv
* TestDataset.csv

**2. Data Preprocessing**

* Remove invalid entries using Validation Dataset
* Output: TrainingDataset\_cleaned.csv

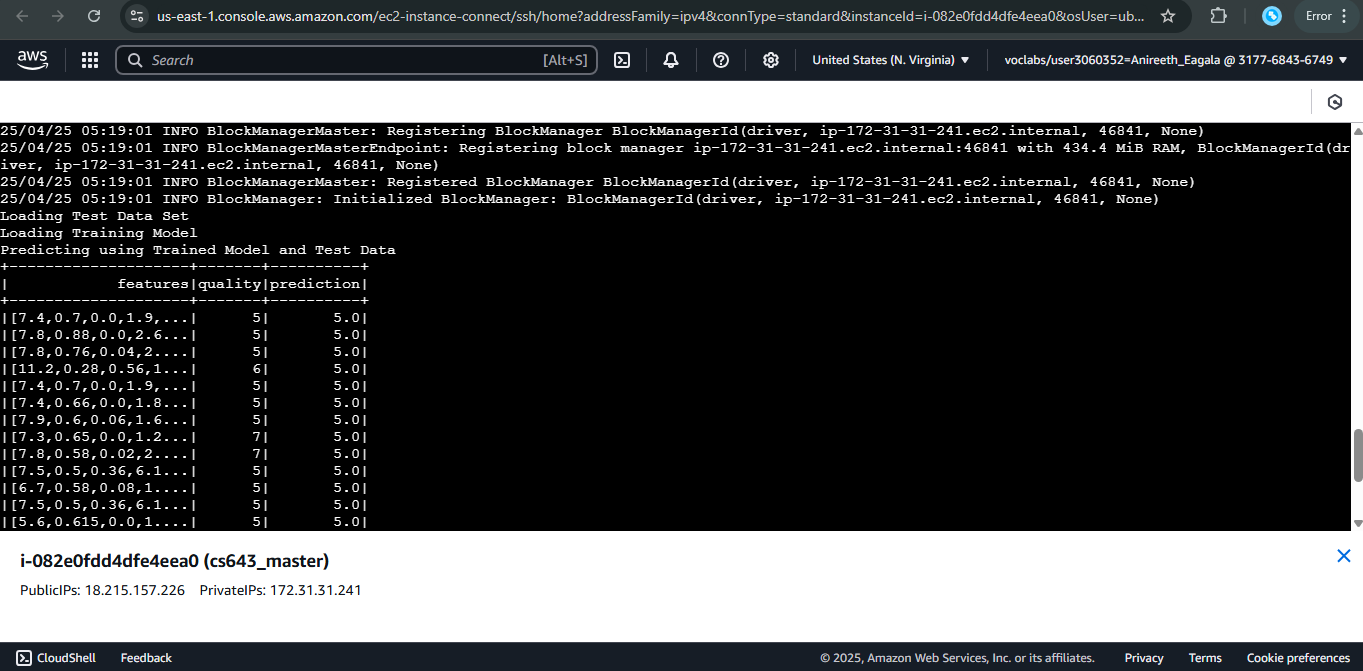
**3. Model Training**

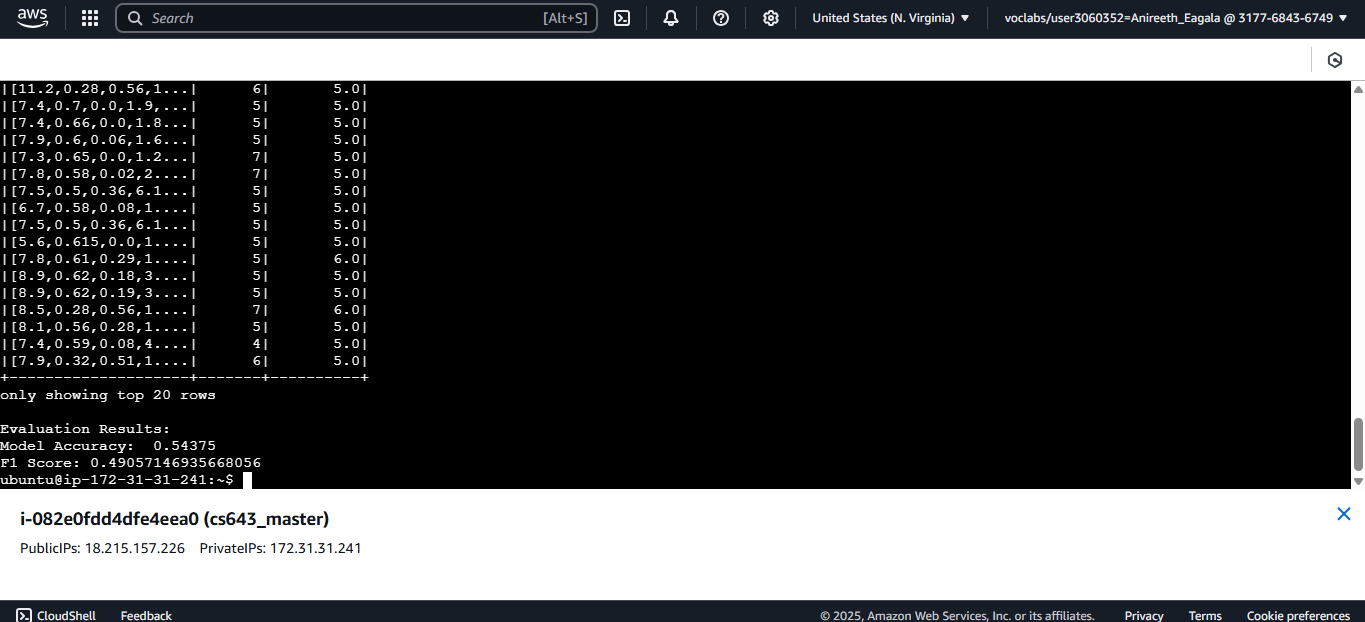
* Logistic Regression using Spark MLlib
* Saved using: LogisticRegressionModel.save("model")



**4. Prediction Pipeline**

* Load model: LogisticRegressionModel.load("model")
* Load test data
* Predict and print quality





**5. Dockerfile**

# Start from a Java base image

FROM openjdk:11

# Set environment variables

ENV SPARK\_HOME=/opt/spark

ENV PATH=$SPARK\_HOME/bin:$PATH

# Create app directory

WORKDIR /app

# Copy Spark and your app files

COPY spark-3.5.0-bin-hadoop3 /opt/spark

COPY Cs643Programming2-1.0-SNAPSHOT.jar .

COPY TrainingDataset\_cleaned.csv .

COPY ValidationDataset.csv .

COPY TestDataset.csv .

# Default command to run the JAR using Spark

CMD ["spark-submit", "--class", "edu.njit.cs643.jtacbianan.WineQualityPrediction", "--master", "local[\*]", "Cs643Programming2-1.0-SNAPSHOT.jar"]

**6. Docker Usage**

**Build the Image**

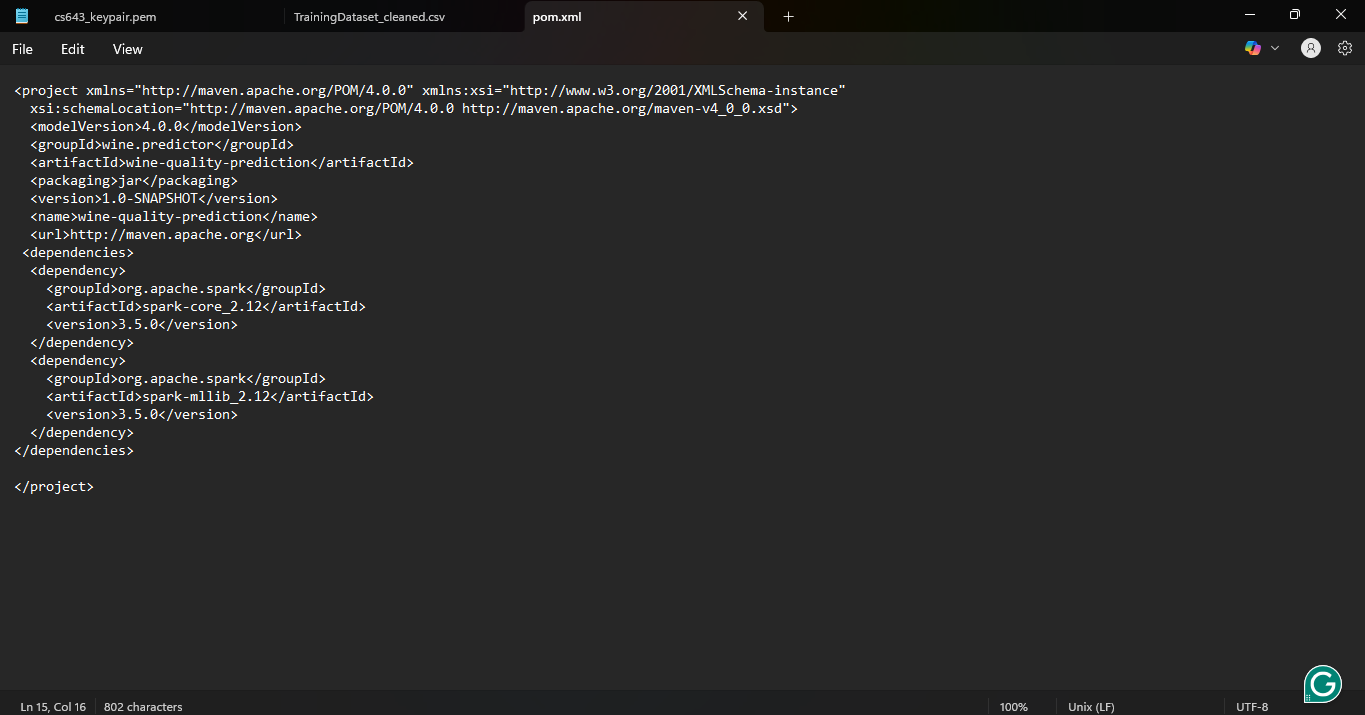
sudo docker build -t wine-quality-prediction .

**Run Prediction**

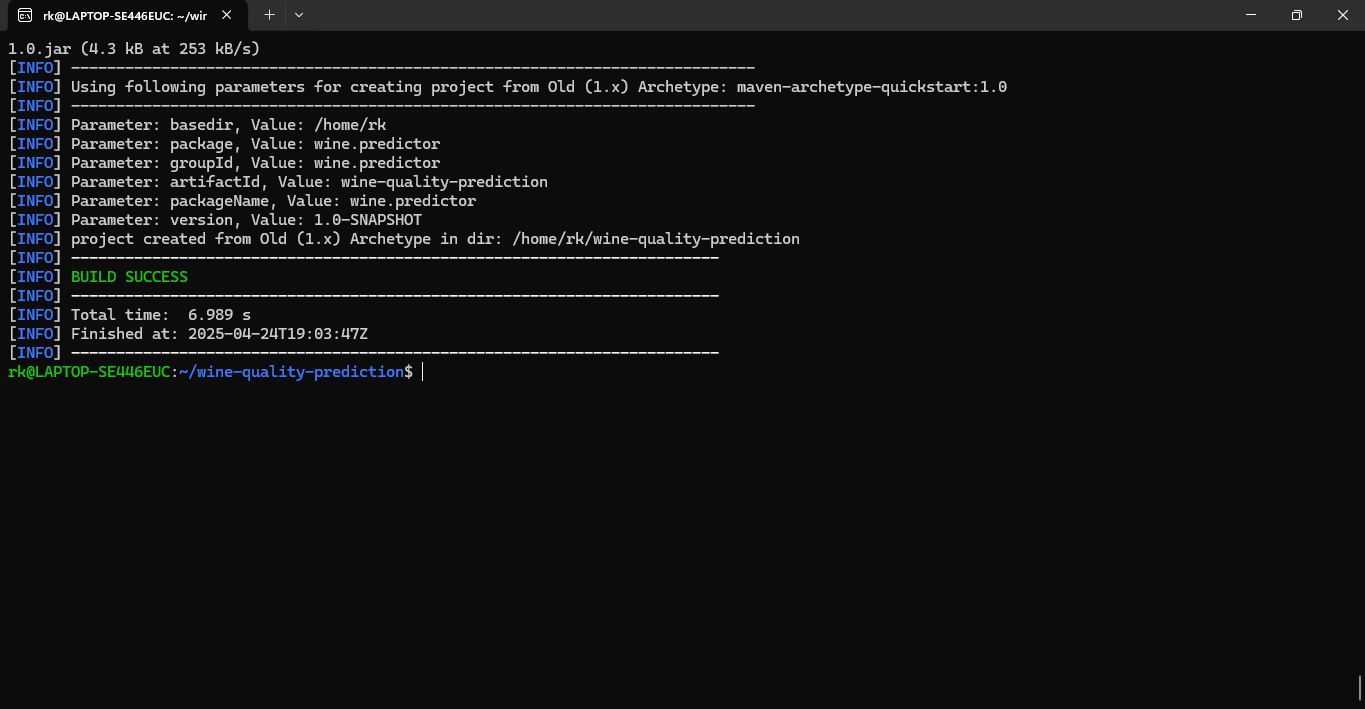
sudo docker run -v /home/ubuntu:/app wine-quality-prediction

**7. Output**

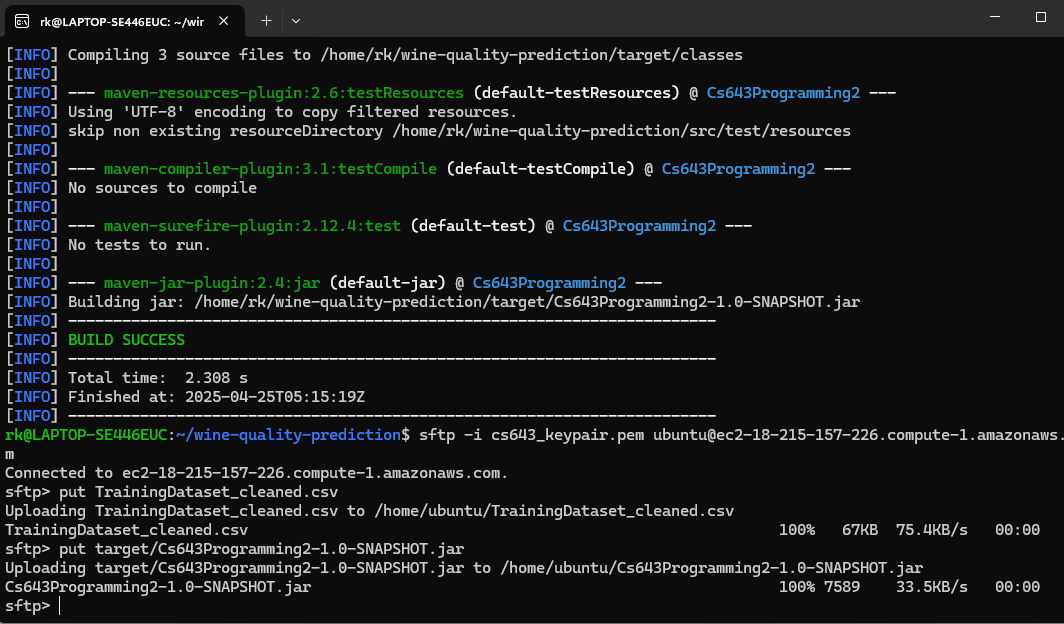
Pom.xml file:



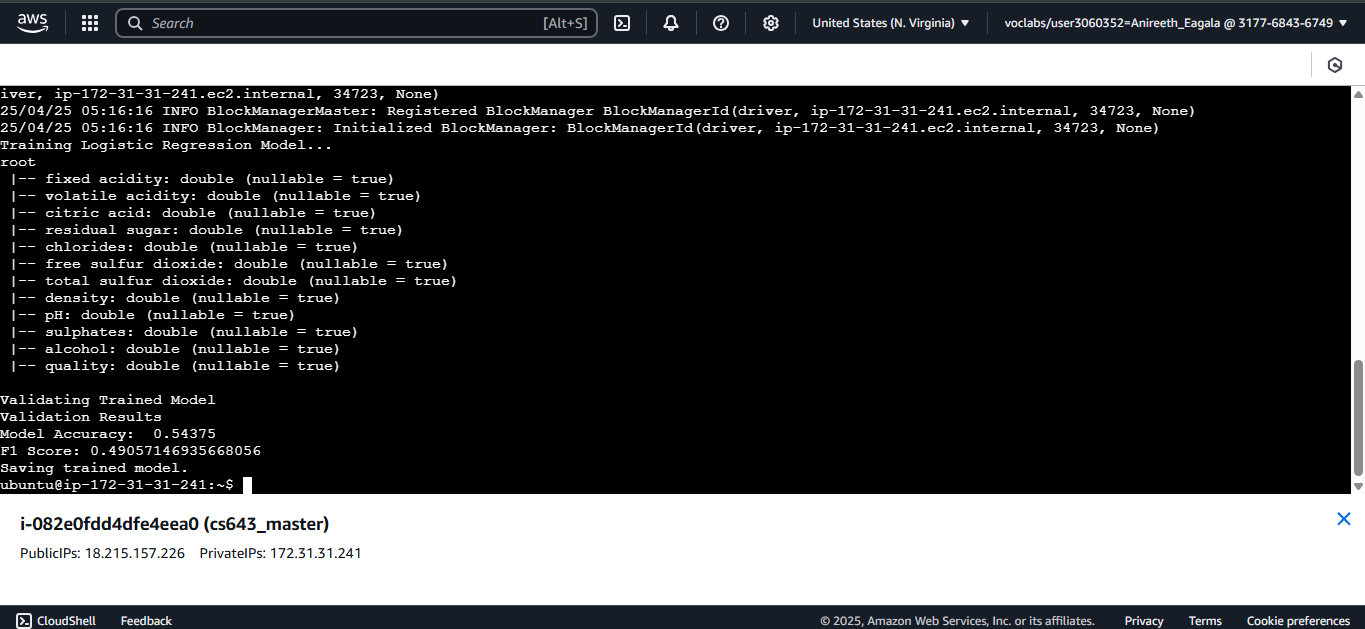
Building jar file:



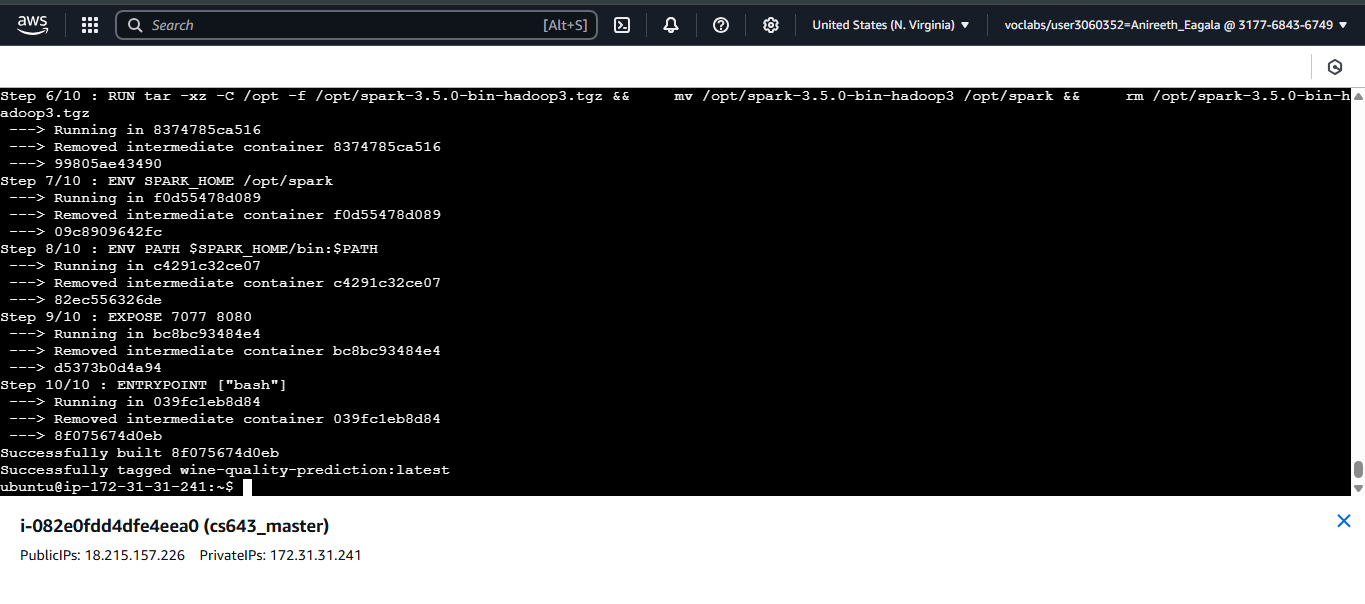
Pushing jar file and other related csv file to the EC2:

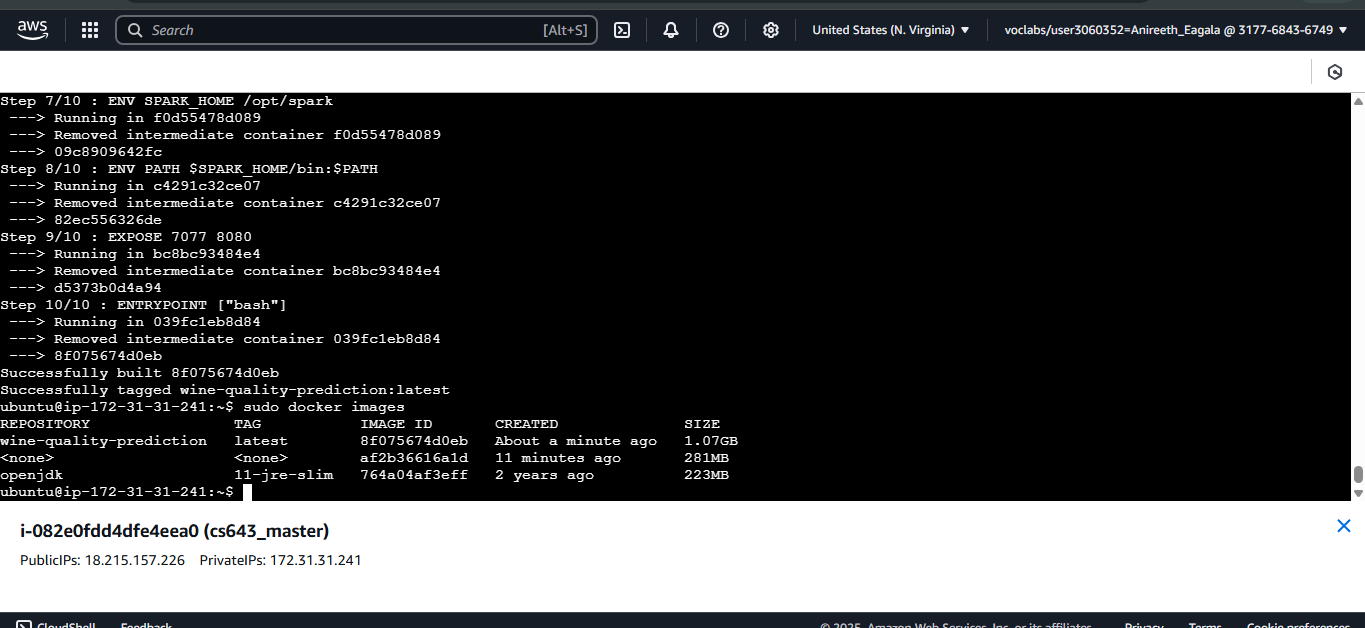


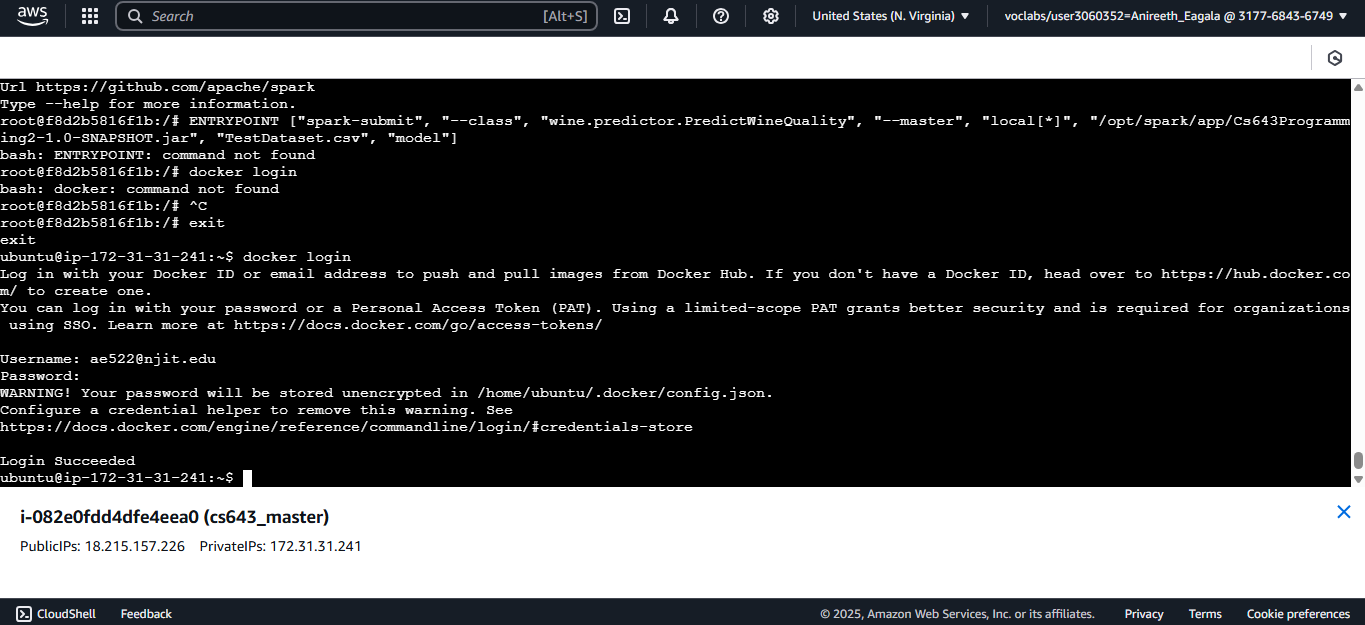
Training the Model:

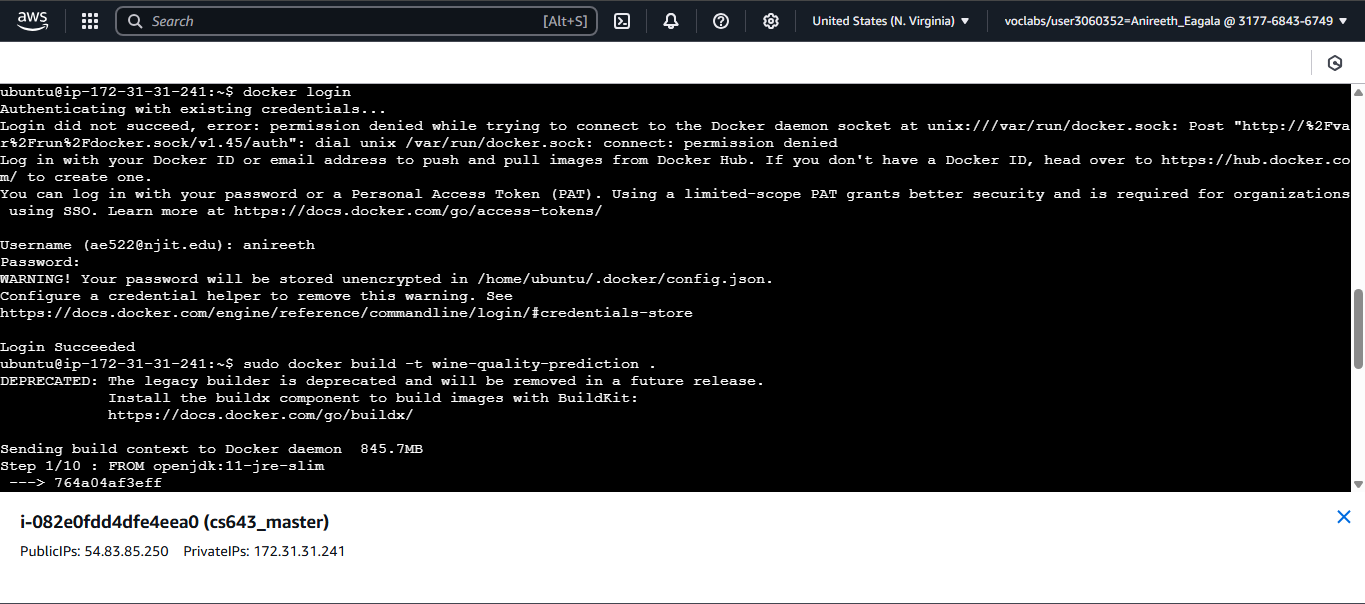


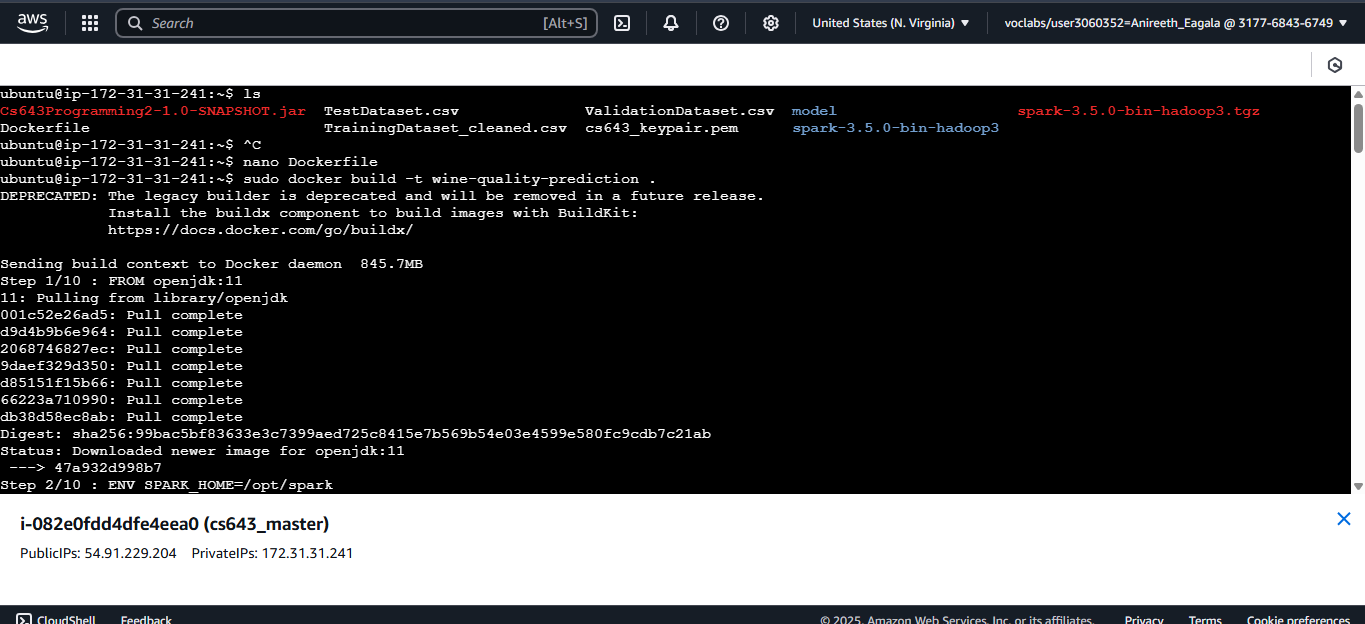
Creating Docker image:



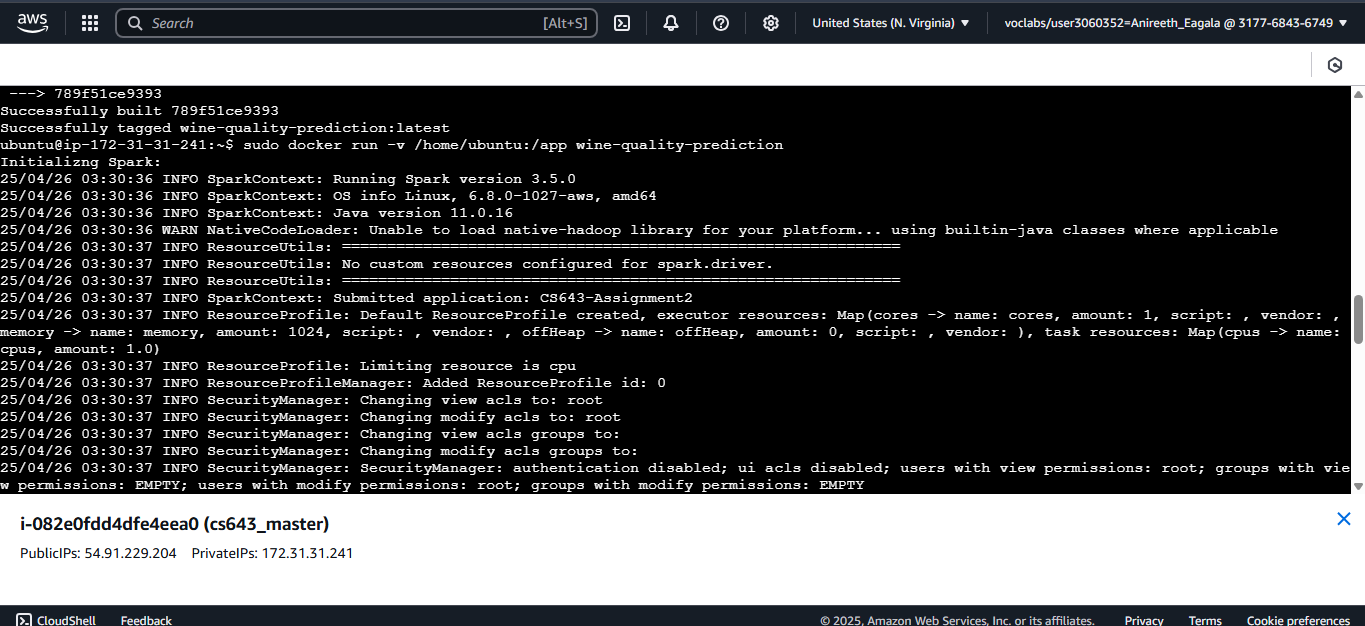




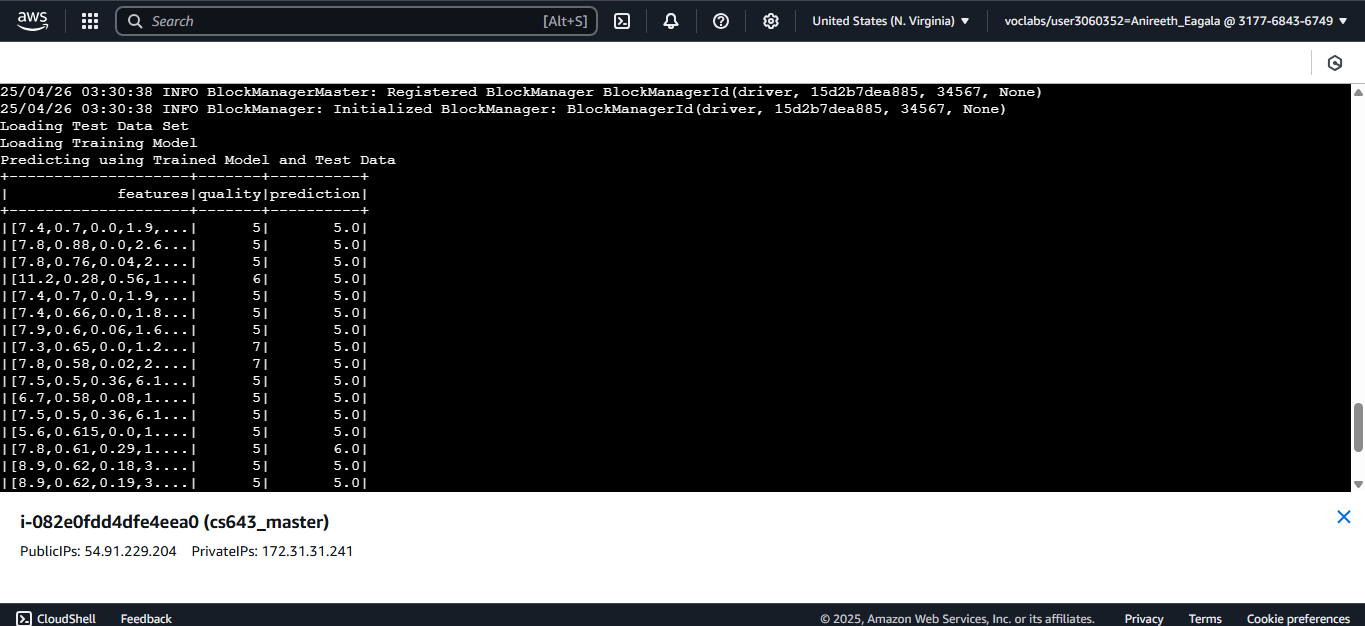


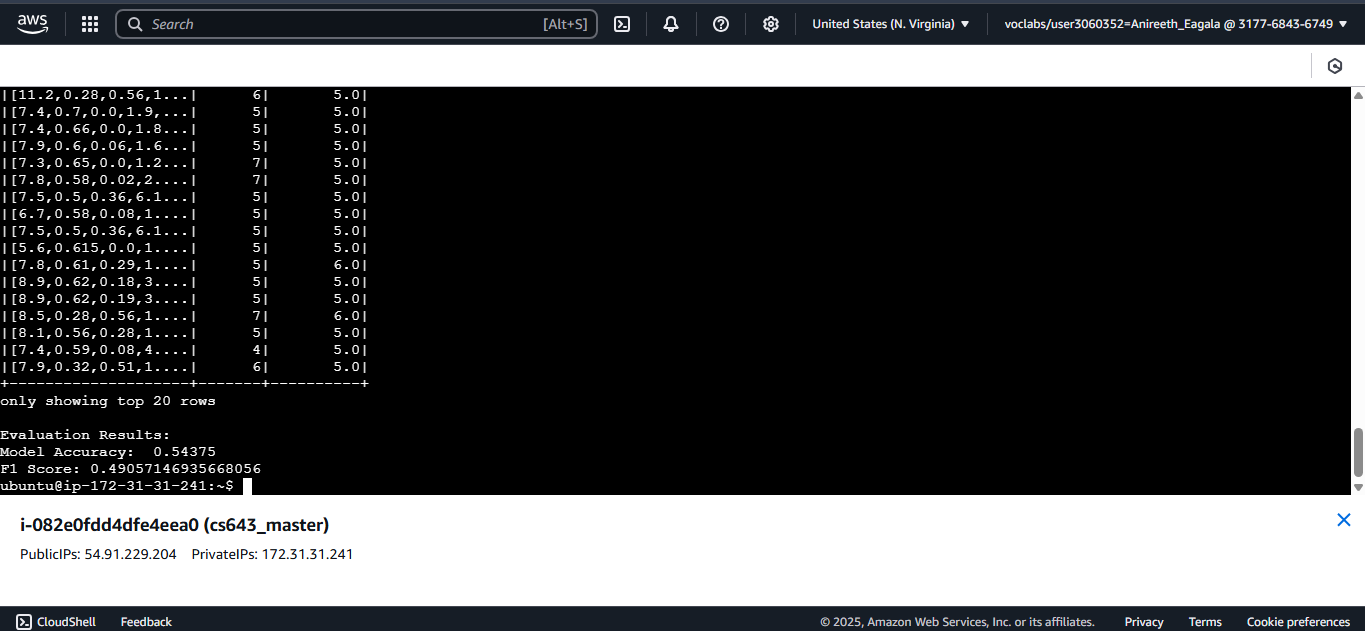


Running docker file:



Making predictions using Docker:





**8. Conclusion**

This project demonstrated:

* Building a prediction pipeline using Spark MLlib
* Preprocessing and validating data
* Dockerizing and running on an AWS EC2 instance
* Predicting wine quality using logistic regression