Camille Balo
CS643
Programming Assignment 2 README

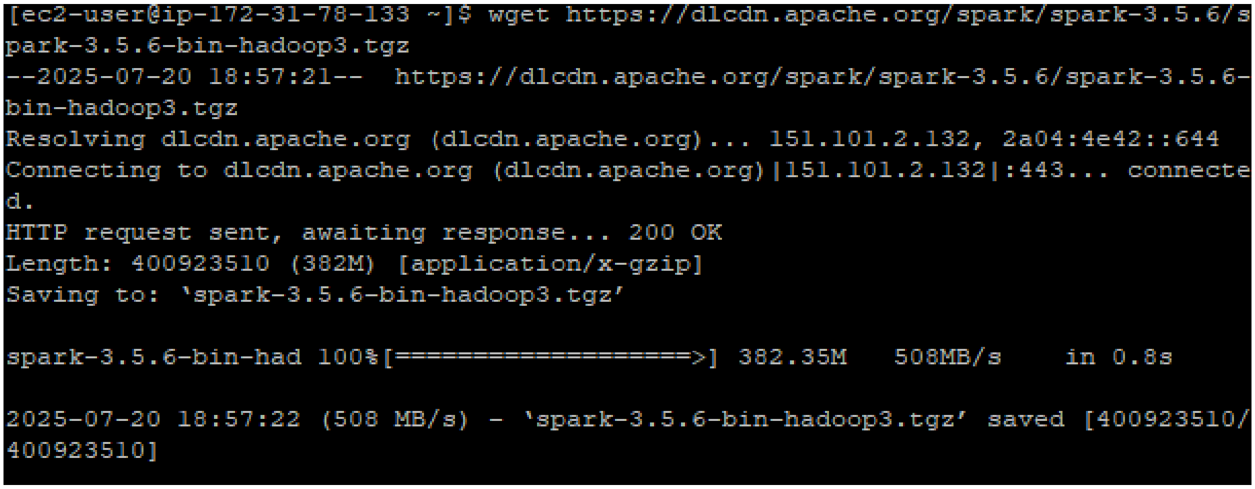
Github Link: https://github.com/cab96/CS643-Project2
Docker Link: https://hub.docker.com/r/cab96/wine-app

Creating EC2 Instances

1. Create 6 EC2 instances with the following configs (4 Spark worker nodes, 1 Spark master node, 1 application/Docker):
    a. OS: Amazon Linux 2023 kernel-6.1 AMI
    b. Instance type: t3.large
        i. I chose this one because it has a bit more memory compared to the other t3 instance types.
    c. Select key pair for login or create a new one
    d. Allow SSH, HTTP, and HTTPS from “My IP”

Configuring EC2 Instances

1. Training instances:
    a. Run: *sudo yum install java-11-amazon-corretto -y*
        i. Installs Amazon’s version of Java
    b. Run: *sudo yum install python3-pip -y*
        i. Installs Python 3
    c. Run: *wget https://dlcdn.apache.org/spark/spark-3.5.6/spark-3.5.6-bin-hadoop3.tgz*
        i. Obtained from: [Apache Download Mirrors]()
        ii. Downloads Apache Spark files
        iii. NOTE: I originally tried the latest version (4.0.0) but it gave me errors with the version of Java I installed on the EC2 instances so I downgraded to 3.5.6 of Spark which worked.



```
[ec2-user@ip-172-31-78-133 ~]$ wget https://dlcdn.apache.org/spark/spark-3.5.6/s
park-3.5.6-bin-hadoop3.tgz
--2025-07-20 18:57:21--  https://dlcdn.apache.org/spark/spark-3.5.6/spark-3.5.6-
bin-hadoop3.tgz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connecte
d.
HTTP request sent, awaiting response... 200 OK
Length: 400923510 (382M) [application/x-gzip]
Saving to: ‘spark-3.5.6-bin-hadoop3.tgz’

spark-3.5.6-bin-had 100%[===================>] 382.35M   508MB/s    in 0.8s

2025-07-20 18:57:22 (508 MB/s) - ‘spark-3.5.6-bin-hadoop3.tgz’ saved [400923510/
400923510]
```

    d. Run: *tar -xvzf spark-3.5.6-bin-hadoop3.tgz*

    i. Extracts the file

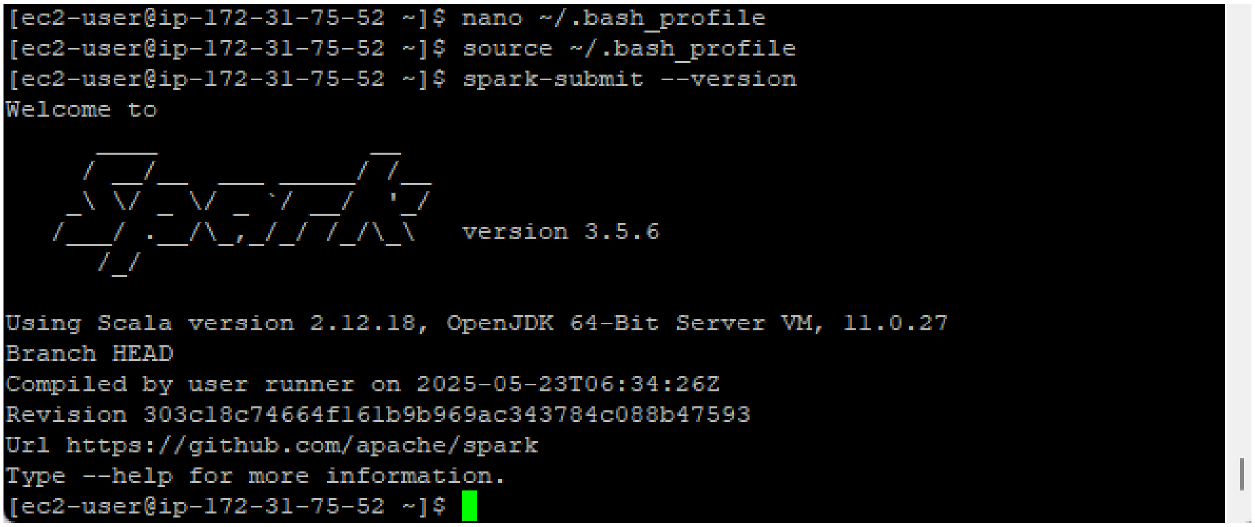
e. Run: *sudo mv spark-3.5.6-bin-hadoop3 /opt/spark*
    i. Moves Spark to /opt/spark folder

f. Run: *nano ~/.bash_profile*
    i. Add: *export SPARK_HOME=/opt/spark*
    ii. Add: *PATH=$PATH:$SPARK_HOME/bin*
    iii. Add: *export PATH*

g. Run: *source ~/.bash_profile*
    i. Reload the file

h. Run: *spark-submit --version*
    i. Verify that Spark is installed

OUTPUT:



```
[ec2-user@ip-172-31-75-52 ~]$ nano ~/.bash_profile
[ec2-user@ip-172-31-75-52 ~]$ source ~/.bash_profile
[ec2-user@ip-172-31-75-52 ~]$ spark-submit --version
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.5.6
      /_/

Using Scala version 2.12.18, OpenJDK 64-Bit Server VM, 11.0.27
Branch HEAD
Compiled by user runner on 2025-05-23T06:34:26Z
Revision 303c18c74664f161b9b969ac343784c088b47593
Url https://github.com/apache/spark
Type --help for more information.
[ec2-user@ip-172-31-75-52 ~]$
```

i. Python Installs:
    i. Pip install numpy
        1. Required for certain Pyspark functions/methods
    ii. Pip install pandas
        1. Required for certain Pyspark functions/methods
    iii. Pip install pyspark
    iv. Pip install quinn
        1. Used to help format the csv data

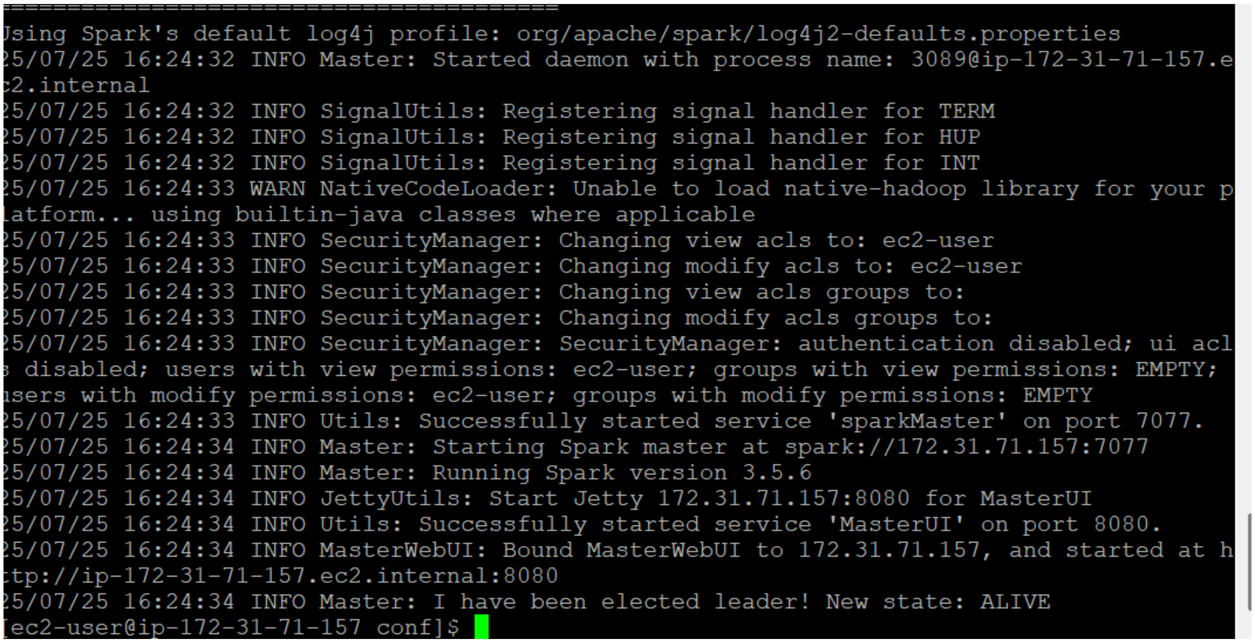
j. Run on the Application/Docker instance specifically:
    i. *Sudo yum install docker -y*
        1. Install Docker
    ii. Sudo service docker start
        1. Start Docker
    iii. Sudo systemctl enable docker
        1. Automatically start Docker on EC2 instance startup
    iv. Sudo usermod -a -G docker ec2-user
        1. Add ec2-user to the Docker group so that I do not have to use ‘sudo’ to run Docker commands

Validate Master and Slaves

1. Run: *cd /opt/spark/conf*
2. Run: *cp /opt/spark/conf/spark-env.sh.template /opt/spark/conf/spark-env.sh*
    a. Copy the template provided to create an actual spark-env.sh file
3. Run: *nano spark-env.sh*
    a. Add: *export SPARK_MASTER_HOST=master IP*
        i. On master only
    b. Add: *export SPARK_LOCAL_IP=node IP*
    c. Add: *export JAVA_HOME=/usr/lib/jvm/java-11-amazon-corretto*

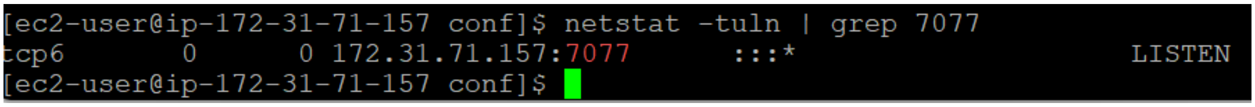
Master

4. Run: *cd /opt/spark/sbin*
5. Run: *./start-master.sh*
    a. This is the provided Spark script to start the master node
6. Run: *tail -n 20 /opt/spark/logs/*master*.out*
    a. Validate the master has been successfully started



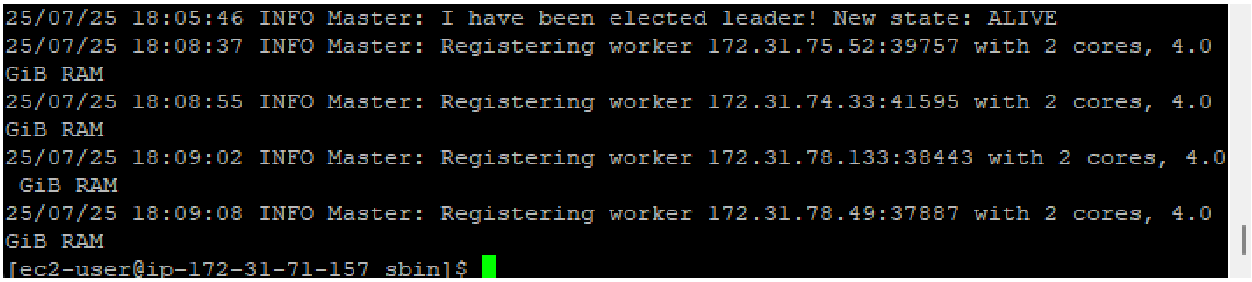
```
========================================
Jsing Spark's default log4j profile: org/apache/spark/log4j2-defaults.properties
25/07/25 16:24:32 INFO Master: Started daemon with process name: 3089@ip-172-31-71-157.e
c2.internal
25/07/25 16:24:32 INFO SignalUtils: Registering signal handler for TERM
25/07/25 16:24:32 INFO SignalUtils: Registering signal handler for HUP
25/07/25 16:24:32 INFO SignalUtils: Registering signal handler for INT
25/07/25 16:24:33 WARN NativeCodeLoader: Unable to load native-hadoop library for your p
latform... using builtin-java classes where applicable
25/07/25 16:24:33 INFO SecurityManager: Changing view acls to: ec2-user
25/07/25 16:24:33 INFO SecurityManager: Changing modify acls to: ec2-user
25/07/25 16:24:33 INFO SecurityManager: Changing view acls groups to:
25/07/25 16:24:33 INFO SecurityManager: Changing modify acls groups to:
25/07/25 16:24:33 INFO SecurityManager: SecurityManager: authentication disabled; ui acl
s disabled; users with view permissions: ec2-user; groups with view permissions: EMPTY;
isers with modify permissions: ec2-user; groups with modify permissions: EMPTY
25/07/25 16:24:33 INFO Utils: Successfully started service 'sparkMaster' on port 7077.
25/07/25 16:24:34 INFO Master: Starting Spark master at spark://172.31.71.157:7077
25/07/25 16:24:34 INFO Master: Running Spark version 3.5.6
25/07/25 16:24:34 INFO JettyUtils: Start Jetty 172.31.71.157:8080 for MasterUI
25/07/25 16:24:34 INFO Utils: Successfully started service 'MasterUI' on port 8080.
25/07/25 16:24:34 INFO MasterWebUI: Bound MasterWebUI to 172.31.71.157, and started at h
ttp://ip-172-31-71-157.ec2.internal:8080
25/07/25 16:24:34 INFO Master: I have been elected leader! New state: ALIVE
[ec2-user@ip-172-31-71-157 conf]$
```

7. Run: netstat -tuln | grep 7077
    a. Validate the master is listening to port 7077



```
[ec2-user@ip-172-31-71-157 conf]$ netstat -tuln | grep 7077
tcp6       0      0 172.31.71.157:7077      :::*                    LISTEN
[ec2-user@ip-172-31-71-157 conf]$
```

8. Once workers are started, verify the master has them registered

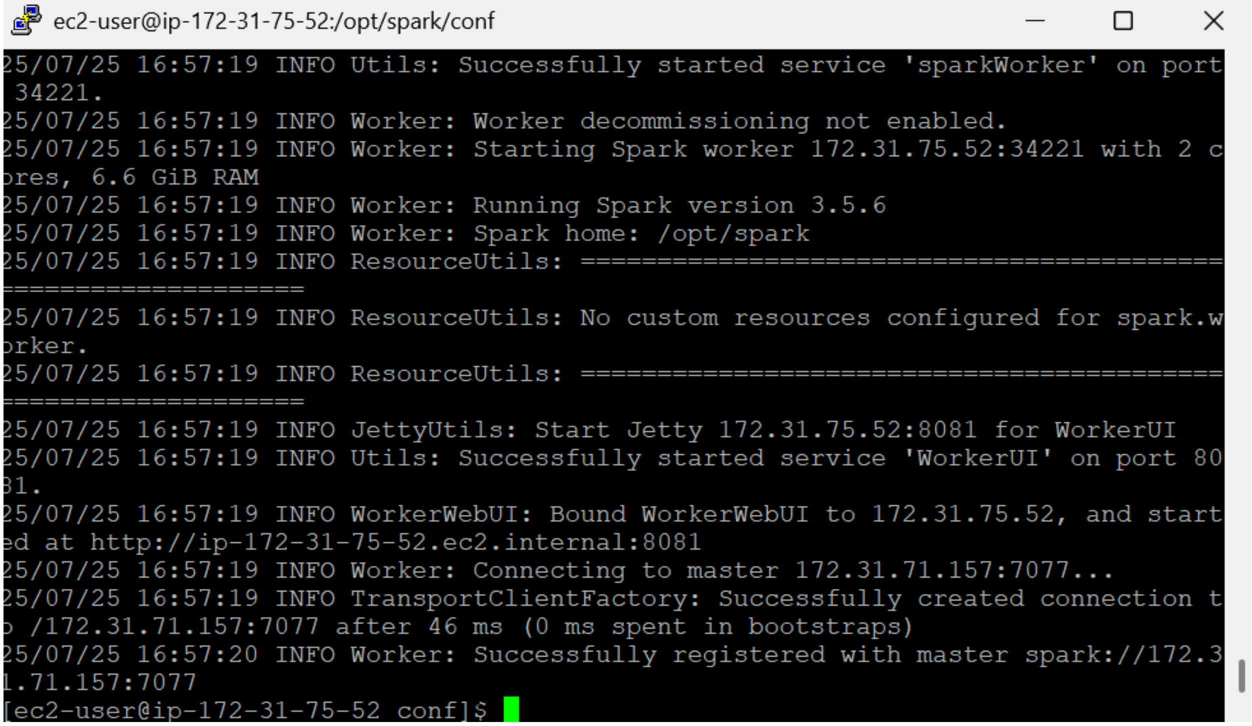


```
25/07/25 18:05:46 INFO Master: I have been elected leader! New state: ALIVE
25/07/25 18:08:37 INFO Master: Registering worker 172.31.75.52:39757 with 2 cores, 4.0
GiB RAM
25/07/25 18:08:55 INFO Master: Registering worker 172.31.74.33:41595 with 2 cores, 4.0
GiB RAM
25/07/25 18:09:02 INFO Master: Registering worker 172.31.78.133:38443 with 2 cores, 4.0
 GiB RAM
25/07/25 18:09:08 INFO Master: Registering worker 172.31.78.49:37887 with 2 cores, 4.0
GiB RAM
[ec2-user@ip-172-31-71-157 sbin]$
```

9. Create an NFS shared drive that will be where the master and nodes save the model to
    a. *Sudo yum install -y nfs-utils*
        i. Install NFS-Utils
    b. *Sudo systemctl start nfs-server*
        i. Manually start the NFS-server service
    c. *Sudo systemctl enable nfs-server*
        i. Configure for the NFS-Server to start when the EC2 instance starts
    d. *Sudo mkdir -p /mnt/shared*
        i. Create /mnt/shared directory
    e. *Sudo chown ec2-user:ec2-user /mnt/shared*
        i. Make ec2-user the owner of /mnt/shared so ‘sudo’ is not necessary
    f. *Sudo nano /etc/exports*
        i. Add: */mnt/shared 172.31.0.0/16(rw,sync,no_subtree_check)*
        ii. This shared /mnt/shared directory with any EC2 instance on the 172.31.0.0/16 subnet (which all of my EC2 instances were on)
    g. *Sudo exportfs -rav*
        i. Load the new rules in /etc/exports
    h. *Sudo systemctl restart nfs-server*
        i. Restart service for configurations to take effect

Workers

1. Run: *cd /opt/spark/sbin*
2. Run: *./start-worker.sh spark://masterIP:7077*
    a. Start the worker node using Spark’s start-worker script
3. Run: *tail -n 30 $SPARK_HOME/logs/*worker*.out*
    a. Validate the worker is running



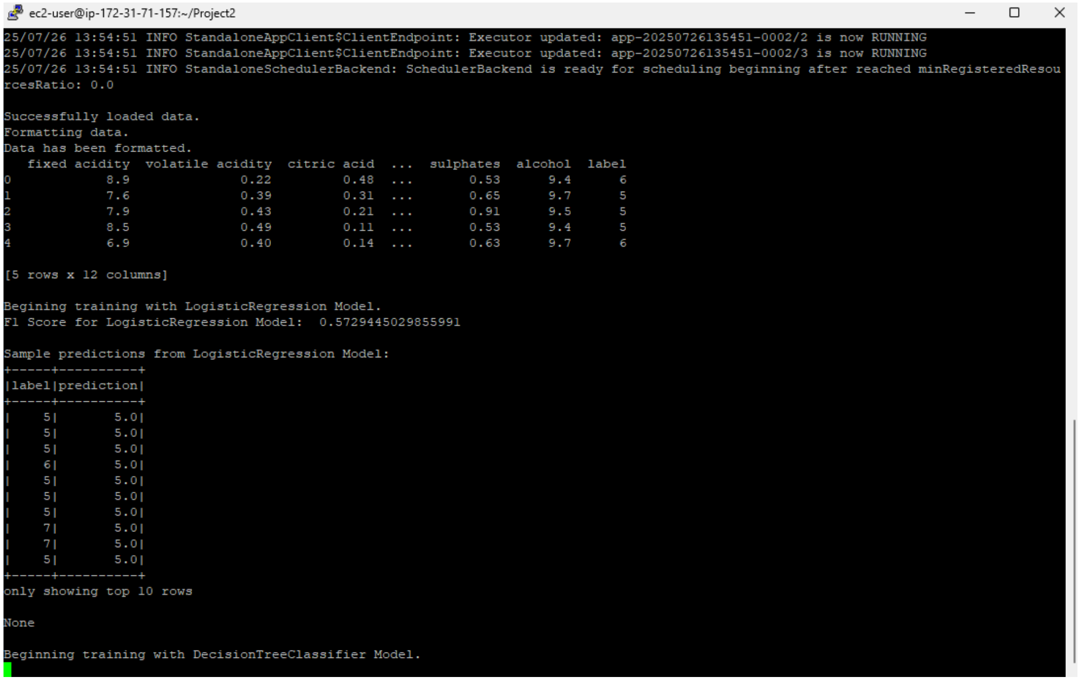
ec2-user@ip-172-31-75-52:/opt/spark/conf

```
25/07/25 16:57:19 INFO Utils: Successfully started service 'sparkWorker' on port
 34221.
25/07/25 16:57:19 INFO Worker: Worker decommissioning not enabled.
25/07/25 16:57:19 INFO Worker: Starting Spark worker 172.31.75.52:34221 with 2 c
ores, 6.6 GiB RAM
25/07/25 16:57:19 INFO Worker: Running Spark version 3.5.6
25/07/25 16:57:19 INFO Worker: Spark home: /opt/spark
25/07/25 16:57:19 INFO ResourceUtils: ==========================================
====================
25/07/25 16:57:19 INFO ResourceUtils: No custom resources configured for spark.w
orker.
25/07/25 16:57:19 INFO ResourceUtils: ==========================================
====================
25/07/25 16:57:19 INFO JettyUtils: Start Jetty 172.31.75.52:8081 for WorkerUI
25/07/25 16:57:19 INFO Utils: Successfully started service 'WorkerUI' on port 80
81.
25/07/25 16:57:19 INFO WorkerWebUI: Bound WorkerWebUI to 172.31.75.52, and start
ed at http://ip-172-31-75-52.ec2.internal:8081
25/07/25 16:57:19 INFO Worker: Connecting to master 172.31.71.157:7077...
25/07/25 16:57:19 INFO TransportClientFactory: Successfully created connection t
o /172.31.71.157:7077 after 46 ms (0 ms spent in bootstraps)
25/07/25 16:57:20 INFO Worker: Successfully registered with master spark://172.3
1.71.157:7077
[ec2-user@ip-172-31-75-52 conf]$
```

4. Mount the NFS shared drive
    a. *Sudo yum install -y nfs-utils*
    b. *Sudo mkdir -p /mnt/shared*
    c. *Sudo mount -t nfs4 172.31.71.157:/mnt/shared /mnt/shared*
        i. Mount and connect to the NFS-server /mnt/shared (which is on the master)
    d. NOTE: DO THIS ON THE APPLICATION EC2 INSTANCE TOO!

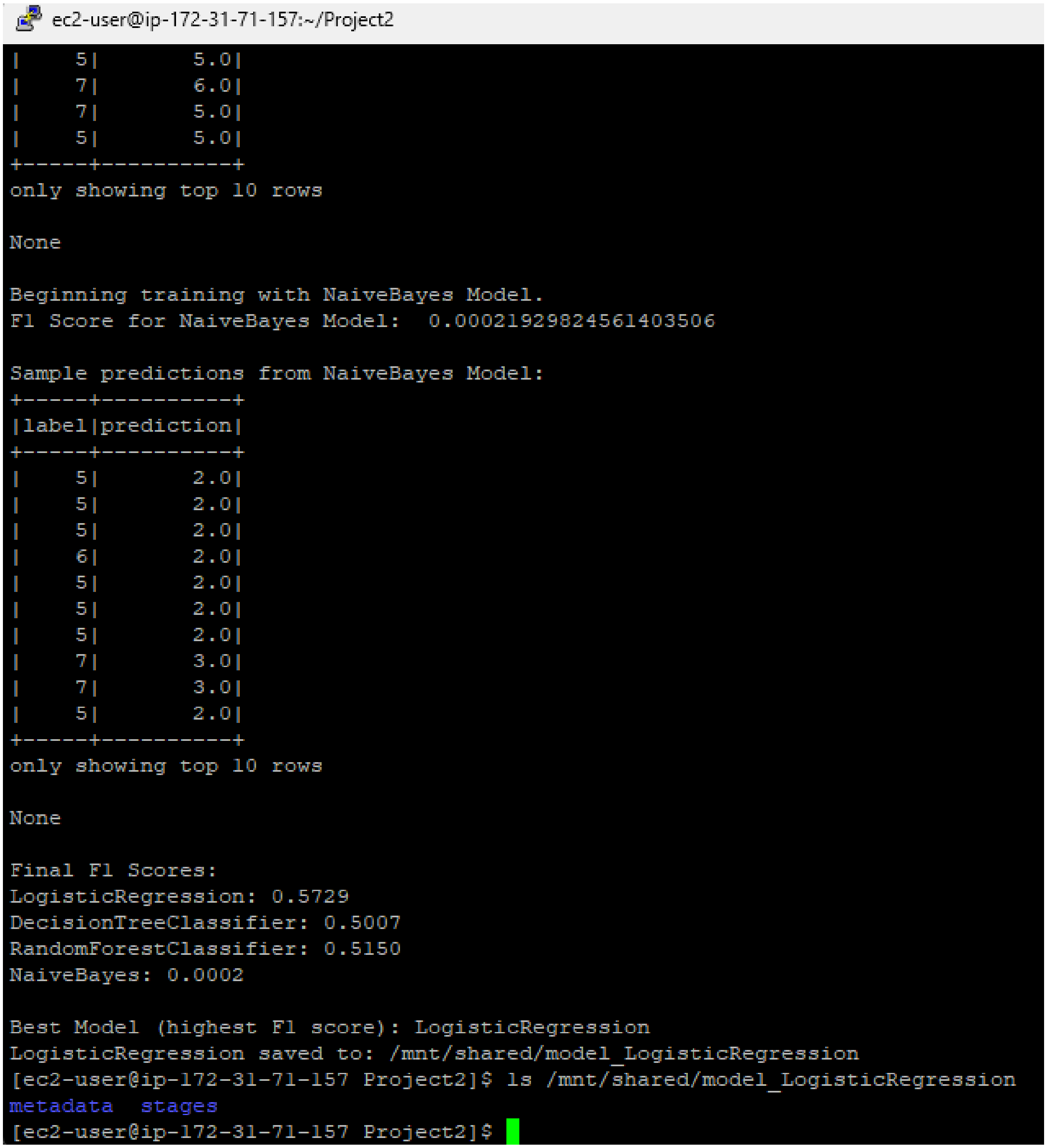
Train the Models

1. Copy the Python and CSV files to the Master and Workers
    a. Path: /home/ec2-user/Project2
2. Run: */opt/spark/bin/spark-submit --master spark ://172.31.71.157:7077 /home/ec2-user/Project2/Train_Models.py* on the master
3. OUTPUT:



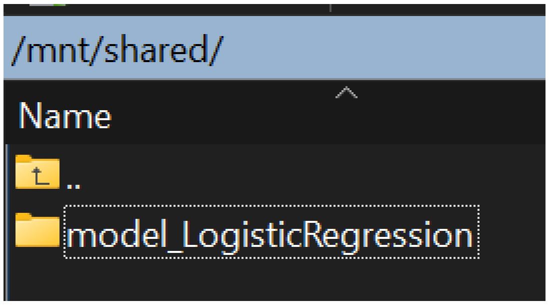
ec2-user@ip-172-31-71-157:~/Project2

```
25/07/26 13:54:51 INFO StandaloneAppClient$ClientEndpoint: Executor updated: app-20250726135451-0002/2 is now RUNNING
25/07/26 13:54:51 INFO StandaloneAppClient$ClientEndpoint: Executor updated: app-20250726135451-0002/3 is now RUNNING
25/07/26 13:54:51 INFO StandaloneSchedulerBackend: SchedulerBackend is ready for scheduling beginning after reached minRegisteredResou
rcesRatio: 0.0

Successfully loaded data.
Formatting data.
Data has been formatted.
   fixed acidity  volatile acidity  citric acid  ...  sulphates  alcohol  label
0            8.9              0.22         0.48  ...       0.53      9.4      6
1            7.6              0.39         0.31  ...       0.65      9.7      5
2            7.9              0.43         0.21  ...       0.91      9.5      5
3            8.5              0.49         0.11  ...       0.53      9.4      5
4            6.9              0.40         0.14  ...       0.63      9.7      6

[5 rows x 12 columns]

Begining training with LogisticRegression Model.
F1 Score for LogisticRegression Model:  0.5729445029855991

Sample predictions from LogisticRegression Model:
+-----+----------+
|label|prediction|
+-----+----------+
|    5|       5.0|
|    5|       5.0|
|    5|       5.0|
|    6|       5.0|
|    5|       5.0|
|    5|       5.0|
|    5|       5.0|
|    7|       5.0|
|    7|       5.0|
|    5|       5.0|
+-----+----------+
only showing top 10 rows

None

Beginning training with DecisionTreeClassifier Model.
```



ec2-user@ip-172-31-71-157:~/Project2

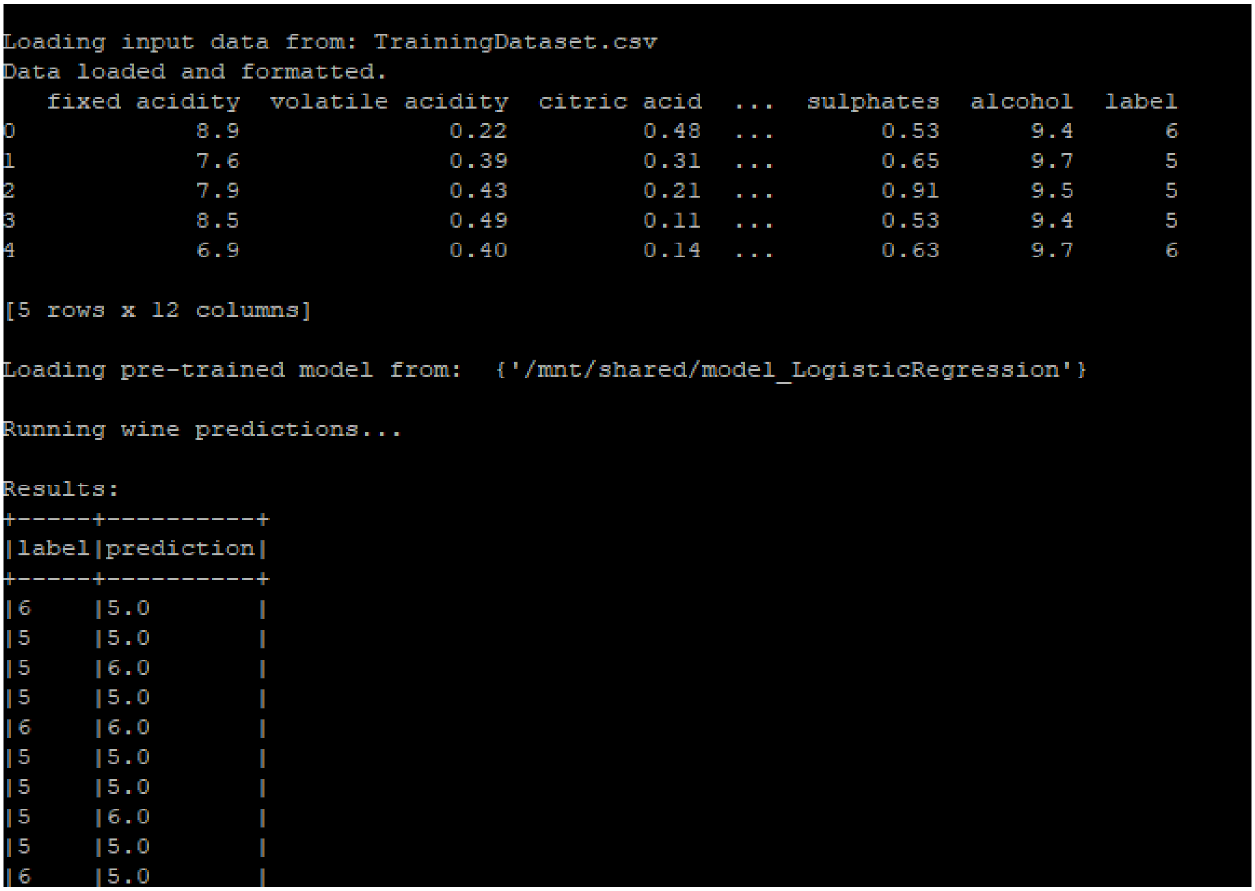
```
|    5|       5.0|
|    7|       6.0|
|    7|       5.0|
|    5|       5.0|
+-----+----------+
only showing top 10 rows

None

Beginning training with NaiveBayes Model.
F1 Score for NaiveBayes Model:  0.00021929824561403506

Sample predictions from NaiveBayes Model:
+-----+----------+
|label|prediction|
+-----+----------+
|    5|       2.0|
|    5|       2.0|
|    5|       2.0|
|    6|       2.0|
|    5|       2.0|
|    5|       2.0|
|    5|       2.0|
|    7|       3.0|
|    7|       3.0|
|    5|       2.0|
+-----+----------+
only showing top 10 rows

None

Final F1 Scores:
LogisticRegression: 0.5729
DecisionTreeClassifier: 0.5007
RandomForestClassifier: 0.5150
NaiveBayes: 0.0002

Best Model (highest F1 score): LogisticRegression
LogisticRegression saved to: /mnt/shared/model_LogisticRegression
[ec2-user@ip-172-31-71-157 Project2]$ ls /mnt/shared/model_LogisticRegression
metadata  stages
[ec2-user@ip-172-31-71-157 Project2]$
```



/mnt/shared/

Running the application without Docker:

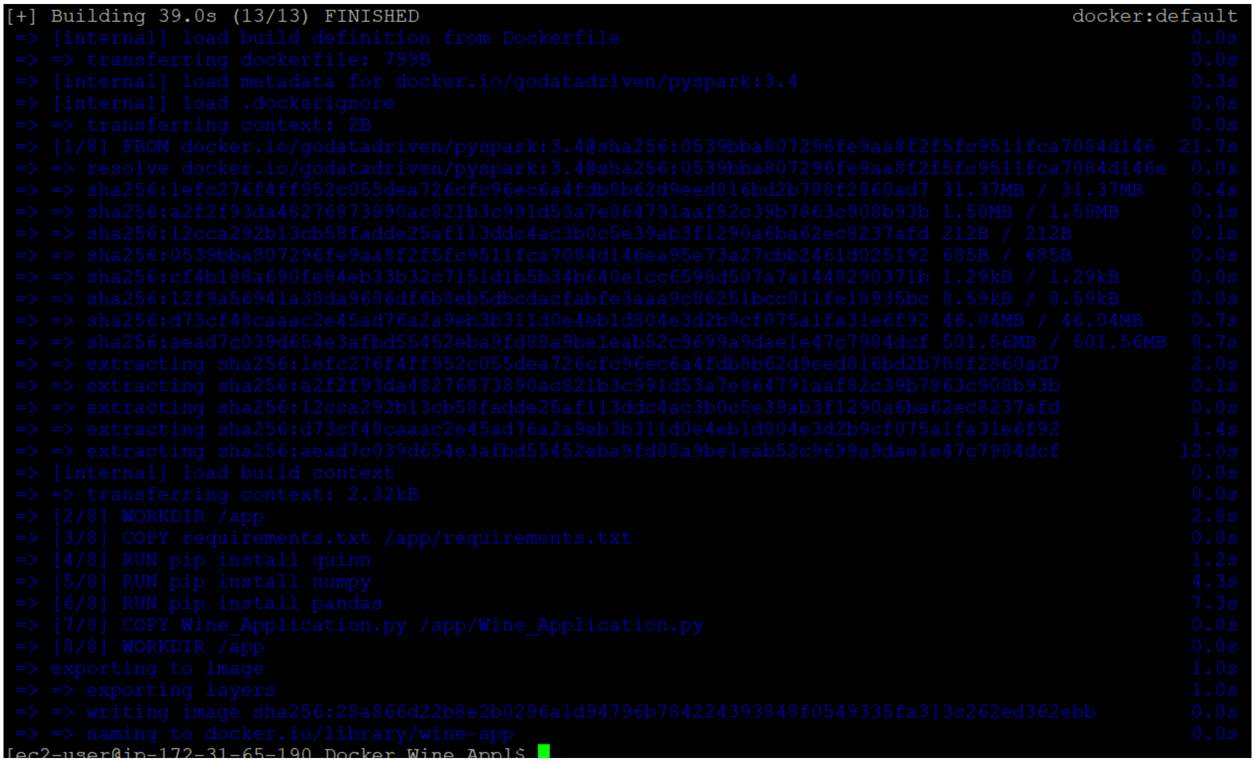
1. Run: *cd /home/ec2-user/Project2*
2. Run: */opt/spark/bin/spark-submit Wine_Application.py <file.csv>*



```
Loading input data from: TrainingDataset.csv
Data loaded and formatted.
   fixed acidity  volatile acidity  citric acid  ...  sulphates  alcohol  label
0            8.9              0.22         0.48  ...       0.53      9.4      6
1            7.6              0.39         0.31  ...       0.65      9.7      5
2            7.9              0.43         0.21  ...       0.91      9.5      5
3            8.5              0.49         0.11  ...       0.53      9.4      5
4            6.9              0.40         0.14  ...       0.63      9.7      6

[5 rows x 12 columns]

Loading pre-trained model from:  {'/mnt/shared/model_LogisticRegression'}

Running wine predictions...

Results:
+-----+----------+
|label|prediction|
+-----+----------+
|6    |5.0       |
|5    |5.0       |
|5    |6.0       |
|5    |5.0       |
|6    |6.0       |
|5    |5.0       |
|5    |5.0       |
|5    |6.0       |
|5    |5.0       |
|6    |5.0       |
```

Setting up Docker:

1. Create a new directory for the Docker files
2. Create requirements.txt with the required dependencies
3. Create Dockerfile
4. Run: *docker build -t wine-app .*

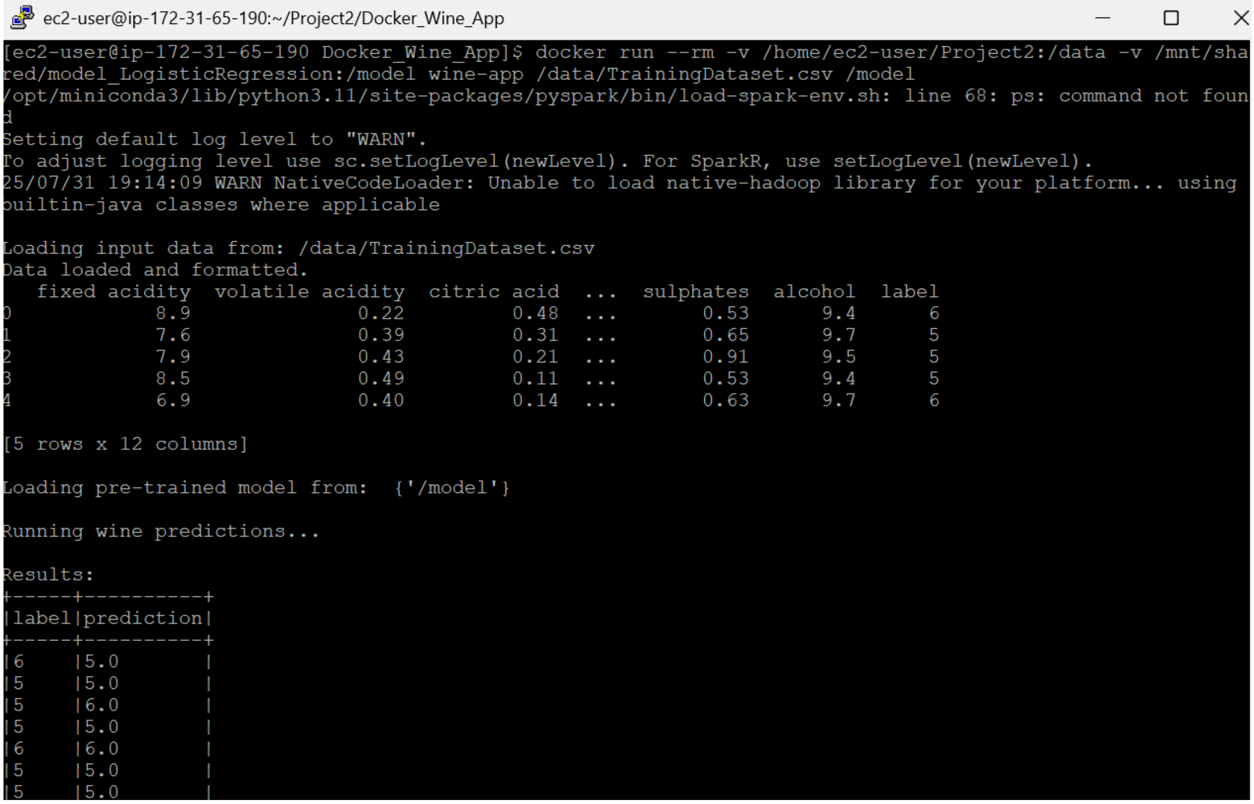


```
[+] Building 39.0s (13/13) FINISHED                                                        docker:default
 => [internal] load build definition from Dockerfile                                                0.0s
 => => transferring dockerfile: 799B                                                                0.0s
 => [internal] load metadata for docker.io/godatadriven/pyspark:3.4                                 0.3s
 => [internal] load .dockerignore                                                                   0.0s
 => => transferring context: 2B                                                                     0.0s
 => [1/8] FROM docker.io/godatadriven/pyspark:3.4@sha256:0539bba807296fe9aa8f2f5fc9511fca7084d146  21.7s
 => => resolve docker.io/godatadriven/pyspark:3.4@sha256:0539bba807296fe9aa8f2f5fc9511fca7084d146e  0.0s
 => => sha256:1efc276f4ff952c055dea726cfc96ec6a4fdb8b62d9eed816bd2b788f2860ad7 31.37MB / 31.37MB    0.4s
 => => sha256:a2f2f93da48276873890ac821b3c991d53a7e864791aaf82c39b7863c908b93b 1.58MB / 1.58MB      0.1s
 => => sha256:12cca292b13cb58fadde25af113ddc4ac3b0c5e39ab3f1290a6ba62ec8237afd 212B / 212B          0.1s
 => => sha256:0539bba807296fe9aa8f2f5fc9511fca7084d146ea95e73a27cbb2461d025192 685B / 685B          0.0s
 => => sha256:cf4b188a690fe84eb33b32c7151d1b5b34b640e1cc6598d507a7a1448290371b 1.29kB / 1.29kB      0.0s
 => => sha256:12f9a56941a38da9686df6b8eb5dbcdacfabfe3aaa9c86251bcc011fe18935bc 8.59kB / 8.59kB      0.0s
 => => sha256:d73cf48caaac2e45ad76a2a9eb3b311d0e4eb1d804e3d2b9cf075a1fa31e6f92 46.04MB / 46.04MB    0.7s
 => => sha256:aead7c039d654e3afbd55452eba9fd88a9be1eab52c9699a9dae1e47c7984dcf 501.56MB / 501.56MB  8.7s
 => => extracting sha256:1efc276f4ff952c055dea726cfc96ec6a4fdb8b62d9eed816bd2b788f2860ad7           2.0s
 => => extracting sha256:a2f2f93da48276873890ac821b3c991d53a7e864791aaf82c39b7863c908b93b           0.1s
 => => extracting sha256:12cca292b13cb58fadde25af113ddc4ac3b0c5e39ab3f1290a6ba62ec8237afd           0.0s
 => => extracting sha256:d73cf48caaac2e45ad76a2a9eb3b311d0e4eb1d804e3d2b9cf075a1fa31e6f92           1.4s
 => => extracting sha256:aead7c039d654e3afbd55452eba9fd88a9be1eab52c9699a9dae1e47c7984dcf          12.0s
 => [internal] load build context                                                                   0.0s
 => => transferring context: 2.32kB                                                                 0.0s
 => [2/8] WORKDIR /app                                                                              2.8s
 => [3/8] COPY requirements.txt /app/requirements.txt                                               0.0s
 => [4/8] RUN pip install quinn                                                                     1.2s
 => [5/8] RUN pip install numpy                                                                     4.3s
 => [6/8] RUN pip install pandas                                                                    7.3s
 => [7/8] COPY Wine_Application.py /app/Wine_Application.py                                         0.0s
 => [8/8] WORKDIR /app                                                                              0.0s
 => exporting to image                                                                              1.0s
 => => exporting layers                                                                             1.0s
 => => writing image sha256:28a866d22b8e2b0296a1d94796b784224393848f0549335fa313c262ed362ebb        0.0s
 => => naming to docker.io/library/wine-app                                                         0.0s
[ec2-user@ip-172-31-65-190 Docker Wine App]$
```

Running the application with Docker:

1. Run: *docker run --rm -v /home/ec2-user/Project2:/data -v /mnt/shared/model_LogisticRegression:/model wine-app /data/{filepath}.csv /model*

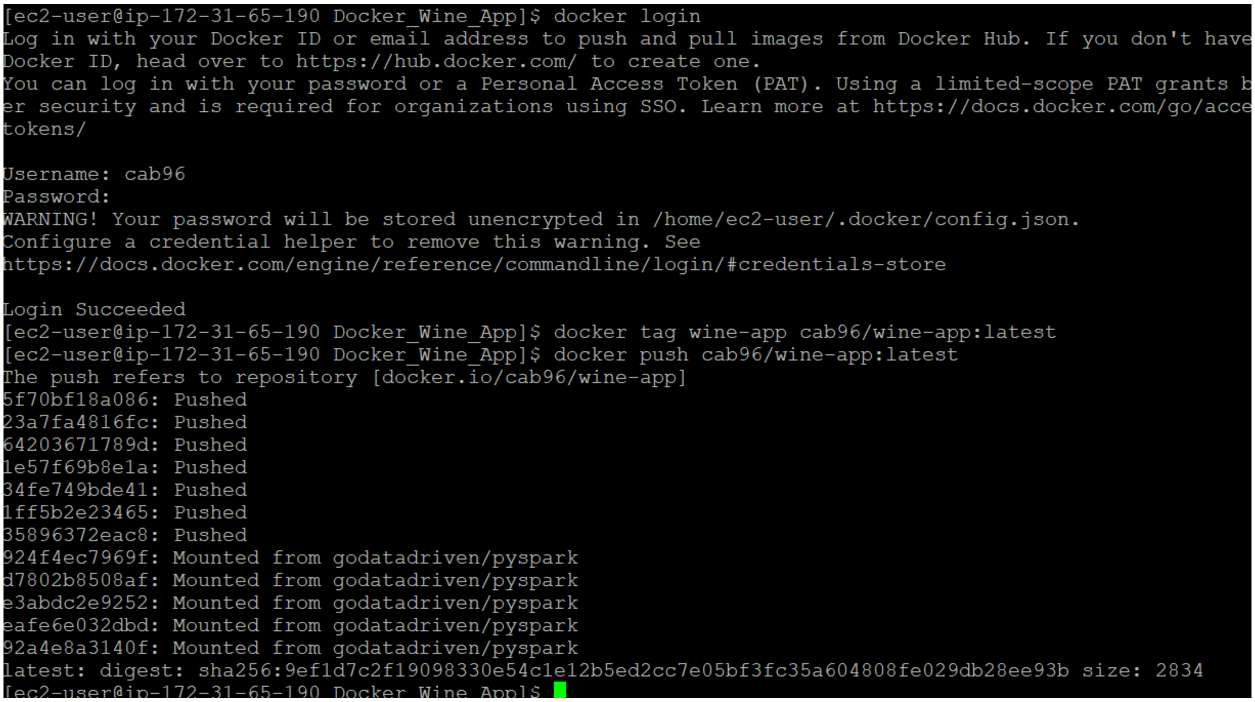
OUTPUT:



```
[ec2-user@ip-172-31-65-190 Docker_Wine_App]$ docker run --rm -v /home/ec2-user/Project2:/data -v /mnt/sha
red/model_LogisticRegression:/model wine-app /data/TrainingDataset.csv /model
/opt/miniconda3/lib/python3.11/site-packages/pyspark/bin/load-spark-env.sh: line 68: ps: command not foun
d
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
25/07/31 19:14:09 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using
builtin-java classes where applicable

Loading input data from: /data/TrainingDataset.csv
Data loaded and formatted.
   fixed acidity  volatile acidity  citric acid  ...  sulphates  alcohol  label
0            8.9              0.22         0.48  ...       0.53      9.4      6
1            7.6              0.39         0.31  ...       0.65      9.7      5
2            7.9              0.43         0.21  ...       0.91      9.5      5
3            8.5              0.49         0.11  ...       0.53      9.4      5
4            6.9              0.40         0.14  ...       0.63      9.7      6

[5 rows x 12 columns]

Loading pre-trained model from:  {'/model'}

Running wine predictions...

Results:
+-----+----------+
|label|prediction|
+-----+----------+
|6    |5.0       |
|5    |5.0       |
|5    |6.0       |
|5    |5.0       |
|6    |6.0       |
|5    |5.0       |
|5    |5.0       |
```

Save Docker container

1. Run: *docker login*
2. Run: *docker tag wine-app {userID}/wine-app:latest*
3. Run: *docker push {userID}/wine-app:latest*
4. Verify on Docker Hub that it was pushed.
    a. cab96/wine-app - Docker Image | Docker Hub



```
[ec2-user@ip-172-31-65-190 Docker_Wine_App]$ docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have
Docker ID, head over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants b
er security and is required for organizations using SSO. Learn more at https://docs.docker.com/go/acce
tokens/

Username: cab96
Password:
WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[ec2-user@ip-172-31-65-190 Docker_Wine_App]$ docker tag wine-app cab96/wine-app:latest
[ec2-user@ip-172-31-65-190 Docker_Wine_App]$ docker push cab96/wine-app:latest
The push refers to repository [docker.io/cab96/wine-app]
5f70bf18a086: Pushed
23a7fa4816fc: Pushed
64203671789d: Pushed
1e57f69b8e1a: Pushed
34fe749bde41: Pushed
1ff5b2e23465: Pushed
35896372eac8: Pushed
924f4ec7969f: Mounted from godatadriven/pyspark
d7802b8508af: Mounted from godatadriven/pyspark
e3abdc2e9252: Mounted from godatadriven/pyspark
eafe6e032dbd: Mounted from godatadriven/pyspark
92a4e8a3140f: Mounted from godatadriven/pyspark
latest: digest: sha256:9ef1d7c2f19098330e54c1e12b5ed2cc7e05bf3fc35a604808fe029db28ee93b size: 2834
[ec2-user@ip-172-31-65-190 Docker_Wine_App]$ 
```