

CS 643-861 CLOUD COMPUTING

PROGRAMMING ASSIGNMENT 2

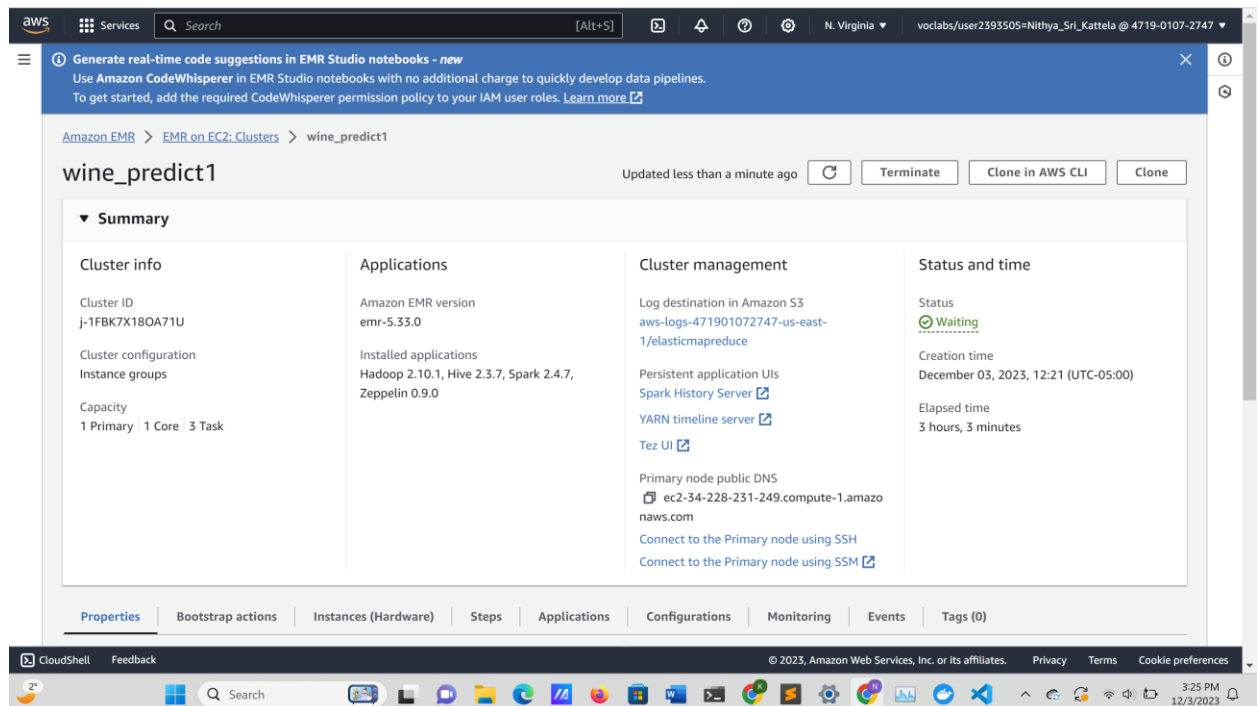
Github Link: https://github.com/nithya2503/cc643861_pa2

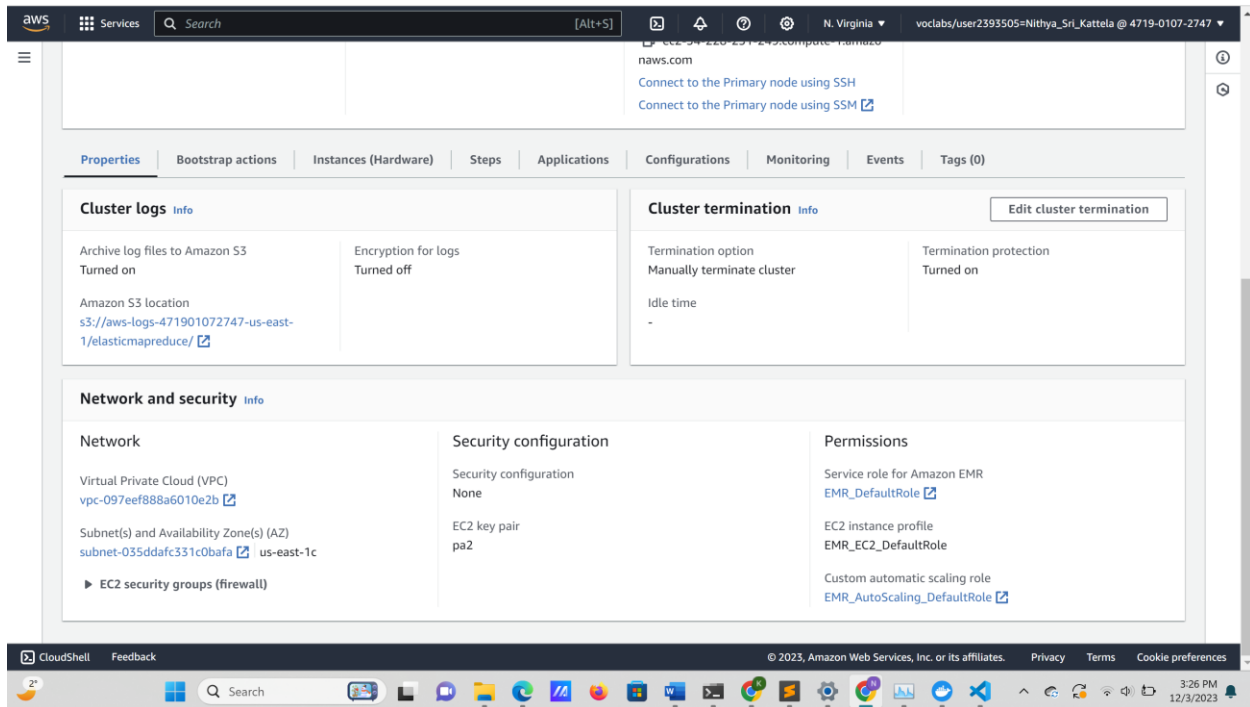
Docker Hub Link: <https://hub.docker.com/repository/docker/nithya252423/nk659wineapp/>

STEPS TO FOLLOW

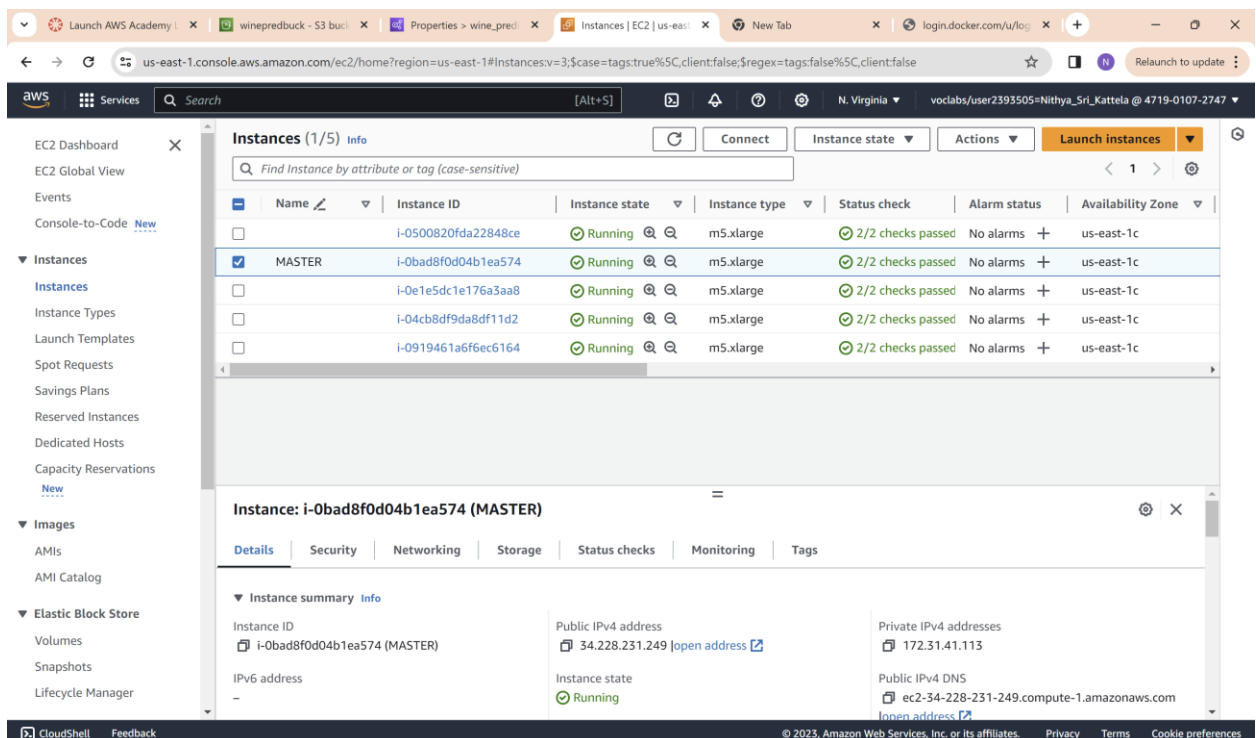
1. To create a spark cluster in AWS

- Login to your AWS account and navigate to EMR console
- Create a key pair by navigating to EC2 – Network and security – key pairs, choose .pem format while downloading the key pair, mark the location of it to use it in future.
- Go to the EMR console and create cluster, give a name to the cluster, choose emr-5.33.0
- Choosing software configurations as Hadoop 2.10.1, Hive 2.3.7, Spark 2.4.7, Zeppelin 0.9.0 in the check box
- Leave the core as default, m5.xlarge, for the task choose 3 tasks and 1 as core
- Leave the rest of all the configurations as default.
- Successful creation of cluster shows waiting as its status.





- All the instances will be running in the EC2.



2. Training the Machine Learning Model with 4 EC2 instances parallelly using Spark cluster.

- Connect Master with SSH using CMD or powershell using `ssh -I "key" user@Public IPV4 DNS`

- Change the user to root user using command `sudo su`

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\katte> cd downloads
PS C:\Users\katte\downloads> ssh -i "pa2.pem" ec2-user@ec2-34-228-231-249.compute-1.amazonaws.com
Last login: Sun Dec  3 17:34:23 2023 from pool-100-35-96-91.nwrknj.fios.verizon.net

  _ _ _ _ _
 _ | ( | /   Amazon Linux 2 AMI
--| \___|___|

https://aws.amazon.com/amazon-linux-2/
88 package(s) needed for security, out of 137 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRRRRRRRRRRR
E:.....E M:.....M M:.....M R:.....R
EE:....EEEEEEEEEE: E M:.....M M:.....M R:....RRRRR:..:R
E:..:E EEEEE M:.....M M:.....M RR:..:R R:..:R
E:..:E M:.....M M:.....M R:..:R R:..:R
E:..:EEEEEEEEEE M:.....M M:..M M:..M M:..M R:..RRRRR:..:R
E:.....E M:.....M M:..M M:..M M:..M R:.....RR
E:..:EEEEEEEEEE M:.....M M:..M M:..M R:..RRRRR:..: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 MMMMMMMM MMMMMMMM RRRRRRRRRRRRRR

[ec2-user@ip-172-31-41-113 ~]$ sudo su

EEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRRRRRRRRRRR
E:.....E M:.....M M:.....M R:.....R
EE:....EEEEEEEEEE: E M:.....M M:.....M R:....RRRRR:..:R
E:..:E EEEEE M:.....M M:.....M RR:..:R R:..:R
E:..:E M:.....M M:.....M R:..:R R:..:R
E:..:EEEEEEEEEE M:.....M M:..M M:..M M:..M R:..RRRRR:..:R
E:.....E M:.....M M:..M M:..M M:..M R:.....RR
E:..:EEEEEEEEEE M:.....M M:..M M:..M R:..RRRRR:..:R
E:..:EEEEEEEEEE M:.....M M:..M M:..M R:..RRRRR:..:R
```

- To submit the job - `spark-submit s3://winepredbuck/wine_qual_pred.py`

```
EEEEEEEEEEEEEEEEEEEE MMMMMM RRRRRR RRRRRR

[root@ip-172-31-41-113 ec2-user]# spark-submit s3://winepredbuck/wine_qual_pred.py
23/12/03 18:35:20 INFO SparkContext: Running Spark version 2.4.7-amzn-1
23/12/03 18:35:20 INFO SparkContext: Submitted application: nithya_cs643_wine_prediction
23/12/03 18:35:20 INFO SecurityManager: Changing view acls to: root
23/12/03 18:35:20 INFO SecurityManager: Changing modify acls to: root
23/12/03 18:35:20 INFO SecurityManager: Changing view acls groups to:
23/12/03 18:35:20 INFO SecurityManager: Changing modify acls groups to:
23/12/03 18:35:20 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(root); groups
with view permissions: Set(); users with modify permissions: Set(root); groups with modify permissions: Set()
23/12/03 18:35:21 INFO Utils: Successfully started service 'sparkDriver' on port 32903.
23/12/03 18:35:21 INFO SparkEnv: Registering MapOutputTracker
23/12/03 18:35:21 INFO SparkEnv: Registering BlockManagerMaster
23/12/03 18:35:21 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
23/12/03 18:35:21 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
23/12/03 18:35:21 INFO DiskBlockManager: Created local directory at /mnt/tmp/blockmgr-16c3694f-3046-4781-b79b-3a1a256c1e82
23/12/03 18:35:21 INFO MemoryStore: MemoryStore started with capacity 912.3 MB
23/12/03 18:35:21 INFO SparkEnv: Registering OutputCommitCoordinator
23/12/03 18:35:21 INFO Utils: Successfully started service 'SparkUI' on port 4040.
23/12/03 18:35:21 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at http://ip-172-31-41-113.ec2.internal:4040
23/12/03 18:35:21 INFO Utils: Using initial executors = 50, max of spark.dynamicAllocation.initialExecutors, spark.dynamicAllocation.minExecutors an
d spark.executor.instances
23/12/03 18:35:21 INFO RMProxy: Connecting to ResourceManager at ip-172-31-41-113.ec2.internal/172.31.41.113:8032
23/12/03 18:35:22 INFO Client: Requesting a new application from cluster with 4 NodeManagers
23/12/03 18:35:22 INFO Configuration: resource-types.xml not found
23/12/03 18:35:22 INFO ResourceUtils: Unable to find 'resource-types.xml'.
23/12/03 18:35:22 INFO ResourceUtils: Adding resource type - name = memory-mb, units = Mi, type = COUNTABLE
23/12/03 18:35:22 INFO ResourceUtils: Adding resource type - name = vcores, units = , type = COUNTABLE
23/12/03 18:35:22 INFO Client: Verifying our application has not requested more than the maximum memory capability of the cluster (12288 MB per cont
ainer)
23/12/03 18:35:22 INFO Client: Will allocate AM container, with 896 MB memory including 384 MB overhead
23/12/03 18:35:22 INFO Client: Setting up container launch context for our AM
23/12/03 18:35:22 INFO Client: Setting up the launch environment for our AM container
23/12/03 18:35:22 INFO Client: Preparing resources for our AM container
23/12/03 18:35:22 WARN Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading libraries under SPARK_HOME.
23/12/03 18:35:24 INFO Client: Uploading resource file:/mnt/tmp/spark-ddc4806d-1232-4dff-88c7-aa052cec43bc/_spark_libs_7635353359947967555.zip ->
hdfs://ip-172-31-41-113.ec2.internal:8020/user/root/.sparkStaging/application.1701624464685_0002/_spark_libs_7635353359947967555.zip
23/12/03 18:35:25 INFO Client: Uploading resource file:/etc/spark/conf/hive-site.xml -> hdfs://ip-172-31-41-113.ec2.internal:8020/user/root/.sparkSt
aging/application.1701624464685_0002/hive-site.xml
23/12/03 18:35:25 INFO Client: Uploading resource file:/usr/lib/spark/python/lib/pyspark.zip -> hdfs://ip-172-31-41-113.ec2.internal:8020/user/root/
```

- The status of the job can be traced out in EMR UI application logs, the model will appear in the s3 bucket. `arn:aws:s3:::winepredbuck`

3. Running Machine Learning Model using Docker

- Install Docker in your machine according to the operating system of your machine
- Create a Docker file with the set of instructions and build an image of it – `docker build -t imagename (docker build -t nk659wineapp .)`
- Tag the image `docker tag imagename username/imagename (docker tag nk659wineapp nithya252423/nk659wineapp)`
- Push the image to the docker hub: `docker push username/imagename (docker push nithya252423/nk659wineapp)`
- Pull the image to your machine by `docker pull username/imagename`
- Trace the path of the test data, where the docker container mounts it.
- `Docker run -v /path to the testdata/: nk659wineapp testdata.csv`

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\katie> cd\
PS C:\> cd pa2-cc
PS C:\pa2-cc> docker login
Authenticating with existing credentials...
Login Succeeded
PS C:\pa2-cc> docker build -t nk659wineapp .
[+] Building 2.7s (29/29) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile                0.0s
=> == transferring dockerfile: 1.32kB                             0.0s
=> [internal] load .dockerignore                                   0.0s
=> == transferring context: 2B                                     0.0s
=> [internal] load metadata for docker.io/library/centos:7        0.5s
=> [auth] library/centos:pull token for registry-1.docker.io     0.0s
=> [ 1/23] FROM docker.io/library/centos:7@sha256:be65f488b7764a 0.0s
=> [internal] load build context                                  0.0s
=> == transferring context: 7.68kB                                 0.0s
=> CACHED [ 2/23] RUN yum -y update && yum -y install python3 py 0.0s
=> CACHED [ 3/23] RUN python -V                                    0.0s
=> CACHED [ 4/23] RUN python3 -V                                   0.0s
=> CACHED [ 5/23] RUN pip3 install --upgrade pip                  0.0s
=> CACHED [ 6/23] RUN pip3 install numpy panda                    0.0s
=> CACHED [ 7/23] RUN pip3 install pandas                         0.0s
=> CACHED [ 8/23] RUN wget --no-verbose -O apache-spark.tgz "htt 0.0s
=> CACHED [ 9/23] RUN ln -s /opt/spark-3.1.2-bin-hadoop2.7 /opt/ 0.0s
=> CACHED [10/23] RUN (echo 'export SPARK_HOME=/opt/spark' >> ~/. 0.0s
=> CACHED [11/23] RUN mkdir /code                                 0.0s
=> CACHED [12/23] RUN mkdir /code/data                           0.0s
=> CACHED [13/23] RUN mkdir /code/data/csv                       0.0s
=> CACHED [14/23] RUN mkdir /code/data/model                     0.0s
=> CACHED [15/23] RUN mkdir /code/src                             0.0s
=> CACHED [16/23] RUN mkdir /code/data/testdata.model/           0.0s
=> [17/23] COPY wine_qual_test_pred.py /code/src                 0.0s
=> [18/23] COPY testdata.model/ /code/data/model/testdata.model 0.1s
=> [19/23] COPY data/csv/ /code/data/csv                         0.1s
=> [20/23] RUN rm /bin/sh && ln -s /bin/bash /bin/sh              0.4s
=> [21/23] RUN /bin/bash -c "source ~/.bashrc"                    0.5s
=> [22/23] RUN /bin/sh -c "source ~/.bashrc"                      0.6s
=> [23/23] WORKDIR /code/                                         0.1s
=> exporting to image                                             0.2s
=> == exporting layers                                             0.2s
=> == writing image sha256:4e9ac2b7b4fa848849939fe31eefac545c207 0.0s
=> == naming to docker.io/library/nk659wineapp                    0.0s

```

```

What's Next?
View a summary of image vulnerabilities and recommendations + docker scout quickview
PS C:\pa2-cc> docker login -u nithya252423
Password:
Login Succeeded
PS C:\pa2-cc> docker tag nk659wineapp nithya252423/nk659wineapp
PS C:\pa2-cc> docker push nithya252423/nk659wineapp
Using default tag: latest
The push refers to repository [docker.io/nithya252423/nk659wineapp]
5f70bf18a086: Pushed
c8bab6540d3e: Pushed
b5fefcc0c04f: Pushed
a6163de5967d: Pushed
49238833b509: Pushed
0c86b4c3b25c: Pushed
dc2ec13ada4d: Pushed
89d9a6320459: Pushed
b52f4fea0d5b: Pushed
6d0015a8eed4: Pushed
8717c9eb7d69: Pushed
ca811b885aa1: Pushed
17932e288033: Pushed
dbc00f2ac01: Pushed
9b6a9c050195: Pushed
061d18e47511: Pushed
ca63f057815d: Pushed
b5d27e9ba5b7: Pushed
c04bfc2b2c0d: Pushed
c285adad3fcb: Pushed
32ef37054d91: Pushed
174f56854903: Pushed
latest: digest: sha256:900b318fc59010ab70fe9a60ad8715136a86b5be780b116398913df5003ba5ab size: 5109
PS C:\pa2-cc> docker pull nithya252423/nk659wineapp
Using default tag: latest
latest: Pulling from nithya252423/nk659wineapp
Digest: sha256:900b318fc59010ab70fe9a60ad8715136a86b5be780b116398913df5003ba5ab
Status: Image is up to date for nithya252423/nk659wineapp:latest
docker.io/nithya252423/nk659wineapp:latest

What's Next?
View a summary of image vulnerabilities and recommendations + docker scout quickview nithya252423/nk659wineapp
PS C:\pa2-cc> docker run -v /data/csv nk659wineapp testdata.csv
23/12/03 18:28:48 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 18:28:49 INFO SparkContext: Running Spark version 3.1.2
23/12/03 18:28:49 INFO ResourceUtils: =====
23/12/03 18:28:49 INFO ResourceUtils: No custom resources configured for spark.driver.

```

```
Windows PowerShell X Windows PowerShell X root@ip-172-31-4 X Windows PowerShell X Windows PowerShell X root@ip-172-31-4 X Windows PowerShell X + -
23/12/03 18:28:50 INFO Utils: Successfully started service 'SparkUI' on port 4040.
23/12/03 18:28:50 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at http://094c129fca8a:4040
23/12/03 18:28:51 INFO Executor: Starting executor ID driver on host 094c129fca8a
23/12/03 18:28:51 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 41745.
23/12/03 18:28:51 INFO NettyBlockTransferService: Server created on 094c129fca8a:41745
23/12/03 18:28:51 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
23/12/03 18:28:51 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, 094c129fca8a, 41745, None)
23/12/03 18:28:51 INFO BlockManagerMasterEndpoint: Registering block manager 094c129fca8a:41745 with 366.3 MiB RAM, BlockManagerId(driver, 094c129fca8a, 41745, None)
23/12/03 18:28:51 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, 094c129fca8a, 41745, None)
23/12/03 18:28:52 INFO SharedState: Setting hive.metastore.warehouse.dir ('null') to the value of spark.sql.warehouse.dir ('file:/code/spark-warehouse').
23/12/03 18:28:52 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|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 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| | |
|3.48851027289842...|[0.06977820545796...]| 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|
|48.1243835079459...|[0.96248767015891...]| 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|
|48.1539002576703...|[0.96307800515340...]| 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|
|47.6785761357096...|[0.95357152277419...]| 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|
|1.82872254349015...|[0.03657445086980...]| 6.3| 0.39| 0.16| 1.4| 0.08| 11.0| 23.0| 0.9955|3.34| 0.56| 9.3| 5.0|[6.3,0.39,0.16,1....] 0.0|
|47.9495689484944...|[0.95899137896988...]| 7.6| 0.41| 0.24| 1.8| 0.08| 4.0| 11.0| 0.9962|3.28| 0.59| 9.5| 5.0|[7.6,0.41,0.24,1....] 0.0|
|48.4901508781797...|[0.96980301756359...]| 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|
|48.1539002576703...|[0.96307800515340...]| 7.1| 0.71| 0.0| 1.9| 0.08| 14.0| 35.0| 0.9972|3.47| 0.55| 9.4| 5.0|[7.1,0.71,0.0,1.9...] 0.0|
|47.9532044267413...|[0.95906408853482...]| 7.8| 0.645| 0.0| 2.0| 0.082| 8.0| 16.0| 0.9964|3.38| 0.59| 9.8| 6.0|[7.8,0.645,0.0,2....] 1.0|
|0.41666666666666...|[0.00833333333333...]| 1.0|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

only showing top 10 rows

None
Accuracy of my wine prediction model = 0.983580922595778
F1 score (Weighted) = 0.9776578095108527
PS C:\pa2-cc>
```

4. Running the Machine learning model without using docker

- Begin by cloning this repository to your local machine.
- Ensure that you have a local Spark environment set up for running this application. If you don't have Spark set up yet, you can follow the instructions provided in the [official Spark documentation](<https://spark.apache.org/docs/latest>).
- Navigate to the 'python file' folder within the cloned repository.
- Place your test data in the 'C:\pa2-cc\data\csv' folder.

By following these steps, you'll be ready to run the trained machine learning model locally without relying on Docker.