CS 643-861 CLOUD COMPUTING

PROGRAMMING ASSIGNMENT 2

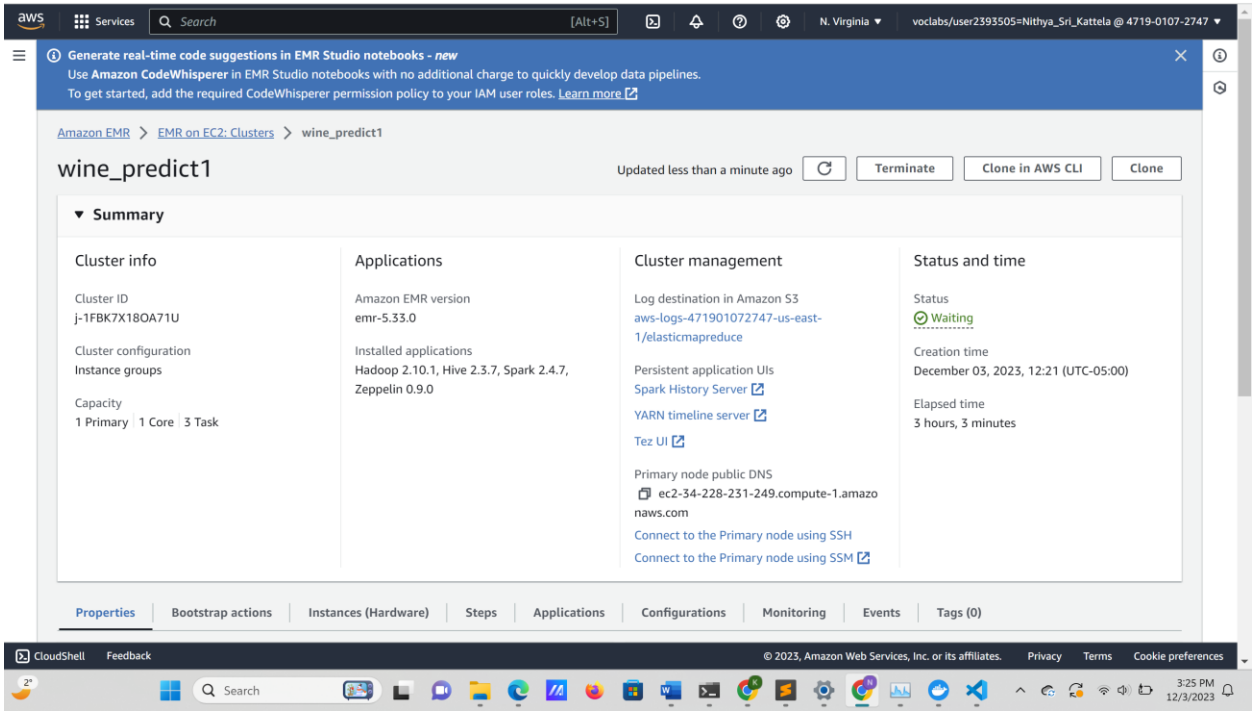
Github Link: https://github.com/nithya2503/cc643861_pa2

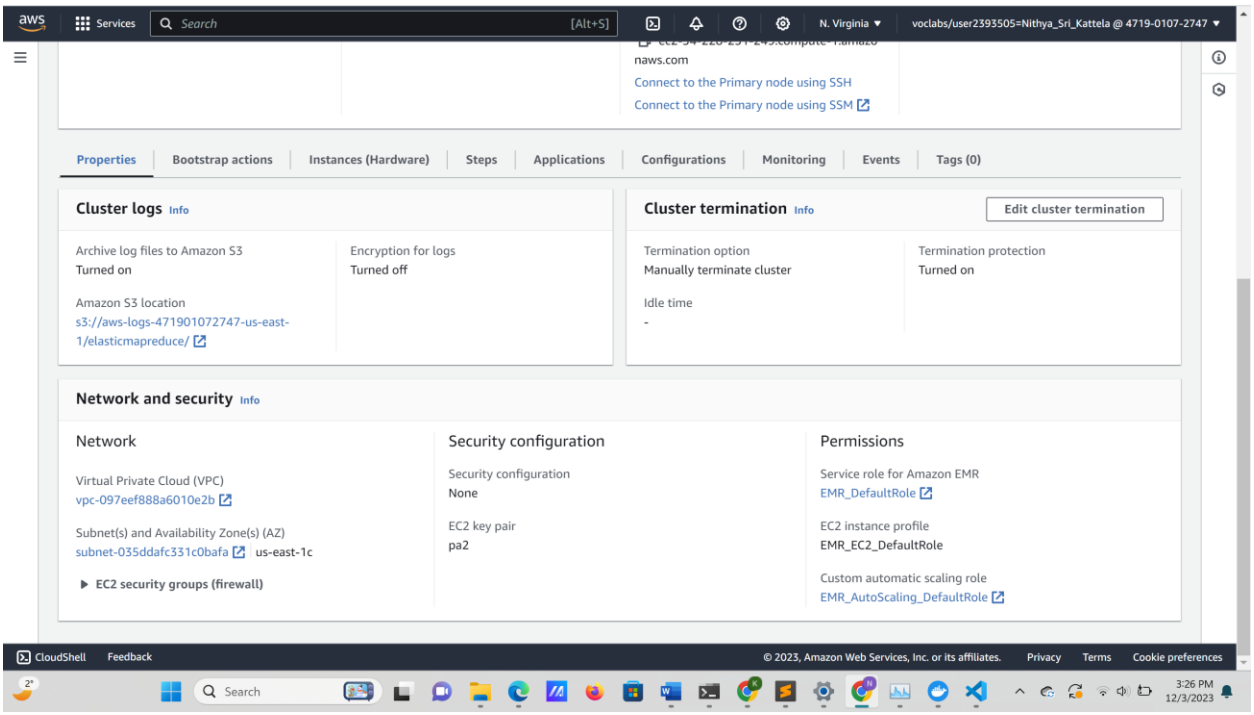
Docker Hub Link: https://hub.docker.com/repository/docker/nithya252423/nk659wineapp/

STEPS TO FOLLOW

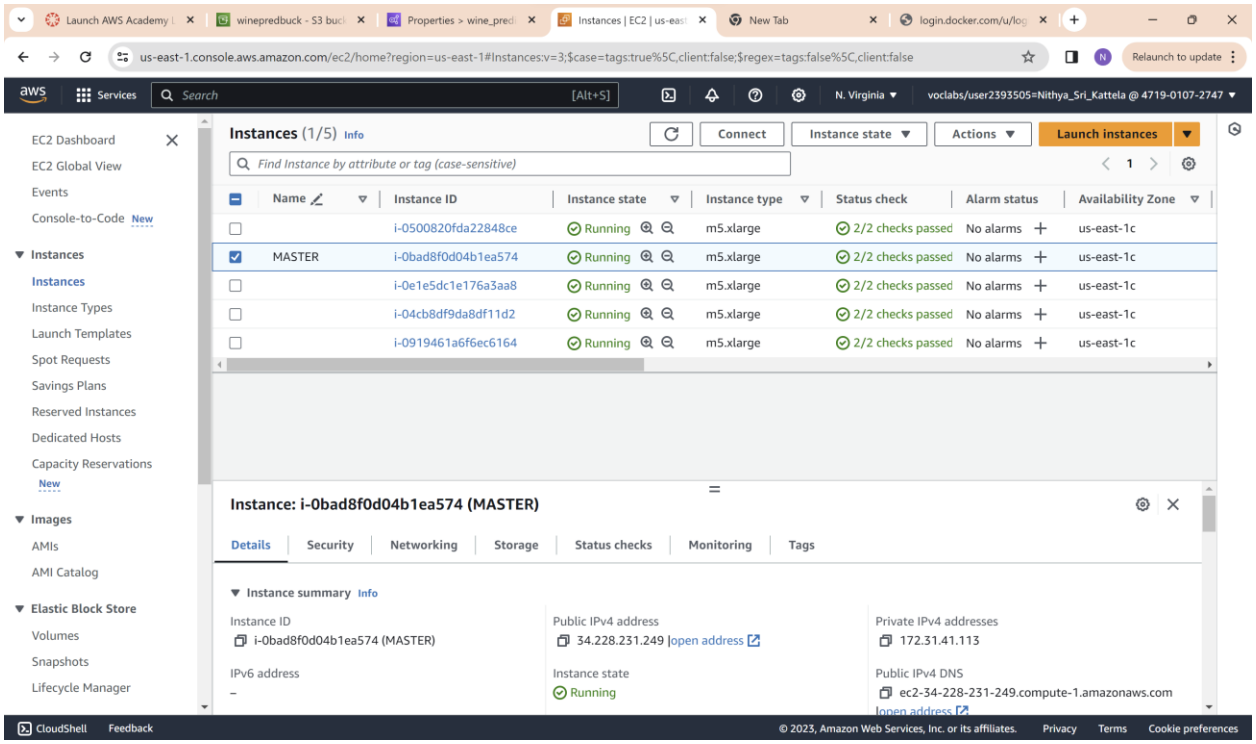
1. To create a spark cluster in AWS

- 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.

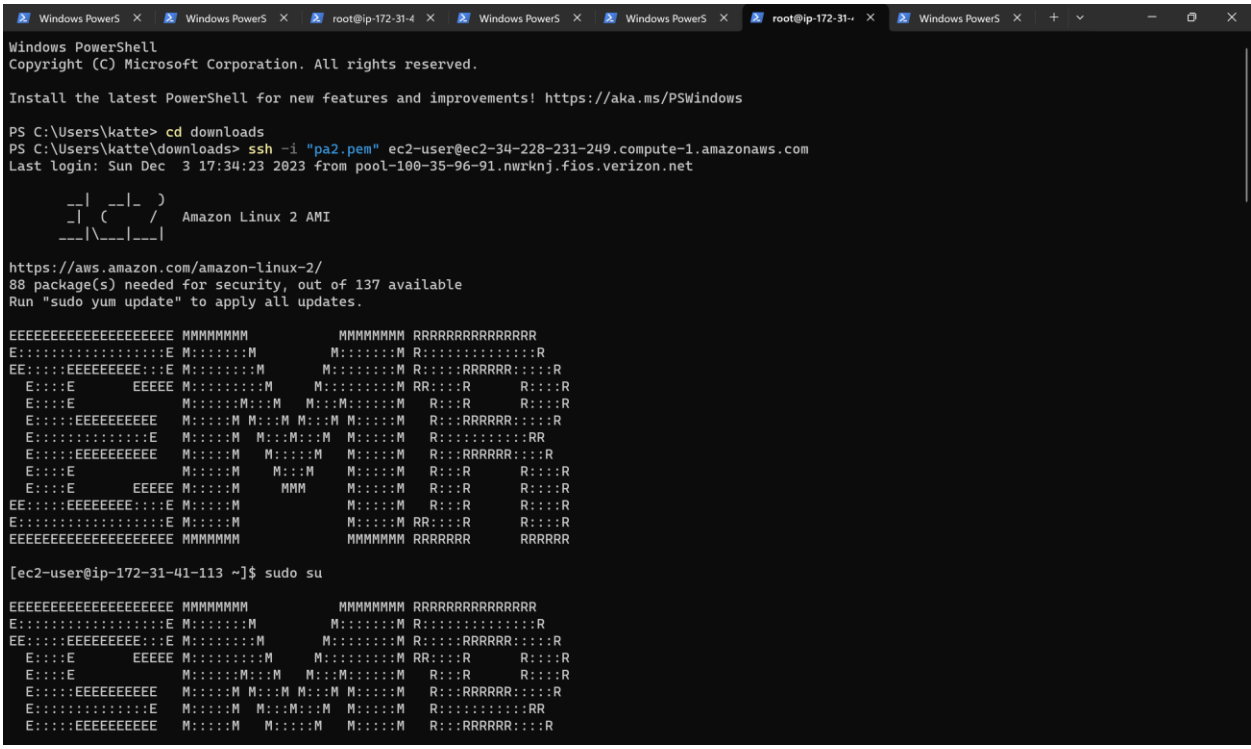


2. **Training the Machine Learning Model with 4 EC2 instances parallelly using Spark cluster**.

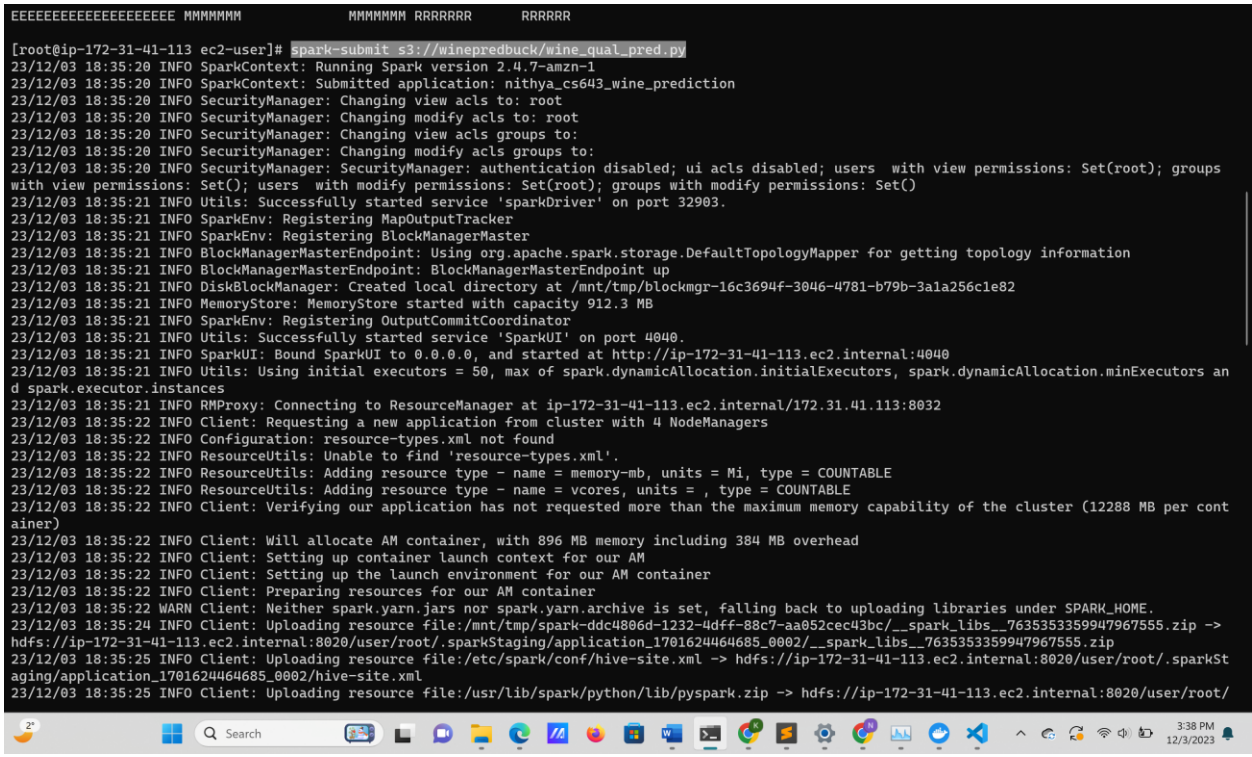
- Connect Master with SSH using CMD or powershell using ssh -I “key” user@Public IPV4 DNS

  ssh -i "pa2.pem" ec2-user@ec2-34-228-231-249.compute-1.amazonaws.com

- Change the user to root user using command ‘’’’ sudo su ‘’’’



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



- 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

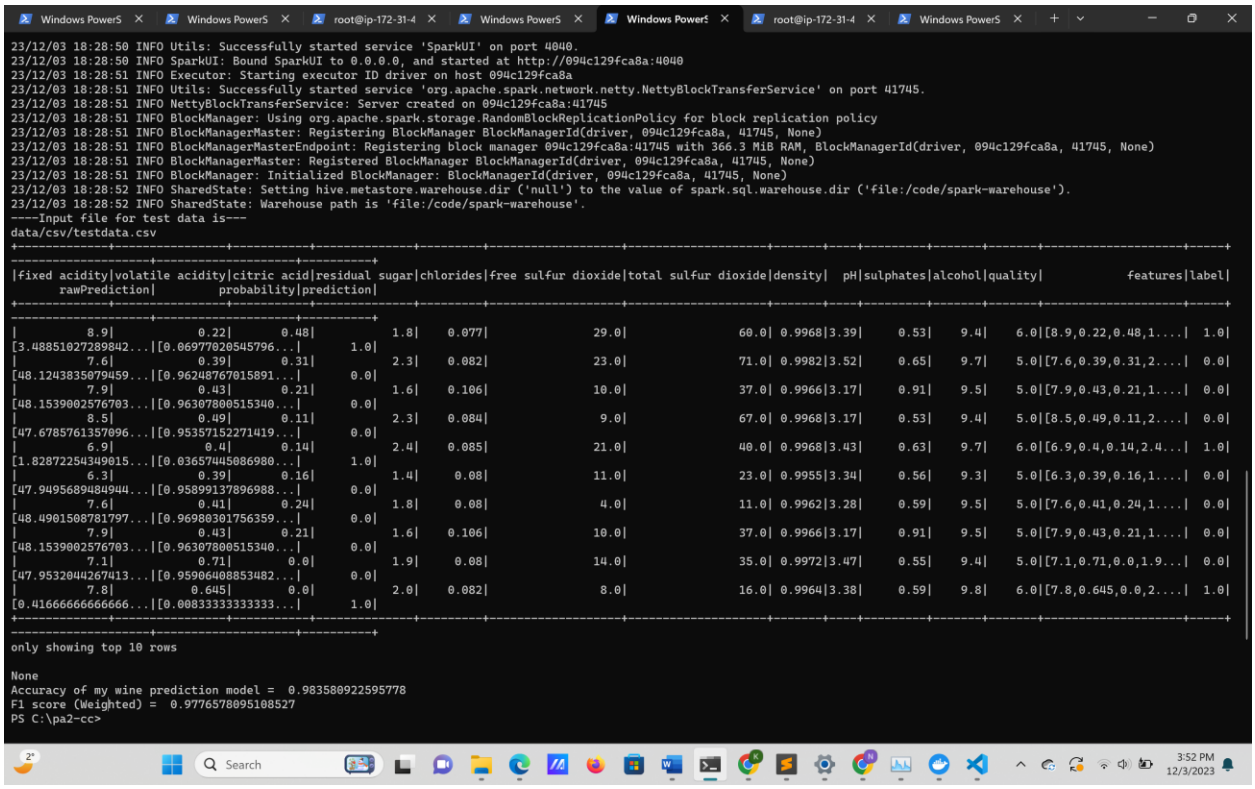
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\Users\katte> 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
89d9a6329459: Pushed
b52f4fea9d5b: Pushed
6d0015a8eed4: Pushed
8717c9eb7d69: Pushed
ca811b885aa1: Pushed
17932e288033: Pushed
dbcf00f2ac01: Pushed
9b6a9c050195: Pushed
061d18e47511: Pushed
ca63f057815d: Pushed
b5d27e9ba5b7: Pushed
c04bfc2b260d: 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.
```



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.