Name: Swathi Muralinathan

Email Address: sm3345@njit.edu

Subject : CS643853-Cloud Computing

Assignment:

Module 03 Assignment 03: Programming Assignment 2

Wine Quality Prediction AWS Spark Application:

Pa2Winepred: This project involves the development of a Python application utilizing the PySpark interface.

The application is deployed on an Amazon Web Services (AWS) Elastic MapReduce (EMR) cluster. The primary objective is to parallelly train a machine learning model on EC2 instances for predicting wine quality using publicly available data. Subsequently, the trained model is employed to predict the quality of wine. Docker is utilized to create a container image for the trained machine learning model, streamlining the deployment process.

Link for GitHub:

https://github.com/Swmural/Wine

Link for Docker:

Docker

Steps for the Execution for Wine Quality Prediction AWS Spark :

1.Create a Key-pair for the EMR Cluster :go to EC2/Network/Key-pairs

Use the format of .pem and download the keypair

Created key pair as: pa2assmahi.pem

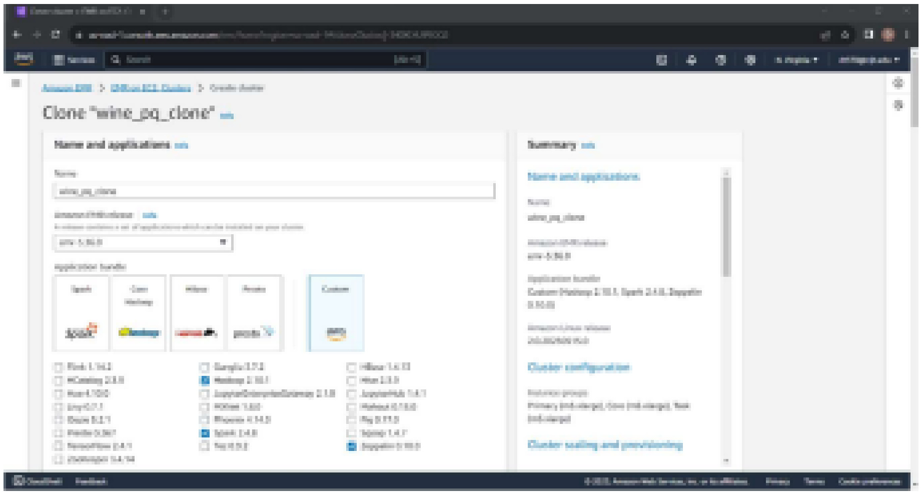
2.Create an S3 bucket

Created an S3 bucket in aws: pa2winebucket1

3.Then go to EMR console and create EMR cluster

4. Creating the spark in the AWS instance by using EMR console:

Creating the spark cluster by using the EMR console, and create the 4 instances: Name and application



Amazon EMR > EMR on EC2: Clusters > Create cluster

Clone "wine_pq_clone" Info

Name and applications Info

Name: wine_pq_clone

Amazon EMR release: emr-5.36.0

Application bundle: Spark | Core Hadoop | HBase | Presto | Custom

☐ Flink 1.14.2 ☐ Ganglia 3.7.2 ☐ HBase 1.4.13
☐ HCatalog 2.3.9 ☑ Hadoop 2.10.1 ☐ Hive 2.3.9
☐ Hue 4.10.0 ☐ JupyterEnterpriseGateway 2.1.0 ☐ JupyterHub 1.4.1
☐ Livy 0.7.1 ☐ MXNet 1.8.0 ☐ Mahout 0.13.0
☐ Oozie 5.2.1 ☐ Phoenix 4.14.3 ☐ Pig 0.17.0
☐ Presto 0.267 ☑ Spark 2.4.8 ☐ Sqoop 1.4.7
☐ TensorFlow 2.4.1 ☐ Tez 0.9.2 ☑ Zeppelin 0.10.0
☐ ZooKeeper 3.4.14

Summary Info

Name and applications

Name: wine_pq_clone

Amazon EMR release: emr-5.36.0

Application bundle: Custom (Hadoop 2.10.1, Spark 2.4.8, Zeppelin 0.10.0)

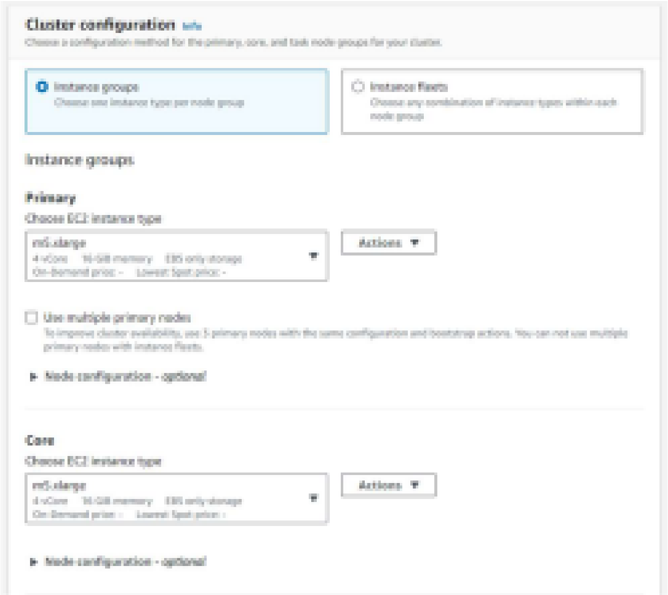
Cluster configuration

Instance groups: Primary (m5.xlarge), Core (m5.xlarge), Task (m5.xlarge)

Cluster scaling and provisioning

Note: Here it says Clone “wine_pq_clone” as I have cloned the previous configuration instead of creating from scratch to save time.

Cluster Configuration:



Cluster configuration Info

Choose a configuration method for the primary, core, and task node groups for your cluster.

● Instance groups – Choose one instance type per node group
○ Instance fleets – Choose any combination of instance types within each node group

Instance groups

Primary

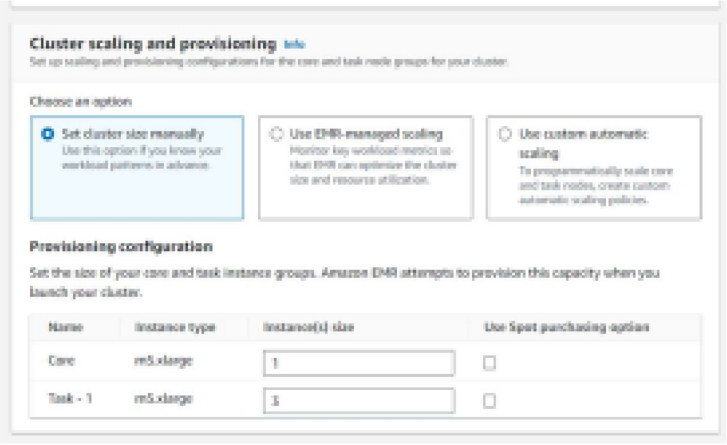
Choose EC2 instance type: m5.xlarge – 4 vCore, 16 GiB memory, EBS only storage, On-Demand price: -, Lowest Spot price: -

☐ Use multiple primary nodes – To improve cluster availability, use 3 primary nodes with the same configuration and bootstrap actions. You can not use multiple primary nodes with instance fleets.

Core

Choose EC2 instance type: m5.xlarge – 4 vCore, 16 GiB memory, EBS only storage, On-Demand price: -, Lowest Spot price: -

Cluster Scaling and provisioning:



Cluster scaling and provisioning Info

Set up scaling and provisioning configurations for the core and task node groups for your cluster.

Choose an option

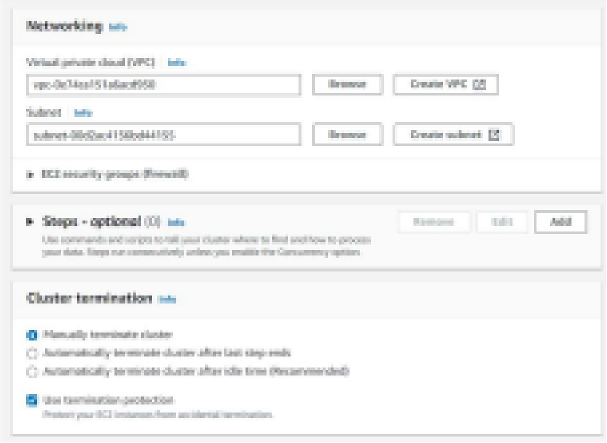
- ● Set cluster size manually
  Use this option if you know your workload patterns in advance.
- ○ Use EMR-managed scaling
  Monitor key workload metrics so that EMR can optimize the cluster size and resource utilization.
- ○ Use custom automatic scaling
  To programmatically scale core and task nodes, create custom automatic scaling policies.

Provisioning configuration

Set the size of your core and task instance groups. Amazon EMR attempts to provision this capacity when you launch your cluster.

Name	Instance type	Instance(s) size	Use Spot purchasing option
Core	m5.xlarge	1	□
Task - 1	m5.xlarge	3	□

Networking & Cluster Termination:



Networking Info

Virtual private cloud (VPC) Info

vpc-0e74ea151a6acd958

Subnet Info

subnet-08d2ac4158bd44155

▸ EC2 security groups (firewall)

▸ Steps - optional (0) Info

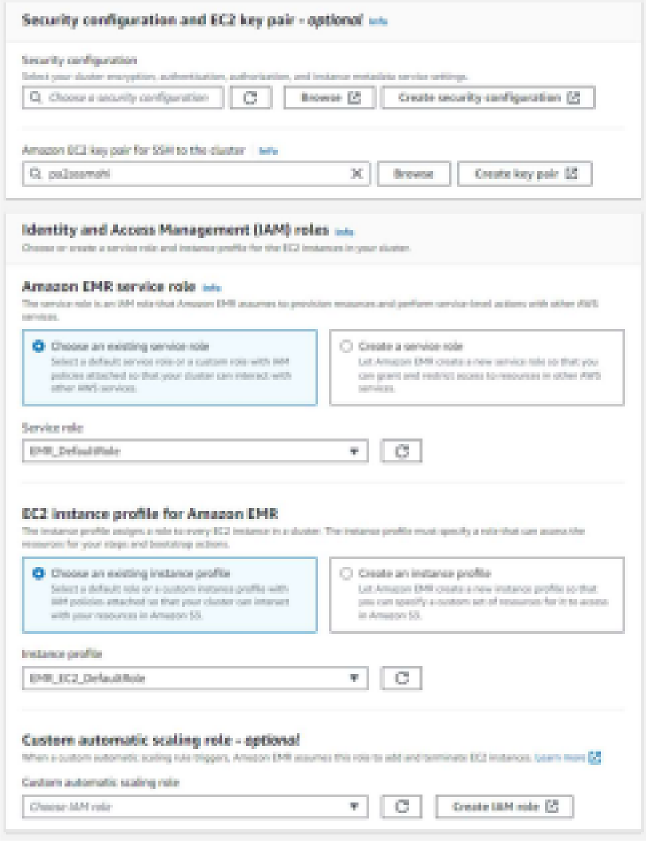
Use commands and scripts to tell your cluster where to find and how to process your data. Steps run consecutively unless you enable the Concurrency option.

Cluster termination Info

- ● Manually terminate cluster
- ○ Automatically terminate cluster after last step ends
- ○ Automatically terminate cluster after idle time (Recommended)

☑ Use termination protection
Protect your EC2 instances from accidental termination.

Security Configuration and EC2 Key pair & Identity and access management(IAM) role:



Security configuration and EC2 key pair - optional Info

Security configuration

Select your cluster encryption, authentication, authorization, and instance metadata service settings.

Choose a security configuration

Amazon EC2 key pair for SSH to the cluster Info

pa2winemahi

Identity and Access Management (IAM) roles Info

Choose or create a service role and instance profile for the EC2 instances in your cluster.

Amazon EMR service role Info

The service role is an IAM role that Amazon EMR assumes to provision resources and perform service-level actions with other AWS services.

● Choose an existing service role
Select a default service role or a custom role with IAM policies attached so that your cluster can interact with other AWS services.

○ Create a service role
Let Amazon EMR create a new service role so that you can grant and restrict access to resources in other AWS services.

Service role: EMR_DefaultRole

EC2 instance profile for Amazon EMR

The instance profile assigns a role to every EC2 instance in a cluster. The instance profile must specify a role that can access the resources for your steps and bootstrap actions.

● Choose an existing instance profile
Select a default role or a custom instance profile with IAM policies attached so that your cluster can interact with your resources in Amazon S3.

○ Create an instance profile
Let Amazon EMR create a new instance profile so that you can specify a custom set of resources for it to access in Amazon S3.

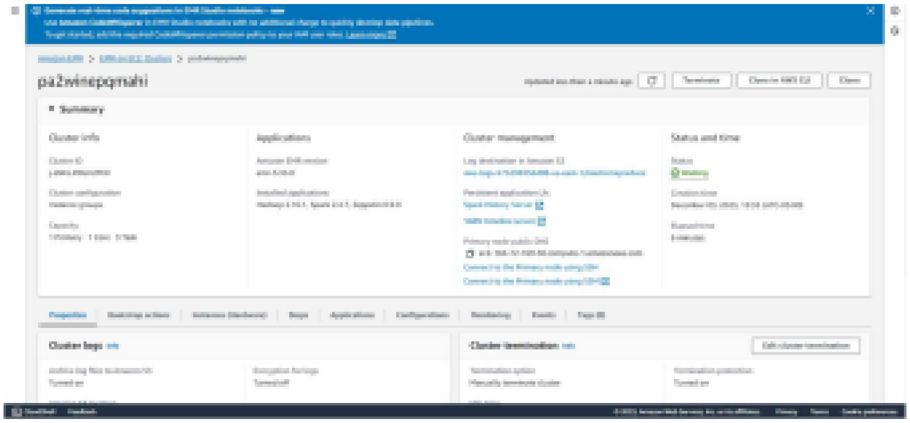
Instance profile: EMR_EC2_DefaultRole

Custom automatic scaling role - optional

When a custom automatic scaling rule triggers, Amazon EMR assumes this role to add and terminate EC2 instances. Learn more

Custom automatic scaling role: Choose IAM role

We can follow above steps for creating EMR cluster for the instances



pa2winepqmahi

Summary

Cluster info

Applications

Cluster management

Status and time

Status: Waiting

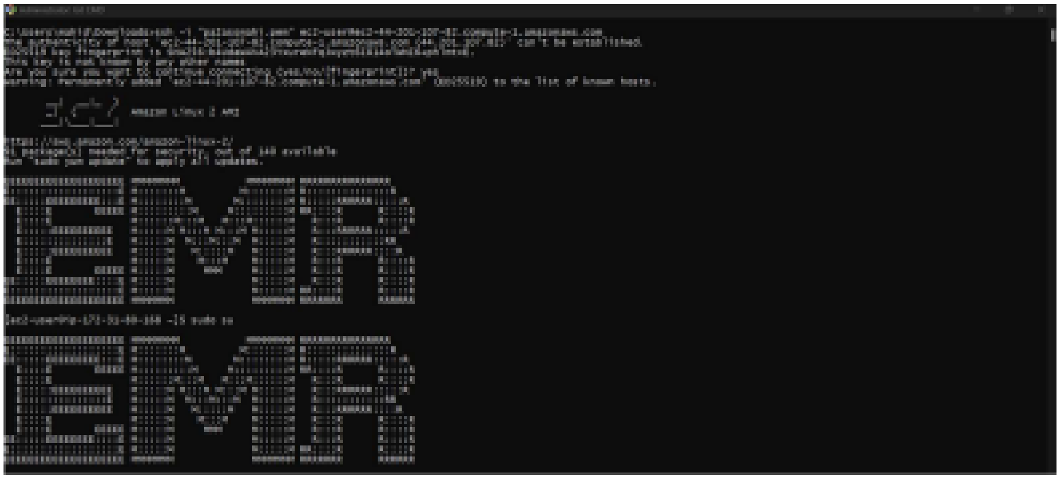
5.Now we are training ML model into spark cluster with ec2 instances in parallel:

1.Now the cluster will accept the tasks to run the ML model

Need to connect the Master instance in the Terminal:

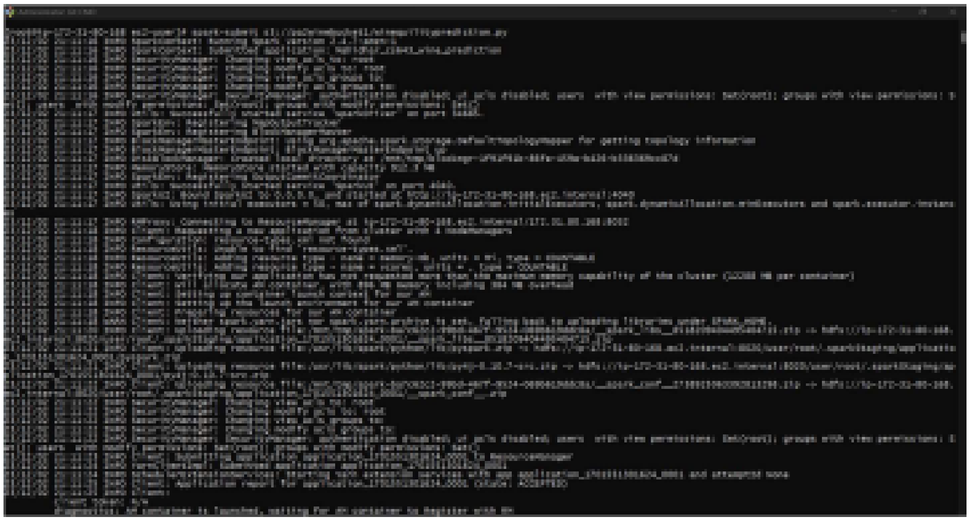
ssh -i "pa2assmahi.pem" ec2-user@ec2-44-201-107-82.compute-1.amazonaws.com and it is successfully login.

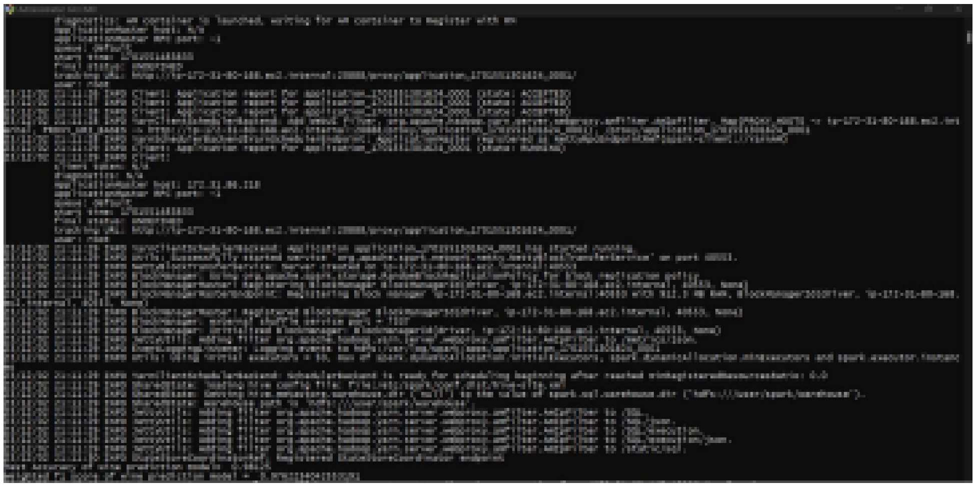
2.After the login of Master instance then change the root by using Sudo su



3. Submit the task by the command:

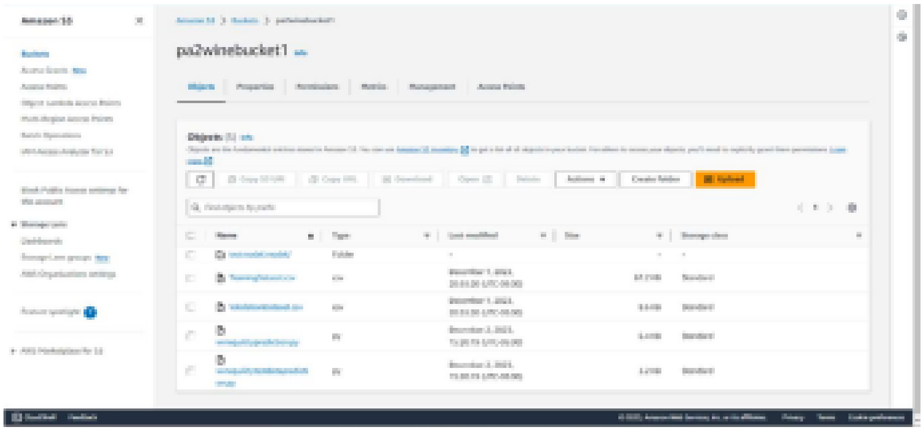
spark-submit s3://pa2winebucket1/ winequilityprediction.py





```
[illegible]
```

4.Then you can find the trace status for the above tasks, The status is succeed then there is a creation of test.model in the s3 bucket s3://pa2winebucket1



6.Now we are running ML model using the Docker:

1.Create an docker account and sign up.

2. After the successful login then download and setup the docker in your local system 3.Install the docker

4. Login the docker in the power shell by the command

docker login

Pwd

5.After login you need to build the image:

docker build -t winequlpred .

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

PUSH:

docker tag winequlpred dt37824/winequlpred

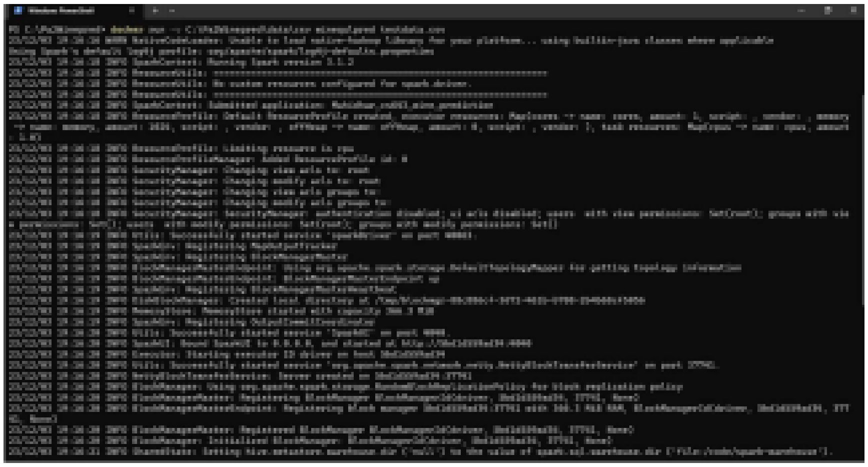
docker push dt37824/winequlpred

PULL:

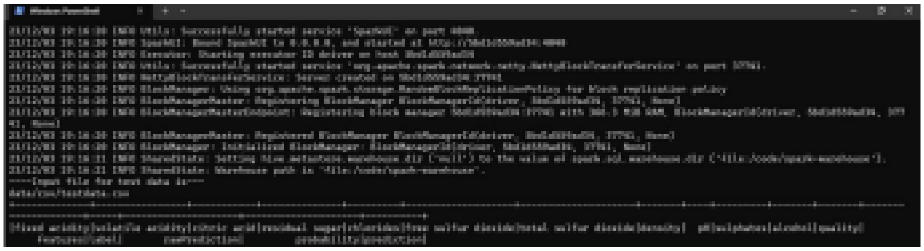
docker pull dt37824/winequlpred

7. Store your test data file in a designated folder, referred to as "dir." Mount this directory with the Docker container, and execute the container using the following command.

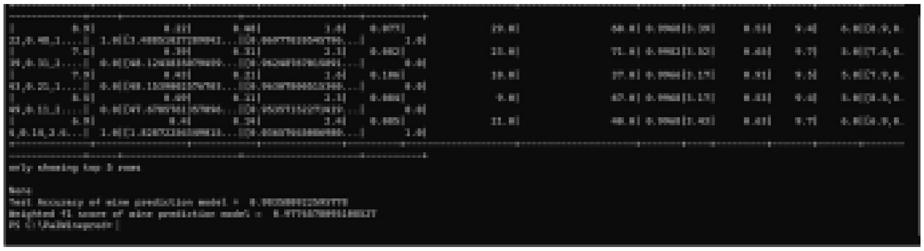
docker run -v C:\Pa2Winepred\data\csv winequlpred testdata.csv



```
PS C:\[illegible]> docker run -v C:\[illegible]\data\csv winequalpred testdata.csv
23/12/03 19:16:18 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
23/12/03 19:16:18 INFO SparkContext: Running Spark version 3.1.2
23/12/03 19:16:18 INFO ResourceUtils: ==============================================================
23/12/03 19:16:18 INFO ResourceUtils: No custom resources configured for spark.driver.
23/12/03 19:16:18 INFO ResourceUtils: ==============================================================
23/12/03 19:16:18 INFO SparkContext: Submitted application: [illegible]_cs643_wine_prediction
23/12/03 19:16:18 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 1, script: , vendor: , memory
 -> name: memory, amount: 1024, script: , vendor: , offHeap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpus -> name: cpus, amount
: 1.0)
23/12/03 19:16:18 INFO ResourceProfile: Limiting resource is cpu
23/12/03 19:16:18 INFO ResourceProfileManager: Added ResourceProfile id: 0
23/12/03 19:16:18 INFO SecurityManager: Changing view acls to: root
23/12/03 19:16:18 INFO SecurityManager: Changing modify acls to: root
23/12/03 19:16:18 INFO SecurityManager: Changing view acls groups to:
23/12/03 19:16:18 INFO SecurityManager: Changing modify acls groups to:
23/12/03 19:16:18 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users  with view permissions: Set(root); groups with vie
w permissions: Set(); users  with modify permissions: Set(root); groups with modify permissions: Set()
23/12/03 19:16:19 INFO Utils: Successfully started service 'sparkDriver' on port [illegible].
23/12/03 19:16:19 INFO SparkEnv: Registering MapOutputTracker
23/12/03 19:16:19 INFO SparkEnv: Registering BlockManagerMaster
23/12/03 19:16:19 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
23/12/03 19:16:19 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
23/12/03 19:16:19 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
23/12/03 19:16:19 INFO DiskBlockManager: Created local directory at /tmp/blockmgr-[illegible]
23/12/03 19:16:19 INFO MemoryStore: MemoryStore started with capacity 366.3 MiB
23/12/03 19:16:19 INFO SparkEnv: Registering OutputCommitCoordinator
23/12/03 19:16:20 INFO Utils: Successfully started service 'SparkUI' on port 4040.
23/12/03 19:16:20 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at http://[illegible]:4040
23/12/03 19:16:20 INFO Executor: Starting executor ID driver on host [illegible]
23/12/03 19:16:20 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 37761.
23/12/03 19:16:20 INFO NettyBlockTransferService: Server created on [illegible]:37761
23/12/03 19:16:20 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
23/12/03 19:16:20 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, [illegible], 37761, None)
23/12/03 19:16:20 INFO BlockManagerMasterEndpoint: Registering block manager [illegible]:37761 with 366.3 MiB RAM, BlockManagerId(driver, [illegible], 377
61, None)
23/12/03 19:16:20 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, [illegible], 37761, None)
23/12/03 19:16:20 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, [illegible], 37761, None)
23/12/03 19:16:21 INFO SharedState: Setting hive.metastore.warehouse.dir ('null') to the value of spark.sql.warehouse.dir ('file:/code/spark-warehouse').
```

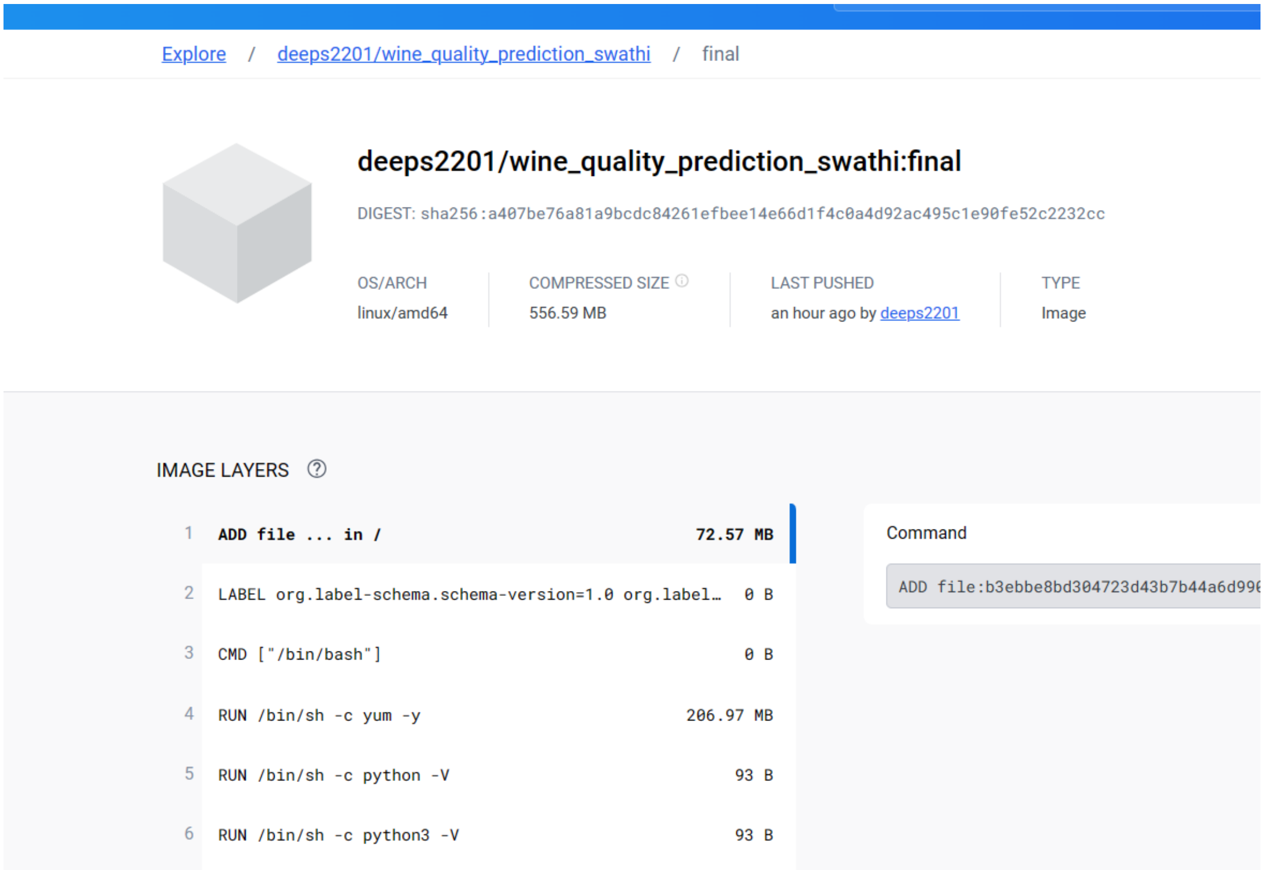


```
23/12/03 19:16:20 INFO Utils: Successfully started service 'SparkUI' on port 4040.
23/12/03 19:16:20 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at http://[illegible]:4040
23/12/03 19:16:20 INFO Executor: Starting executor ID driver on host [illegible]
23/12/03 19:16:20 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 37761.
23/12/03 19:16:20 INFO NettyBlockTransferService: Server created on [illegible]:37761
23/12/03 19:16:20 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
23/12/03 19:16:20 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, [illegible], 37761, None)
23/12/03 19:16:20 INFO BlockManagerMasterEndpoint: Registering block manager [illegible]:37761 with 366.3 MiB RAM, BlockManagerId(driver, [illegible], 377
61, None)
23/12/03 19:16:20 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, [illegible], 37761, None)
23/12/03 19:16:20 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, [illegible], 37761, None)
23/12/03 19:16:21 INFO SharedState: Setting hive.metastore.warehouse.dir ('null') to the value of spark.sql.warehouse.dir ('file:/code/spark-warehouse').
23/12/03 19:16:21 INFO SharedState: Warehouse path is 'file:/code/spark-warehouse'.
----Input file for test data is---
data/csv/testdata.csv
+-------------+----------------+-----------+--------------+---------+-------------------+--------------------+-------+----+---------+-------+-------+--------
------------+-----+--------------------+--------------------+----------+
|fixed acidity|volatile acidity|citric acid|residual sugar|chlorides|free sulfur dioxide|total sulfur dioxide|density|  pH|sulphates|alcohol|quality|
    features|label|       rawPrediction|         probability|prediction|
```



```
+-------------+----------------+-----------+--------------+---------+-------------------+--------------------+-------+----+---------+-------+-------+--------
------------+-----+--------------------+--------------------+----------+
|          8.9|            0.22|       0.48|           1.8|    0.077|               29.0|                60.0| 0.9968|3.39|     0.53|    9.4|    6.0|[8.9,0.
22,0.48,1....|  1.0|[1.4885[illegible]...|[0.0[illegible]...|       1.0|
|          7.6|            0.39|       0.31|           2.3|    0.082|               23.0|                71.0| 0.9982|3.52|     0.65|    9.7|    5.0|[7.6,0.
39,0.31,2....|  0.0|[40.1[illegible]...|[0.0[illegible]...|       0.0|
|          7.9|            0.43|       0.21|           1.6|    0.106|               10.0|                37.0| 0.9966|3.17|     0.91|    9.5|    5.0|[7.9,0.
43,0.21,1....|  0.0|[40.1[illegible]...|[0.0[illegible]...|       0.0|
|          8.5|            0.49|       0.11|           2.3|    0.084|                9.0|                67.0| 0.9968|3.17|     0.53|    9.4|    5.0|[8.5,0.
49,0.11,2....|  0.0|[47.[illegible]...|[0.0[illegible]...|       0.0|
|          6.9|             0.4|       0.14|           2.4|    0.085|               21.0|                40.0| 0.9968|3.43|     0.63|    9.7|    6.0|[6.9,0.
4,0.14,2.4...|  1.0|[1.[illegible]...|[0.0[illegible]...|       1.0|
+-------------+----------------+-----------+--------------+---------+-------------------+--------------------+-------+----+---------+-------+-------+--------
------------+-----+--------------------+--------------------+----------+
only showing top 5 rows

None
Test Accuracy of wine prediction model =  0.98[illegible]
Weighted f1 score of wine prediction model =  0.97[illegible]
PS C:\[illegible]>
```



Explore / deeps2201/wine_quality_prediction_swathi / final

deeps2201/wine_quality_prediction_swathi:final

DIGEST: sha256:a407be76a81a9bcdc84261efbee14e66d1f4c0a4d92ac495c1e90fe52c2232cc

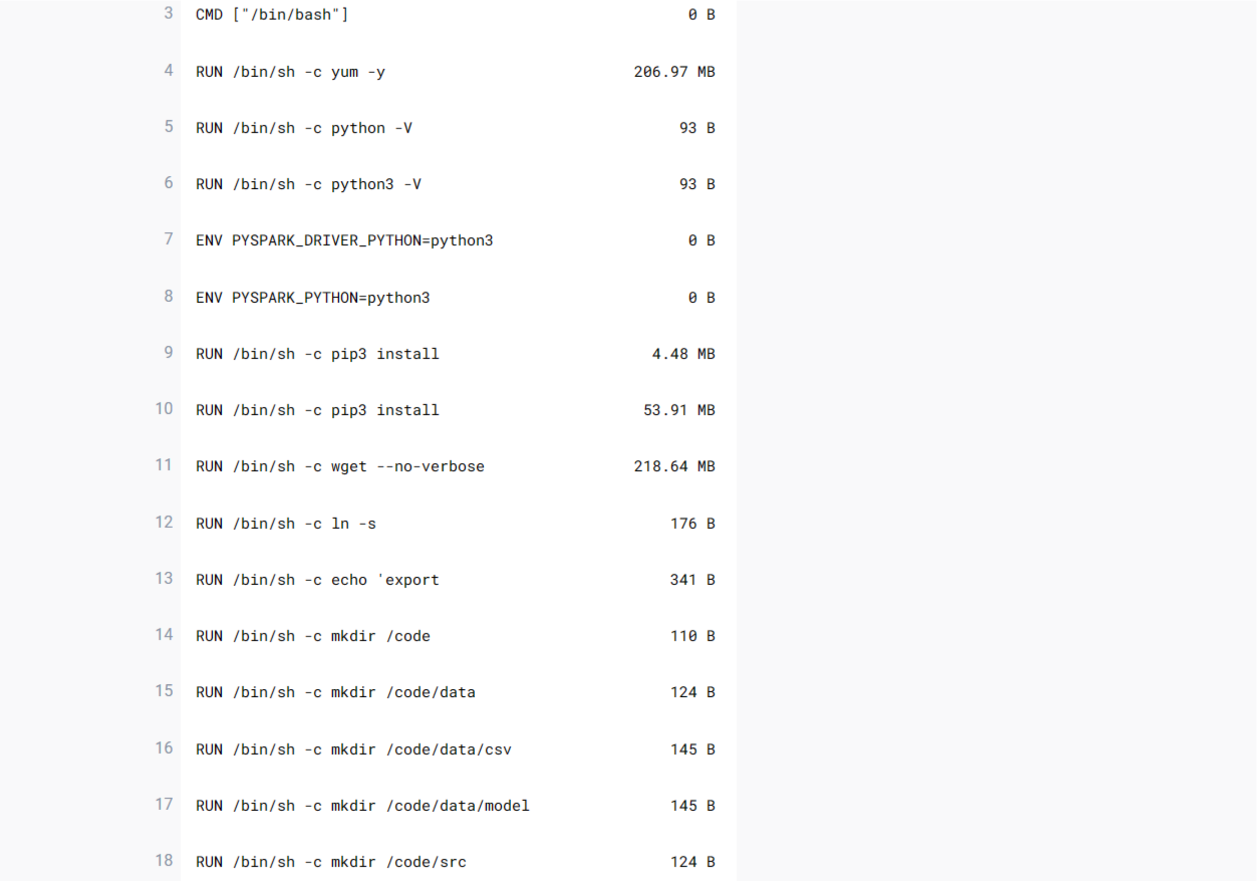
OS/ARCH	COMPRESSED SIZE ⓘ	LAST PUSHED	TYPE
linux/amd64	556.59 MB	an hour ago by deeps2201	Image

IMAGE LAYERS ⓘ

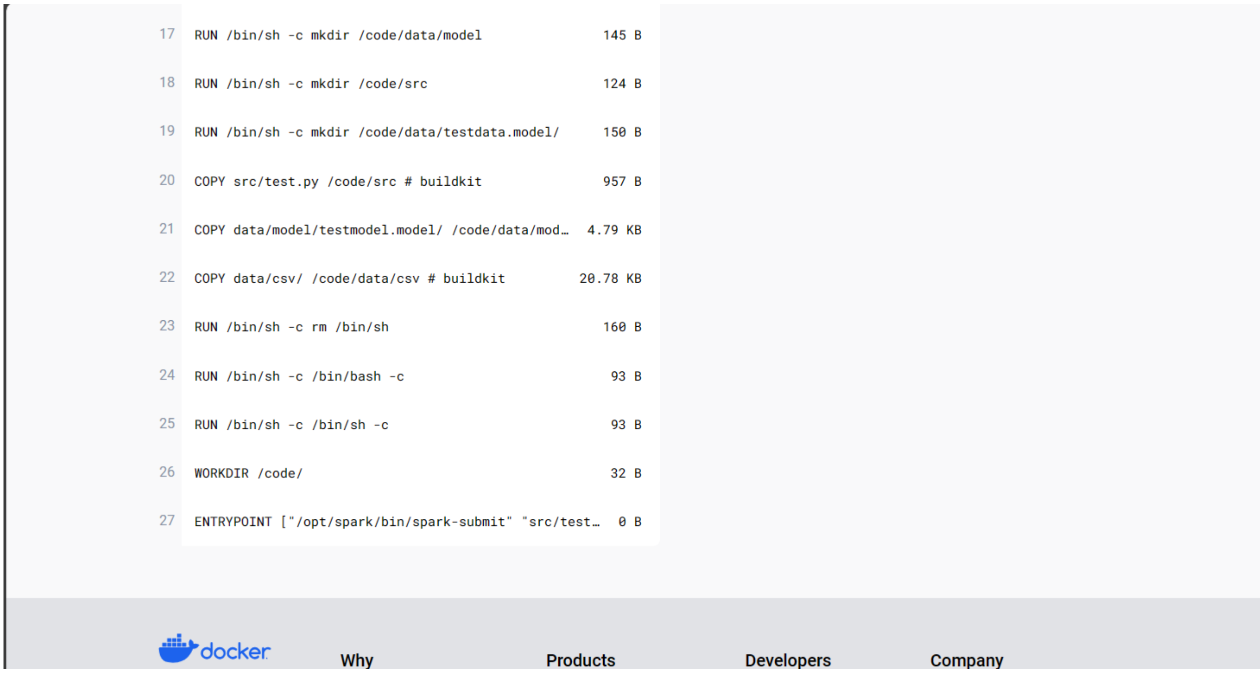
1	**ADD file ... in /**	**72.57 MB**
2	LABEL org.label-schema.schema-version=1.0 org.label…	0 B
3	CMD ["/bin/bash"]	0 B
4	RUN /bin/sh -c yum -y	206.97 MB
5	RUN /bin/sh -c python -V	93 B
6	RUN /bin/sh -c python3 -V	93 B

Command

```
ADD file:b3ebbe8bd304723d43b7b44a6d99
```



3	CMD ["/bin/bash"]	0 B
4	RUN /bin/sh -c yum -y	206.97 MB
5	RUN /bin/sh -c python -V	93 B
6	RUN /bin/sh -c python3 -V	93 B
7	ENV PYSPARK_DRIVER_PYTHON=python3	0 B
8	ENV PYSPARK_PYTHON=python3	0 B
9	RUN /bin/sh -c pip3 install	4.48 MB
10	RUN /bin/sh -c pip3 install	53.91 MB
11	RUN /bin/sh -c wget --no-verbose	218.64 MB
12	RUN /bin/sh -c ln -s	176 B
13	RUN /bin/sh -c echo 'export	341 B
14	RUN /bin/sh -c mkdir /code	110 B
15	RUN /bin/sh -c mkdir /code/data	124 B
16	RUN /bin/sh -c mkdir /code/data/csv	145 B
17	RUN /bin/sh -c mkdir /code/data/model	145 B
18	RUN /bin/sh -c mkdir /code/src	124 B



```
RUN /bin/sh -c mkdir /code/data/model                 145 B
RUN /bin/sh -c mkdir /code/src                        124 B
RUN /bin/sh -c mkdir /code/data/testdata.model/       150 B
COPY src/test.py /code/src # buildkit                 957 B
COPY data/model/testmodel.model/ /code/data/mod…   4.79 KB
COPY data/csv/ /code/data/csv # buildkit          20.78 KB
RUN /bin/sh -c rm /bin/sh                             160 B
RUN /bin/sh -c /bin/bash -c                            93 B
RUN /bin/sh -c /bin/sh -c                              93 B
WORKDIR /code/                                         32 B
ENTRYPOINT ["/opt/spark/bin/spark-submit" "src/test…    0 B
```

Conclusion: As shown in the image above, got an accuracy of ~98% while predicting the wine quality.