Wine Quality Prediction - AWS Spark Application

Objective:

The goal of this project is to create a Python application using **PySpark** for predicting the quality of wine. The model is trained and tested on an **AWS Elastic MapReduce (EMR)** cluster. Training is parallelized across multiple **EC2 instances**, and the model is deployed using a Docker container for easy scalability and deployment.

Links:

- **GitHub Repository**: https://github.com/rk94407/rohancc
- **Docker Hub Repository**: https://hub.docker.com/repository/docker/rohankatkam1698/testwinequalityprediction/general

---

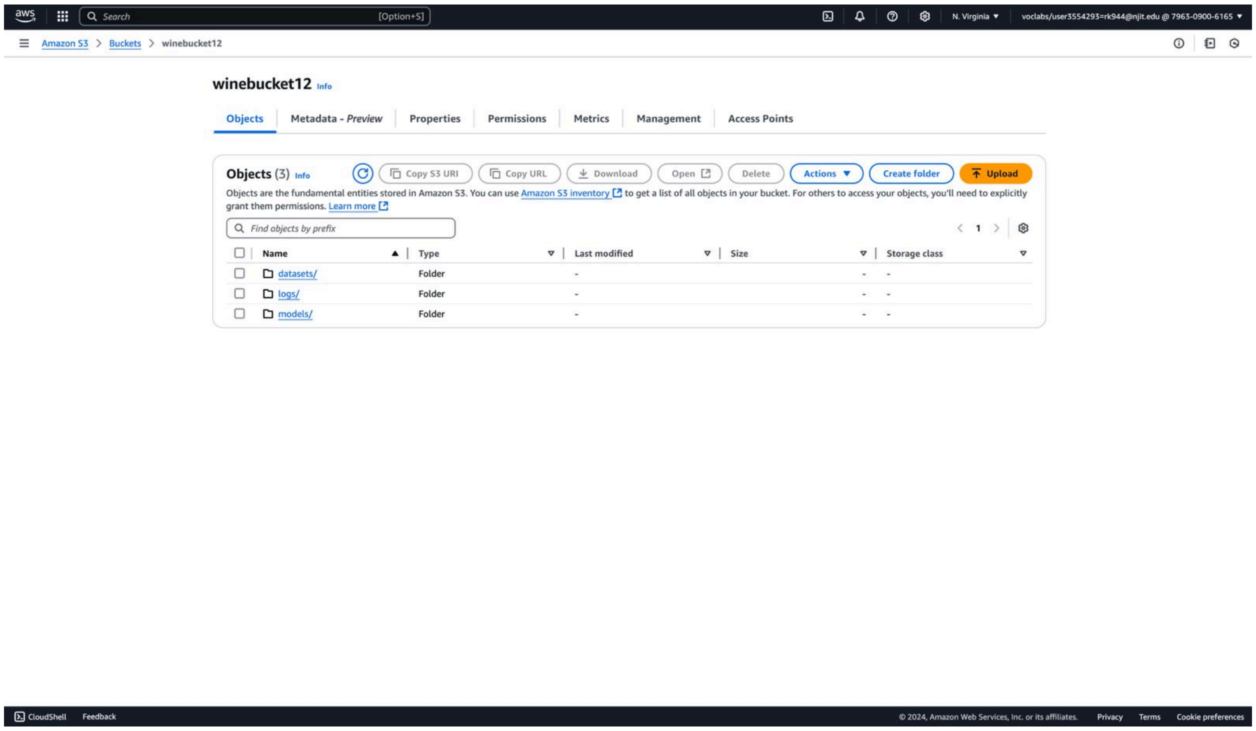
Steps for Execution

1. Create an EC2 Key Pair

- Go to **EC2 > Key Pairs** in your AWS console.
- Generate a new key pair named hemanth.pem and download it in .pem format for SSH access to the cluster.

2. Set Up an S3 Bucket

- Create an S3 bucket named winebucket12 to store your datasets and trained models.



winebucket12

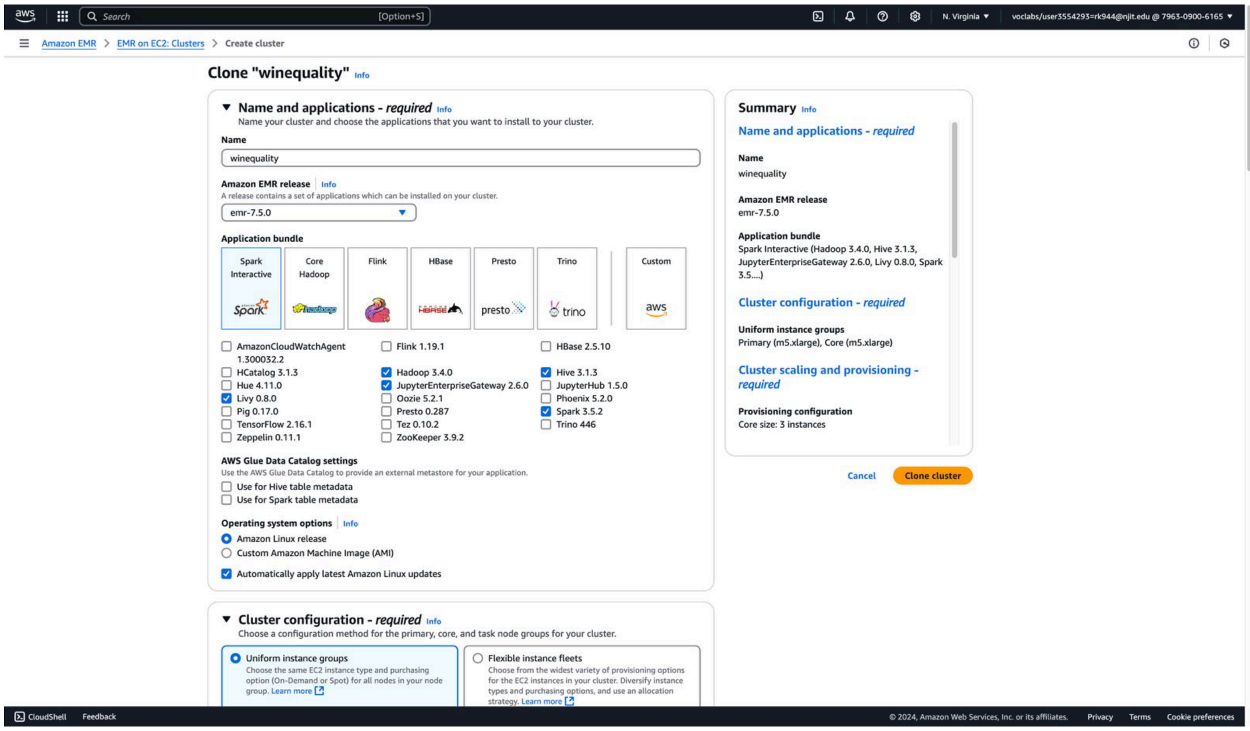
Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
datasets/	Folder	-	-	-
logs/	Folder	-	-	-
models/	Folder	-	-	-

3. Launch an EMR Cluster

- Navigate to the **EMR Console** and create a new EMR cluster with the following configurations:
  - **Cluster Name**: winequality
  - **EMR Release Version**: emr-7.5.0
  - **Applications**: Include **Hadoop 3.4.0** and **Spark 3.5.2**.


Amazon EMR > EMR on EC2: Clusters > Create cluster

Clone "winequality" Info

Name and applications - *required* Info
Name your cluster and choose the applications that you want to install to your cluster.

Name
winequality

Amazon EMR release Info
A release contains a set of applications which can be installed on your cluster.
emr-7.5.0

Application bundle
Spark Interactive | Core Hadoop | Flink | HBase | Presto | Trino | Custom

- [ ] AmazonCloudWatchAgent 1.300032.2
- [ ] HCatalog 3.1.3
- [ ] Hue 4.11.0
- [✓] Livy 0.8.0
- [ ] Pig 0.17.0
- [ ] TensorFlow 2.16.1
- [ ] Zeppelin 0.11.1
- [ ] Flink 1.19.1
- [✓] Hadoop 3.4.0
- [✓] JupyterEnterpriseGateway 2.6.0
- [ ] Oozie 5.2.1
- [ ] Presto 0.287
- [ ] Tez 0.10.2
- [ ] ZooKeeper 3.9.2
- [ ] HBase 2.5.10
- [✓] Hive 3.1.3
- [ ] JupyterHub 1.5.0
- [ ] Phoenix 5.2.0
- [✓] Spark 3.5.2
- [ ] Trino 446

AWS Glue Data Catalog settings
Use the AWS Glue Data Catalog to provide an external metastore for your application.
- [ ] Use for Hive table metadata
- [ ] Use for Spark table metadata

Operating system options Info
[✓ Amazon Linux release] Custom Amazon Machine Image (AMI)
- [✓] Automatically apply latest Amazon Linux updates

Cluster configuration - *required* Info
Choose a configuration method for the primary, core, and task node groups for your cluster.
[✓ Uniform instance groups] Choose the same EC2 instance type and purchasing option (On-Demand or Spot) for all nodes in your node group. Learn more
Flexible instance fleets: Choose from the widest variety of provisioning options for the EC2 instances in your cluster. Diversify instance types and purchasing options, and use an allocation strategy. Learn more

Summary Info
Name and applications - *required*
Name
winequality
Amazon EMR release
emr-7.5.0
Application bundle
Spark Interactive (Hadoop 3.4.0, Hive 3.1.3, JupyterEnterpriseGateway 2.6.0, Livy 0.8.0, Spark 3.5....)
Cluster configuration - *required*
Uniform instance groups
Primary (m5.xlarge), Core (m5.xlarge)
Cluster scaling and provisioning - *required*
Provisioning configuration
Core size: 3 instances

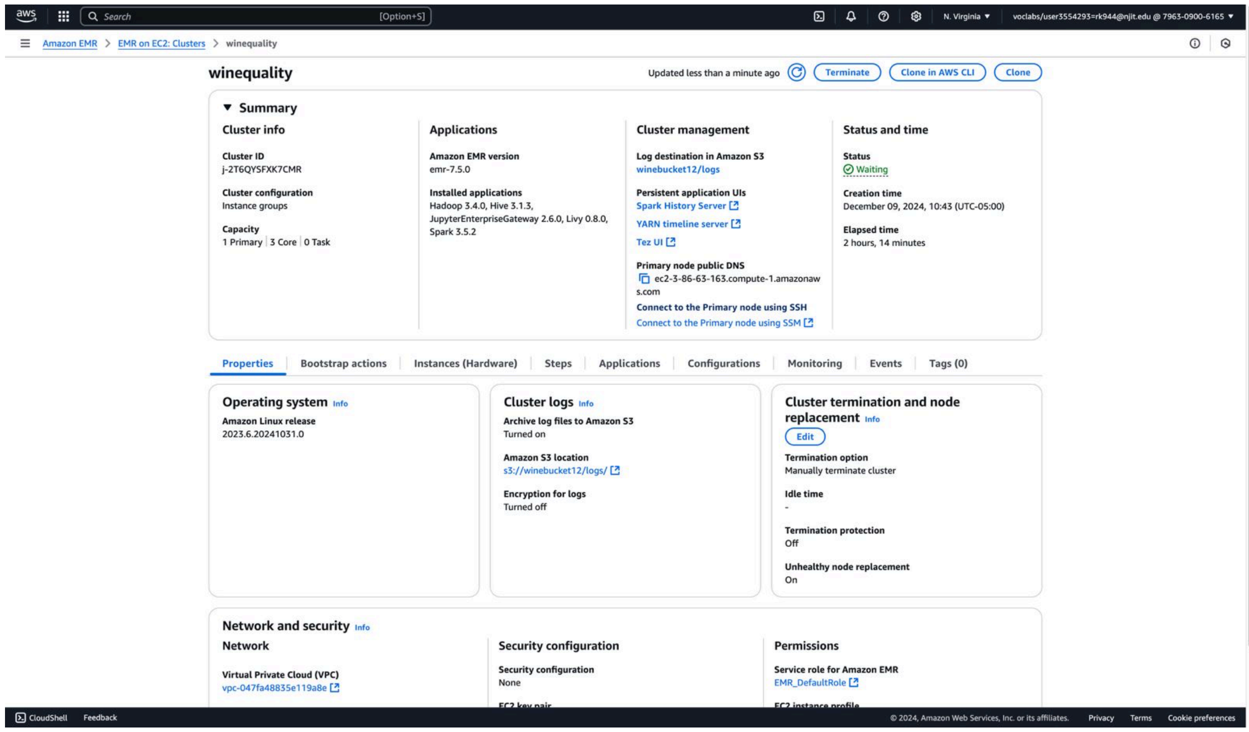
Cancel | Clone cluster

© 2024, Amazon Web Services, Inc. or its affiliates.

4. Configure the Spark Cluster

- Use an existing cluster configuration or create a new one.
- Configuration includes:
  - Cluster scaling and provisioning.
  - Networking settings and termination policies.

- Set up **IAM roles** and configure the EC2 key pair (rohan.pem) for security.


Amazon EMR > EMR on EC2: Clusters > winequality

winequality

Updated less than a minute ago

Summary

Cluster info

Cluster ID
j-2T6QYSFXK7CMR

Cluster configuration
Instance groups

Capacity
1 Primary | 3 Core | 0 Task

Applications

Amazon EMR version
emr-7.5.0

Installed applications
Hadoop 3.4.0, Hive 3.1.3, JupyterEnterpriseGateway 2.6.0, Livy 0.8.0, Spark 3.5.2

Cluster management

Log destination in Amazon S3
winebucket12/logs

Persistent application UIs
Spark History Server
YARN timeline server
Tez UI

Primary node public DNS
ec2-3-86-63-163.compute-1.amazonaws.com

Connect to the Primary node using SSH
Connect to the Primary node using SSM

Status and time

Status
Waiting

Creation time
December 09, 2024, 10:43 (UTC-05:00)

Elapsed time
2 hours, 14 minutes

Operating system

Amazon Linux release
2023.6.20241031.0

Cluster logs

Archive log files to Amazon S3
Turned on

Amazon S3 location
s3://winebucket12/logs/

Encryption for logs
Turned off

Cluster termination and node replacement

Termination option
Manually terminate cluster

Idle time
-

Termination protection
Off

Unhealthy node replacement
On

Network and security

Network

Virtual Private Cloud (VPC)
vpc-047fa48835e119a8e

Security configuration

Security configuration
None

Permissions

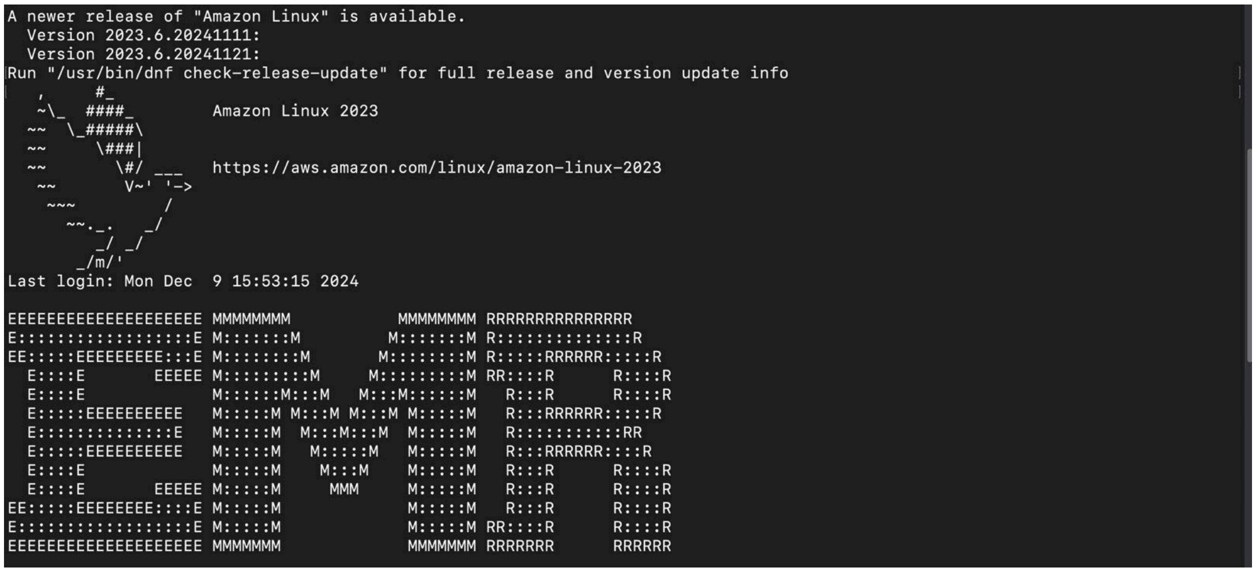
Service role for Amazon EMR
EMR_DefaultRole

5. Train the ML Model on EC2 Instances

Without Docker:

SSH into the **Master Node** of your EMR cluster:

```
ssh -i "rohan.pem" ec2-user@<ec2-public-dns>
```

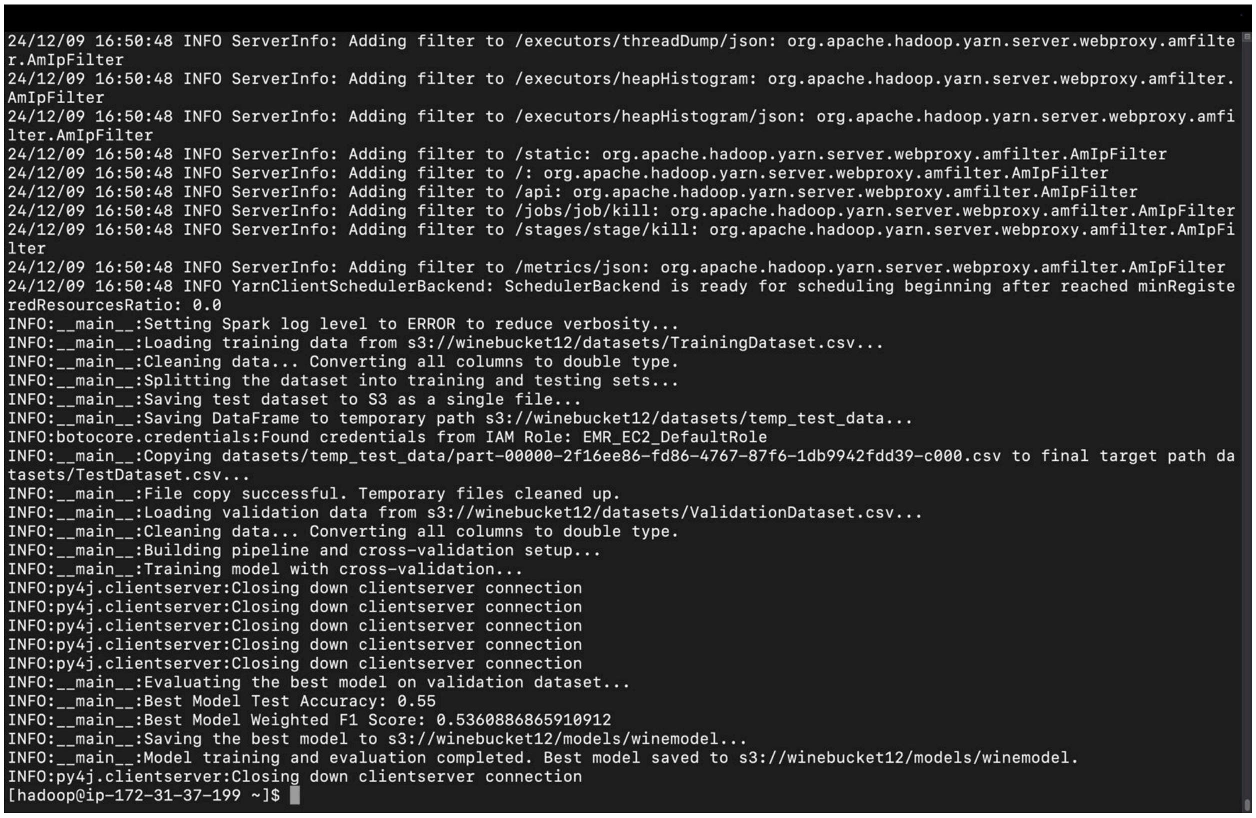


```
A newer release of "Amazon Linux" is available.
  Version 2023.6.20241111:
  Version 2023.6.20241121:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Mon Dec  9 15:53:15 2024

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR
```

Submit the training job using spark-submit:

spark-submit winequality.py s3://winebucket12/datasets/TrainingDataset.csv s3://winebucket12/datasets/ValidationDataset.csv



```
24/12/09 16:50:48 INFO ServerInfo: Adding filter to /executors/threadDump/json: org.apache.hadoop.yarn.server.webproxy.amfilte
r.AmIpFilter
24/12/09 16:50:48 INFO ServerInfo: Adding filter to /executors/heapHistogram: org.apache.hadoop.yarn.server.webproxy.amfilter.
AmIpFilter
24/12/09 16:50:48 INFO ServerInfo: Adding filter to /executors/heapHistogram/json: org.apache.hadoop.yarn.server.webproxy.amfi
lter.AmIpFilter
24/12/09 16:50:48 INFO ServerInfo: Adding filter to /static: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/12/09 16:50:48 INFO ServerInfo: Adding filter to /: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/12/09 16:50:48 INFO ServerInfo: Adding filter to /api: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/12/09 16:50:48 INFO ServerInfo: Adding filter to /jobs/job/kill: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/12/09 16:50:48 INFO ServerInfo: Adding filter to /stages/stage/kill: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFi
lter
24/12/09 16:50:48 INFO ServerInfo: Adding filter to /metrics/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/12/09 16:50:48 INFO YarnClientSchedulerBackend: SchedulerBackend is ready for scheduling beginning after reached minRegiste
redResourcesRatio: 0.0
INFO:__main__:Setting Spark log level to ERROR to reduce verbosity...
INFO:__main__:Loading training data from s3://winebucket12/datasets/TrainingDataset.csv...
INFO:__main__:Cleaning data... Converting all columns to double type.
INFO:__main__:Splitting the dataset into training and testing sets...
INFO:__main__:Saving test dataset to S3 as a single file...
INFO:__main__:Saving DataFrame to temporary path s3://winebucket12/datasets/temp_test_data...
INFO:botocore.credentials:Found credentials from IAM Role: EMR_EC2_DefaultRole
INFO:__main__:Copying datasets/temp_test_data/part-00000-2f16ee86-fd86-4767-87f6-1db9942fdd39-c000.csv to final target path da
tasets/TestDataset.csv...
INFO:__main__:File copy successful. Temporary files cleaned up.
INFO:__main__:Loading validation data from s3://winebucket12/datasets/ValidationDataset.csv...
INFO:__main__:Cleaning data... Converting all columns to double type.
INFO:__main__:Building pipeline and cross-validation setup...
INFO:__main__:Training model with cross-validation...
INFO:py4j.clientserver:Closing down clientserver connection
INFO:py4j.clientserver:Closing down clientserver connection
INFO:py4j.clientserver:Closing down clientserver connection
INFO:py4j.clientserver:Closing down clientserver connection
INFO:py4j.clientserver:Closing down clientserver connection
INFO:__main__:Evaluating the best model on validation dataset...
INFO:__main__:Best Model Test Accuracy: 0.55
INFO:__main__:Best Model Weighted F1 Score: 0.5360886865910912
INFO:__main__:Saving the best model to s3://winebucket12/models/winemodel...
INFO:__main__:Model training and evaluation completed. Best model saved to s3://winebucket12/models/winemodel.
INFO:py4j.clientserver:Closing down clientserver connection
[hadoop@ip-172-31-37-199 ~]$ 
```

- This script splits the **TrainingDataset.csv** into 90% for training and 10% for testing. The test data is saved as **TestDataset.csv** in the S3 dataset folder.

6. Save the Trained Model

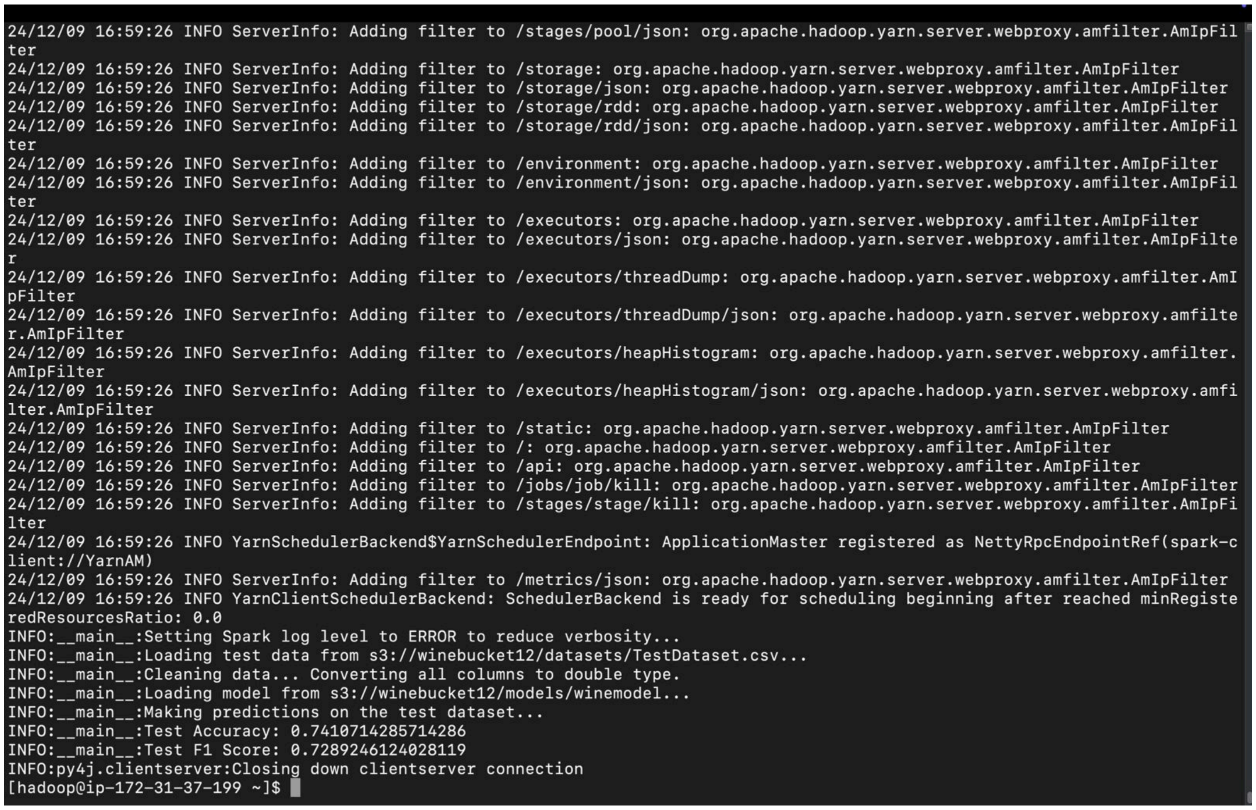
After training, the model is saved in the following S3 location:

s3://winebucket12/models/winemodel

7. Test the Model

To test the trained model, submit the test job with the following command:

spark-submit winequality.py s3://winebucket12/datasets/TestDataset.csv s3://winebucket12/models/winemodel



```
24/12/09 16:59:26 INFO ServerInfo: Adding filter to /stages/pool/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFil
ter
24/12/09 16:59:26 INFO ServerInfo: Adding filter to /storage: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/12/09 16:59:26 INFO ServerInfo: Adding filter to /storage/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/12/09 16:59:26 INFO ServerInfo: Adding filter to /storage/rdd: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/12/09 16:59:26 INFO ServerInfo: Adding filter to /storage/rdd/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFil
ter
24/12/09 16:59:26 INFO ServerInfo: Adding filter to /environment: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/12/09 16:59:26 INFO ServerInfo: Adding filter to /environment/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFil
ter
24/12/09 16:59:26 INFO ServerInfo: Adding filter to /executors: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/12/09 16:59:26 INFO ServerInfo: Adding filter to /executors/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilte
r
24/12/09 16:59:26 INFO ServerInfo: Adding filter to /executors/threadDump: org.apache.hadoop.yarn.server.webproxy.amfilter.AmI
pFilter
24/12/09 16:59:26 INFO ServerInfo: Adding filter to /executors/threadDump/json: org.apache.hadoop.yarn.server.webproxy.amfilte
r.AmIpFilter
24/12/09 16:59:26 INFO ServerInfo: Adding filter to /executors/heapHistogram: org.apache.hadoop.yarn.server.webproxy.amfilter.
AmIpFilter
24/12/09 16:59:26 INFO ServerInfo: Adding filter to /executors/heapHistogram/json: org.apache.hadoop.yarn.server.webproxy.amfi
lter.AmIpFilter
24/12/09 16:59:26 INFO ServerInfo: Adding filter to /static: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/12/09 16:59:26 INFO ServerInfo: Adding filter to /: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/12/09 16:59:26 INFO ServerInfo: Adding filter to /api: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/12/09 16:59:26 INFO ServerInfo: Adding filter to /jobs/job/kill: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/12/09 16:59:26 INFO ServerInfo: Adding filter to /stages/stage/kill: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFi
lter
24/12/09 16:59:26 INFO YarnSchedulerBackend$YarnSchedulerEndpoint: ApplicationMaster registered as NettyRpcEndpointRef(spark-c
lient://YarnAM)
24/12/09 16:59:26 INFO ServerInfo: Adding filter to /metrics/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/12/09 16:59:26 INFO YarnClientSchedulerBackend: SchedulerBackend is ready for scheduling beginning after reached minRegiste
redResourcesRatio: 0.0
INFO:__main__:Setting Spark log level to ERROR to reduce verbosity...
INFO:__main__:Loading test data from s3://winebucket12/datasets/TestDataset.csv...
INFO:__main__:Cleaning data... Converting all columns to double type.
INFO:__main__:Loading model from s3://winebucket12/models/winemodel...
INFO:__main__:Making predictions on the test dataset...
INFO:__main__:Test Accuracy: 0.7410714285714286
INFO:__main__:Test F1 Score: 0.7289246124028119
INFO:py4j.clientserver:Closing down clientserver connection
[hadoop@ip-172-31-37-199 ~]$ 
```

Docker Setup

1. Install Docker

- Create a **Docker** account if you don't already have one.
- Install **Docker** on your local machine.

2. Build the Docker Image

Once your Docker environment is set up, build the Docker image for the application:

```
docker build -t testwinequalityprediction .
```

3. Push and Pull the Docker Image

Tag the Docker image:

```
docker tag testwinequalityprediction rohankatkam1698/testwinequalityprediction
```

Push the image to Docker Hub:

```
docker push rohankatkam1698/testwinequalityprediction
```

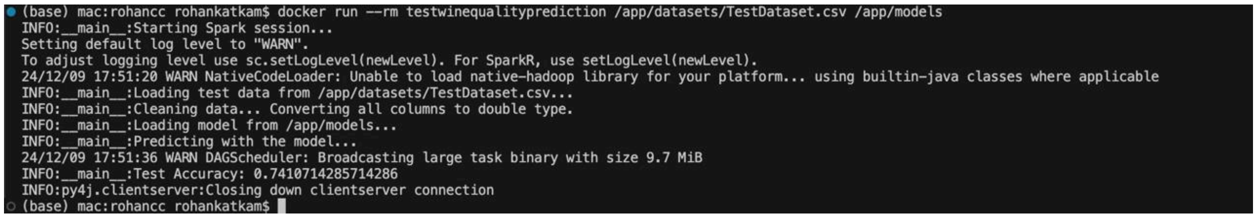
Pull the image from Docker Hub:

```
docker pull rohankatkam1698/testwinequalityprediction
```

4. Run the Docker Container

After pulling the image, run the container to make predictions using the trained model:

```
docker run --rm testwinequalityprediction /app/datasets/TestDataset.csv
/app/models
```



```
(base) mac:rohancc rohankatkam$ docker run --rm testwinequalityprediction /app/datasets/TestDataset.csv /app/models
INFO:__main__:Starting Spark session...
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/12/09 17:51:20 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
INFO:__main__:Loading test data from /app/datasets/TestDataset.csv...
INFO:__main__:Cleaning data... Converting all columns to double type.
INFO:__main__:Loading model from /app/models...
INFO:__main__:Predicting with the model...
24/12/09 17:51:36 WARN DAGScheduler: Broadcasting large task binary with size 9.7 MiB
INFO:__main__:Test Accuracy: 0.7410714285714286
INFO:py4j.clientserver:Closing down clientserver connection
(base) mac:rohancc rohankatkam$
```

Model Accuracy

The **wine quality prediction model** achieved an accuracy of **0.74** based on the test dataset.