

# AWS Spark Wine Quality Prediction Application

## by Hemanjali Bogadi

FesmaAWSWinePredictionPysparkApp is an application developed with Python's pySpark interface and installed on an AWS EMR cluster. The objective of this project is to simultaneously train a machine learning model on EC2 instances to predict wine quality using publically accessible data, and then utilize the trained model to make wine quality predictions. In order to make deployments easier, the project also uses Docker to produce a container image for a trained machine learning model.

Link to GitHub code - <https://github.com/Hema542/AWSSPARKMLAPK.git>

Link to docker container image:

<https://hub.docker.com/layers/hemanjali693/fesmaawssparkwinepredictionapp/fesmawine-quality-prediction/images/sha256ab902bd02003f713a9514da242ccd34bd7790256f57343a70bc8d2a687a1fefc?context=repo>

### Source files Description

**fesmawine\_prediction.py** - reads Training dataset from S3 and trains model in parallel on EMR spark cluster. Once model is trained, it can be run on provided test data provided via the S3 bucket. This program stores trained model in S3 bucket.

**fesmawine\_test\_data\_prediction.py** - program loads trained model and executes that model on given test data file. This will then print F1 score as metrics for accuracy of the trained model.

**Dockerfile** - creates docker image and run container for easy deployment.

Create Spark cluster in AWS

User can create spark cluster using EMR console provided by AWS.

Steps to create one with 4 ec2 instance:

- Create Key-Pair for EMR cluster using navigation ``EC2-> Network & Security -> Key-pairs``. Use .pem as format. This will download {name of key pair}>.pem file. Keep it safe you will need that to do SSH to EC2 instances.
- Navigate to Amazon EMR console. Then, navigate to clusters-> create cluster.
- Now fill in respective sections:  
``

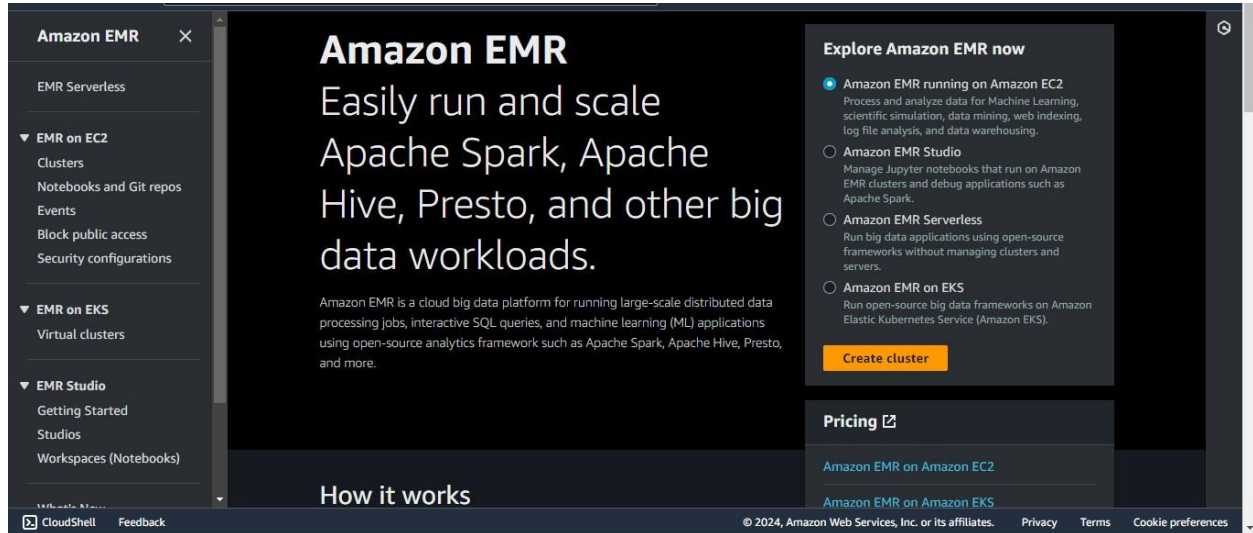
General Configuration -> Cluster Name  
Software Configuration -> EMR 5.33, do select 'Spark: Spark 2.4.7 on Hadoop 2.10.1 YARN and Zeppelin 0.9.0' option menu.

Hardware Configuration -> Make instance count as 4

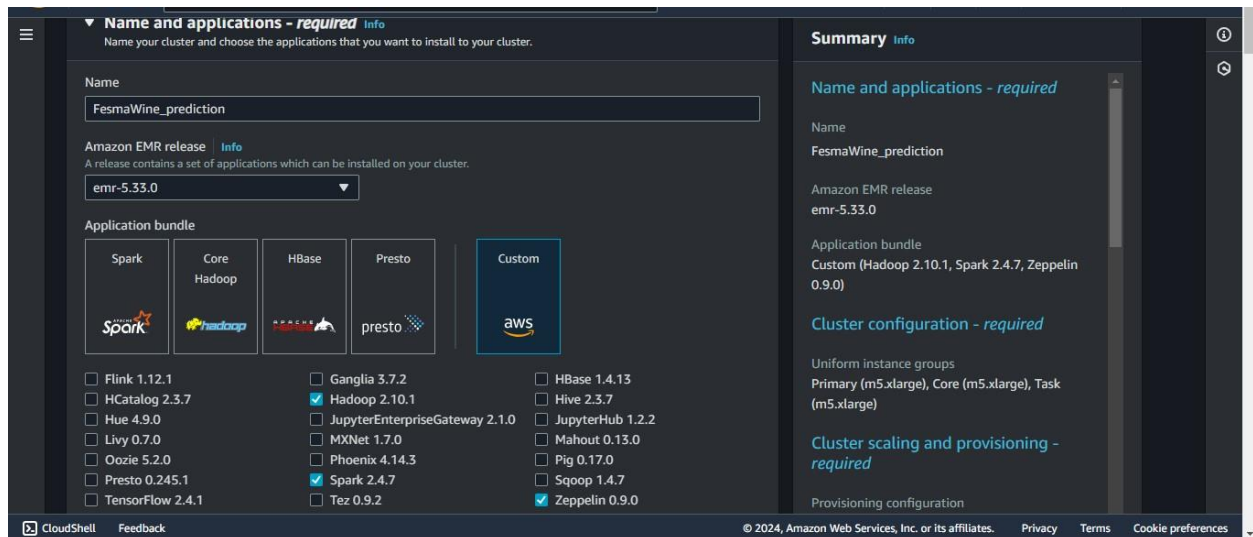
Security Access -> Provide .pem key created in above step.

Rest of parameters can be left default.

- Cluster status should be 'Waiting' on successful cluster creation



The choose the framework of choice and in this case we will configure our cluster to use Spark 2.4.7 on Hadoop 2.10.1 and Zeppelin 0.90.



After that, we make the EMR to have same instance groups for the application.

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](#)

Uniform instance groups

Primary

Choose EC2 instance type

m5.xlarge

4 vCore 16 GiB memory EBS only storage

On-Demand price: \$0.204 per instance/hour

Lowest Spot price: \$0.069 (eu-north-1a)

Actions

☒ Use high availability

Launch highly available, more resilient cluster with three primary nodes on On-Demand Instances. This configuration applies for the lifetime of your cluster. [Learn more](#)

Node configuration - optional

Summary

Info

Name and applications - required

Name

FesmaWine\_prediction

Amazon EMR release

emr-7.1.0

Application bundle

Custom (Hadoop 3.3.6, 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), Task (m5.xlarge)

Cluster scaling and provisioning - required

Windows taskbar

Core

Choose EC2 instance type

m5.xlarge

4 vCore 16 GiB memory EBS only storage

On-Demand price: \$0.204 per instance/hour

Lowest Spot price: \$0.069 (eu-north-1a)

Actions

Node configuration - optional

Task 1 of 1

Remove instance group

Name

Task - 1

Choose EC2 instance type

m5.xlarge

4 vCore 16 GiB memory EBS only storage

On-Demand price: \$0.204 per instance/hour

Lowest Spot price: \$0.069 (eu-north-1a)

Actions

Node configuration - optional

Add task instance group

You can add up to 47 more task instance groups.

Summary

Info

Name and applications - required

Name

FesmaWine\_prediction

Amazon EMR release

emr-7.1.0

Application bundle

Custom (Hadoop 3.3.6, 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), Task (m5.xlarge)

Cluster scaling and provisioning - required

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

EBS root volume

EBS root volume applies to the operating systems and applications that you install on the cluster. [EBS root volume ratio constraints](#)

Size (GiB)

IOPS

Throughput (MiB/s)

15

3000

125

15 - 100 GiB per volume

3000 - 16000 IOPS per volume.

125 - 1000 MiB/s per volume.

General Purpose SSD (gp3)

Choose a maximum ratio of 500:1 between IOPS and volume size.

Choose a maximum ratio of 0.25:1 between throughput and IOPS.

Cluster scaling and provisioning - required

Info

Choose how Amazon EMR should size 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

Summary

Info

Name and applications - required

Name

FesmaWine\_prediction

Amazon EMR release

emr-7.1.0

Application bundle

Custom (Hadoop 3.3.6, Hive 3.1.3, JupyterEnterpriseGateway 2.6.0, Livy 0.8.0, Spark 3.5....)

Cluster configuration - required

Uniform instance groups

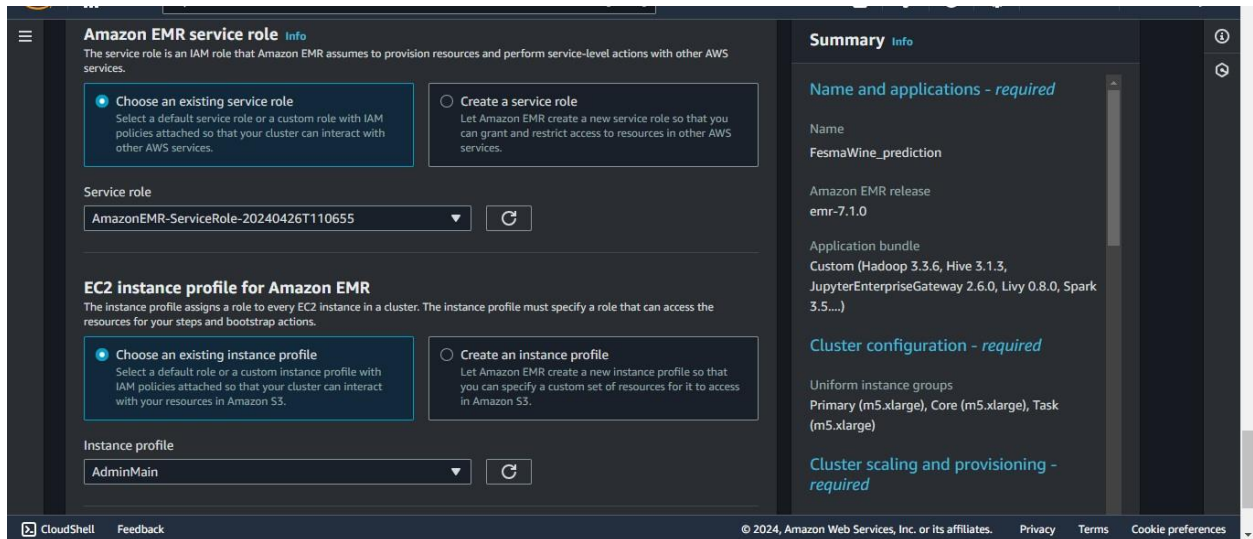
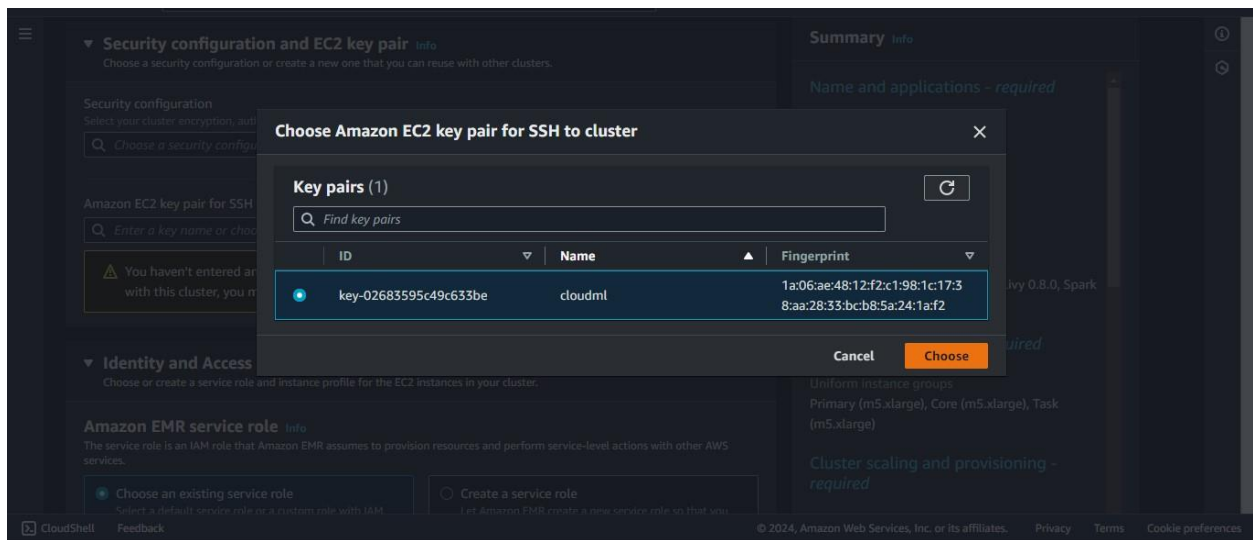
Primary (m5.xlarge), Core (m5.xlarge), Task (m5.xlarge)

Cluster scaling and provisioning - required

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences





Then we click the “Create Cluster” button to create the EMR cluster.



Your cluster "FesmaWine\_prediction" has been successfully created.

Amazon EMR > EMR on EC2: Clusters > FesmaWine\_prediction

## FesmaWine\_prediction

Updated less than a minute ago [Refresh](#) [Terminate](#) [Clone in AWS CLI](#) [Clone](#)

### ▼ Summary

#### Cluster info

Cluster ID  
j-D7XYCK80FAFB

Cluster configuration  
Instance groups

Capacity  
3 Primary 1 Core 4 Task

#### Applications

Amazon EMR version  
emr-5.33.0

Installed applications  
Hadoop 2.10.1, Spark 2.4.7, Zeppelin 0.9.0

#### Cluster management

Log destination in Amazon S3  
[aws-logs-310246848176-eu-north-1/elasticmapreduce](#)

Primary node public DNS  
-

#### Status and time

Status  
Starting

Creation time  
April 26, 2024, 18:48 (UTC+03:00)

Elapsed time  
-6 seconds

[Properties](#) [Bootstrap actions](#) [Instances \(Hardware\)](#) [Steps](#) [Applications](#) [Configurations](#) [Monitoring](#) [Events](#) [Tags \(1\)](#)

#### Cluster logs

[Info](#)

#### Cluster termination and node replacement

[Info](#) [Edit](#)

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Here goes a snapshot of the Spark application on EMR.

#### Cluster info

Cluster ID  
j-D7XYCK80FAFB

#### Applications

Amazon EMR version  
emr-5.33.0

#### Cluster management

Log destination in Amazon S3  
[aws-logs-310246848176-eu-north-1/elasticmapreduce](#)

#### Status and time

Status  
Starting

### Log destination in Amazon S3

S3 buckets > [aws-logs-310246848176-eu-north-1](#) > [elasticmapreduce/](#) > [j-1WO0N29W63A3K/](#) > [AWS\\_ML\\_PARALLEL/](#) > [FesmaAWSWinePredictionPysparkApp/](#) > [src/](#)

#### Objects (5)

 [Refresh](#)

Key	Last modified	Size
<a href="#">TrainingDataset.csv</a>	April 26, 2024, 13:28:23 (UTC+03:00)	67.2 KB
<a href="#">ValidationDataset.csv</a>	April 26, 2024, 13:28:25 (UTC+03:00)	8.6 KB
<a href="#">fesmawine_prediction.py</a>	April 26, 2024, 13:28:19 (UTC+03:00)	4.5 KB
<a href="#">fesmawine_test_data_prediction.py</a>	April 26, 2024, 13:28:21 (UTC+03:00)	2.5 KB
<a href="#">testdata.csv</a>	April 26, 2024, 13:28:22 (UTC+03:00)	67.2 KB

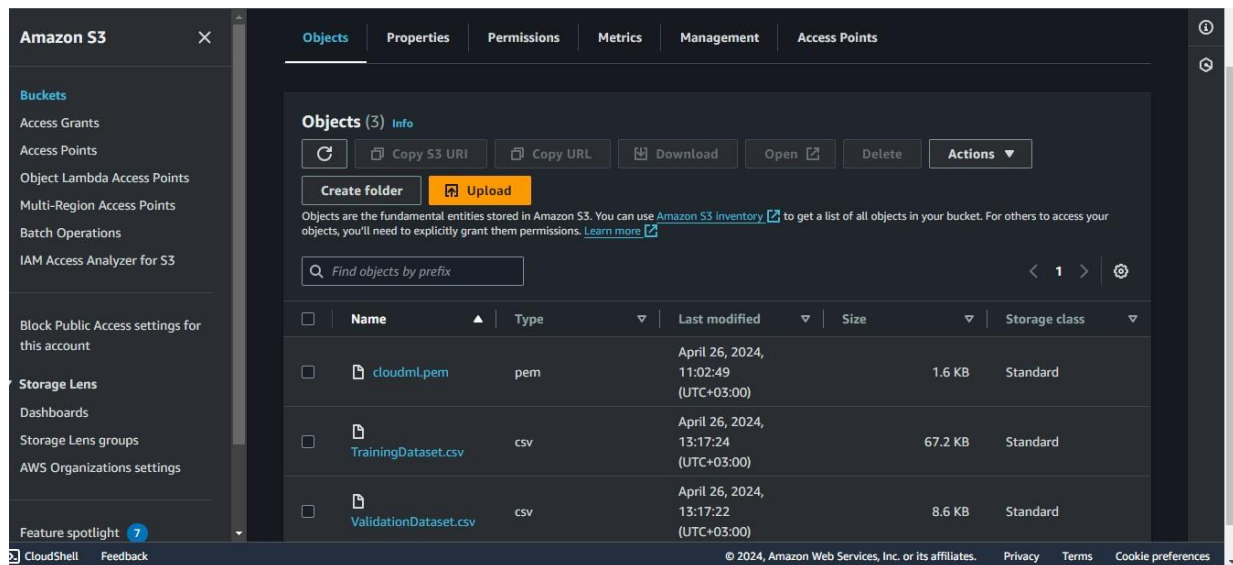
[Close](#)

[s3://aws-logs-310246848176-eu-north-1/elasticmapreduce/](#)

Termination protection  
On

Unhealthy node replacement  
On

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



## 2:TRAINING ML models in the Spark EMR Cluster in parallel with the 4 EC2 instances.

When the cluster has been provisioned and ready to accept jobs, to submit one you can either use step button to add steps or submit them manually. Doing so manually, we will have to perform SSH to the Master of the cluster using the following cmd:

```
ssh -i "cloudml.pem" <<User>>@<<Public IPv4 DNS>>
```

And on successful login to master , we change to root user by running the cmd:

```
sudo su
```

Then finally we submit the job using following command :

```
spark-submit s3://{nameofbucket}/fesmawine_prediction.py
```

[illegible]

Under Ec2 instances click on instances running then connect then it will establish to the connection.

```
C:\Users\HP>sftp -i C:\Users\HP\Downloads\CS643_project2.pem hadoop@ec2-34-231-180-238.compute-1.amazonaws.com
Connected to hadoop@ec2-34-231-180-238.compute-1.amazonaws.com.
sftp>
```

After logging into Emr we have to give permission through Pem key, as we know pem will be generated by using the EMR cluster

We can trace status of the above job in EMR UI application logs. Once status is showing success, a testmodel will be created in s3 bucket-s3://{name of our bucket}.

### 3.Run ML model using Docker

- **Install docker**
- Build the image of the dockerfile using cmd by typing - **docker build -t fesmawine-qualityprediction**
- You can see the image using cmd by typing - **docker image ls**



- You can push this in docker hub repository by typing - **docker push hemanjali693/fesmaawssparkwinepredictionapp: fesmawine-quality-prediction**

```

$ docker build -t fesmawine-quality-prediction -f Dockerfile .
[+] Building 0.0s (0/0) docker:default
2024/04/26 21:16:14 http2: server: error reading preface from client //./pipe/docker_engine: file has already been close
[+] Building 1.7s (21/26) docker:default
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 1.32kB 0.0s
=> [internal] load metadata for docker.io/library/centos:7 1.5s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> CANCELED [ 1/22] FROM docker.io/library/centos:7@sha256:be65f488b7764ad3638f236b7b515b3678369a5124c47b8d32916 0.1s
=> => resolve docker.io/library/centos:7@sha256:be65f488b7764ad3638f236b7b515b3678369a5124c47b8d32916d6487418ea4 0.0s
=> => sha256:be65f488b7764ad3638f236b7b515b3678369a5124c47b8d32916d6487418ea4 1.20kB / 1.20kB 0.0s
=> => sha256:dead07b4d8ed7e29e98de0f4504d87e8880d4347859d839686a31da35a3b532f 529B / 529B 0.0s
=> => sha256:eeb6ee3f44bd0b5103bb561b4c16bcb82328cfe5809ab675bb17ab3a16c517c9 2.75kB / 2.75kB 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 4.94kB 0.0s
=> CACHED [ 2/22] RUN yum -y update && yum -y install python3 python3-dev python3-pip python3-virtualenv java-1 0.0s
=> CACHED [ 3/22] RUN python -V 0.0s
=> CACHED [ 4/22] RUN python3 -V 0.0s
=> CACHED [ 5/22] RUN pip3 install --upgrade pip 0.0s
=> CACHED [ 6/22] RUN pip3 install numpy panda 0.0s
=> CACHED [ 7/22] RUN pip3 install pandas 0.0s
=> CACHED [ 8/22] RUN wget --no-verbose -O apache-spark.tgz "https://archive.apache.org/dist/spark/spark-3.1.2/s 0.0s
=> CACHED [ 9/22] RUN ln -s /opt/spark-3.1.2-bin-hadoop2.7 /opt/spark 0.0s
=> CACHED [10/22] RUN (echo 'export SPARK_HOME=/opt/spark' >> ~/.bashrc && echo 'export PATH=$SPARK_HOME/bin:$PA 0.0s
=> CACHED [11/22] RUN mkdir /code 0.0s
=> CACHED [12/22] RUN mkdir /code/data 0.0s
=> CACHED [13/22] RUN mkdir /code/data/csv 0.0s
=> CACHED [14/22] RUN mkdir /code/data/model 0.0s

=> => sha256:96918c57e42509b97f10c074d80672ecd3bb7dcd38c1bd95960cf291207416 11.98MB / 11.98MB 42.1s
=> => extracting sha256:96918c57e42509b97f10c074d80672ecd3bb7dcd38c1bd95960cf291207416 0.9s
=> [app internal] load metadata for docker.io/library/alpine:latest 5.5s
=> [app auth] library/alpine:pull token for registry-1.docker.io 0.0s
=> [app internal] load .dockerignore 0.1s
=> => transferring context: 671B 0.1s
=> [app base 1/1] FROM docker.io/library/alpine:latest@sha256:c5b1261d6d3e43071626931fc004f70149baeba2c8ec672bd 12.0s
=> => resolve docker.io/library/alpine:latest@sha256:c5b1261d6d3e43071626931fc004f70149baeba2c8ec672bd4f27761f8e 0.1s
=> => sha256:c5b1261d6d3e43071626931fc004f70149baeba2c8ec672bd4f27761f8e1ad6b 1.64kB / 1.64kB 0.0s
=> => sha256:6457d53fb065d6f250e1504b9bc42d5b6c65941d57532c072d929dd0628977d0 528B / 528B 0.0s
=> => sha256:05455a08881ea9cf0e752bc48e61bbd71a34c029bb13df01e40e3e70e0d007bd 1.47kB / 1.47kB 0.0s
=> => sha256:4abcf20661432fb2d719aaf90656f55c287f8ca915dc1c92ec14ff61e67fbaf8 3.41MB / 3.41MB 10.9s
=> => extracting sha256:4abcf20661432fb2d719aaf90656f55c287f8ca915dc1c92ec14ff61e67fbaf8 0.7s
=> [app final 1/2] RUN adduser --disabled-password --gecos "" --home "/nonexistent" --shell "/sb 2.8s
=> [app build 1/2] RUN echo -e '#!/bin/sh\nnecho Hello world from $(whoami)!\nIn order to get your application run 2.6s
=> [app build 2/2] RUN chmod +x /bin/hello.sh 1.0s
=> [app final 2/2] COPY --from=build /bin/hello.sh /bin/ 0.3s
=> [app] exporting to image 0.4s
=> => exporting layers 0.2s
=> => writing image sha256:93de7a9fd5827800be0b3550b103e1039a5447a711917961dc56aafd94ea422e 0.0s
=> => naming to docker.io/library/fesmaawssparkwinepredictionpysparkapp-app 0.0s
[+] Running 1/2
✔ Network fesmaawssparkwinepredictionpysparkapp_default Created 0.2s
- Container fesmaawssparkwinepredictionpysparkapp-app-1 Created 0.5s
Attaching to app-1
app-1 | Hello world from appuser! In order to get your application running in a container, take a look at the comments
in the Dockerfile to get started.
app-1 exited with code 0

```

```

PS C:\Users\MAHESH\OneDrive\Desktop\Anjali\AWS_ML_PARALLEL> docker push hemanjali693/fesmaawssparkwinepredictionapp:fesmawine-quality
-prediction
The push refers to repository [docker.io/hemanjali693/fesmaawssparkwinepredictionapp]
3212334c57d0: Pushed
5f70bf18a086: Pushed
54a537c48445: Pushed
47e31a4d606a: Mounted from library/python
bf4966b4b813: Mounted from library/python
da15a2a37253: Mounted from library/python
89ca33c95b2e: Mounted from library/python
83db175c22e2: Mounted from library/python
c5d13b2949a2: Mounted from library/python
7e43f593c900: Mounted from library/python
072686bcd3db: Mounted from library/python
fesmawine-quality-prediction: digest: sha256:ab902bd02003f713a9514da242ccd34bd7790256f57343a70bc8d2a687alfefc size: 2637
PS C:\Users\MAHESH\OneDrive\Desktop\Anjali\AWS_ML_PARALLEL>
PS C:\Users\MAHESH\OneDrive\Desktop\Anjali\AWS_ML_PARALLEL> docker image ls
REPOSITORY                                TAG                                IMAGE ID                                CREATED                                SIZE
hemanjali693/fesmaawssparkwinepredictionapp  fesmawine-quality-prediction      32682071a7bc                          About an hour ago                     2GB
fesmawine-qualityprediction                 latest                             32682071a7bc                          About an hour ago                     2GB
hemanjali693                               latest                             32682071a7bc                          About an hour ago                     2GB
docker/welcome-to-docker                   latest                             c1f619b6477e                          5 months ago                         18.6MB
hello-world                                latest                             d2c94e258dcb                          12 months ago                        13.3kB

```

A public image has been created and posted on DockerHub. Use the command :

**docker pull hemanjali693/fesmaawssparkwinepredictionapp: fesmawine-quality-prediction** to get the image on your machine.

```

PS C:\Users\MAHESH\OneDrive\Desktop\Anjali\AWS_ML_PARALLEL> docker pull hemanjali693/fesmaawssparkwinepredictionapp:fesmawine-quality
-prediction
fesmawine-quality-prediction: Pulling from hemanjali693/fesmaawssparkwinepredictionapp
Digest: sha256:ab902bd02003f713a9514da242ccd34bd7790256f57343a70bc8d2a687alfefc
Status: Image is up to date for hemanjali693/fesmaawssparkwinepredictionapp:fesmawine-quality-prediction
docker.io/hemanjali693/fesmaawssparkwinepredictionapp:fesmawine-quality-prediction

```

Place your testdata file in a folder (lets call it directory dirA) , which you will mount with docker container and run it using below cmd

**docker run -v C:\Users\MAHESH\OneDrive\Desktop\Anjali\AWS\_ML\_PARALLEL\dirA fesmawine-qualityprediction testdata.csv**

```

----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|[8.9,0.22,0.48,0.077,1.8,0.077,29.0,60.0]|1.0|[82.2080479354075...][0.54805365290271...]|0.0|
|      7.6|      0.39|      0.31|      2.3|      0.082|      23.0|      71.0|      0.9982|3.
52|      0.65|      9.7|      5.0|[7.6,0.39,0.31,0.077,2.3,0.082,23.0,71.0]|0.0|[68.3944260463905...][0.45596284030927...]|1.0|
|      7.9|      0.43|      0.21|      1.6|      0.106|      10.0|      37.0|      0.9966|3.
17|      0.91|      9.5|      5.0|[7.9,0.43,0.21,0.077,1.6,0.106,10.0,37.0]|0.0|[95.1540560921219...][0.63436037394747...]|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|[8.5,0.49,0.11,0.077,2.3,0.084,9.0,67.0]|0.0|[120.977603367684...][0.80651735578456...]|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|[6.9,0.4,0.14,0.077,2.4,0.085,21.0,40.0]|1.0|[52.9461488794193...][0.35297432586279...]|1.0|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
only showing top 5 rows

None
Test Accuracy = 0.7935887412040656
Weighted f1 score = 0.772186634151117

```