

# Wine Quality Prediction Model with Spark on AWS

**Description:** In this individual assignment, the objective is to develop a wine quality prediction machine learning model on the Amazon AWS cloud platform. Utilizing Apache Spark, the model will be trained in parallel across four EC2 instances to enhance computational efficiency. Leveraging Spark's MLlib, the model will be constructed and utilized within the cloud environment. Furthermore, Docker will be employed to streamline model deployment by encapsulating it into a container. The entire implementation will be carried out in Python on Ubuntu Linux, ensuring compatibility and ease of development.

**GitHub Repository:** [https://github.com/saipraneethkommu/Wine\\_prediction](https://github.com/saipraneethkommu/Wine_prediction)

**Docker Hub:** <https://hub.docker.com/repository/docker/praneethdocker1/assignment2/general>

Step by Step Execution:

## Cluster Creation:

1. Log in to your AWS Account and initiate the lab.
2. Access the AWS Console and search for "EMR".
3. Click on "Create cluster" and assign a name to your cluster

▼ **Name and applications - required** [Info](#)

Name your cluster and choose the applications that you want to install to your cluster.

Name


programming Assignment2


Amazon EMR release [Info](#)


A release contains a set of applications which can be installed on your cluster.


emr-7.1.0 ▼


Application bundle


Spark  
Interactive  



Core  
Hadoop  


Flink  


HBase  


Presto  


Trino  


Custom  


- Set the scaling and provisioning parameters: assign 1 instance for core and 5 instances for task.

#### ▼ 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 launch your cluster.

Name	Instance type	Instance(s) size	Use Spot purchasing option
Task - 1	m5.xlarge	<input type="text" value="5"/>	<input type="checkbox"/>
Core	m5.xlarge	<input type="text" value="1"/>	<input type="checkbox"/>

- Enable termination protection and choose to manually terminate the cluster under "Cluster termination and node replacement".

#### ▼ Cluster termination and node replacement [Info](#)

Choose termination settings and protect your cluster from accidental shutdown.

Termination option

☒ **Manually terminate cluster**

☐ Automatically terminate cluster after last step ends

☐ Automatically terminate cluster after idle time (Recommended)

☐ **Use termination protection**

Protects your cluster from accidental termination. If on, you must first turn off protection to terminate the cluster. We recommend turning on termination protection for your long running clusters.

6. Under Security configuration and EC2 key pair, click on create key pair > name the key > select .pem and click on create key and go back and click on Browse and add the key.

### Key pair

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove you own an instance.

Name

assignment2

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type [Info](#)

☒ RSA

☐ ED25519

Private key file format

☒ .pem

For use with OpenSSH

7. Assign IAM roles: EMR\_DefaultRole for Service role and EMR\_EC2\_DefaultRole for Instance profile.

### 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



8. Click on "Create Cluster" to initialize the cluster.

**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

↻

Create IAM role

Provisioning configuration

Core size: 1 instance

Cancel

Clone cluster

9. Access the cluster details by searching for "EMR", then selecting the created cluster.

Clusters (7) Info							↻	View details	Terminate	Clone	Create cluster
Filter clusters by status		Find clusters			Filter clusters by creation date-time		< 1 > ⚙				
<input type="checkbox"/>		Cluster ID	Cluster name	Status	Creation time (UTC-04:00)	Elapsed					
<input type="checkbox"/>			<a href="#">j-22RTMEC060OQW</a>	programming Assignment2	Starting Preparing cluster	April 26, 2024, 01:50	40 sec				

10. Connect to the primary node using SSM and note down the Public IP address.

programming Assignment2				Updated less than a minute ago	↻	Terminate	Clone in AWS CLI	Clone
▼ Summary								
Cluster info		Applications		Cluster management		Status and time		
Cluster ID j-22RTMEC060OQW		Amazon EMR version emr-7.1.0		Log destination in Amazon S3 <a href="#">aws-logs-654654352224-us-east-1/elasticmapreduce</a>		Status Starting		
Cluster configuration Instance groups		Installed applications Hadoop 3.3.6, Hive 3.1.3, JupyterEnterpriseGateway 2.6.0, Livy 0.8.0, Spark 3.5.0		Primary node public DNS <a href="#">ec2-44-195-62-179.compute-1.amazonaws.com</a>		Creation time April 26, 2024, 01:50 (UTC-04:00)		
Capacity 1 Primary 1 Core 5 Task				<a href="#">Connect to the Primary node using SSH</a> <a href="#">Connect to the Primary node using SSM</a>		Elapsed time 1 minute, 16 seconds		

## EC2 Instances

11. Locate the corresponding EC2 instance using the Public IP address.

EC2 Instance Connect

Session Manager

SSH client

EC2 serial console

Instance ID

i-07894eaa1da110b4c

Connection Type

☒ Connect using EC2 Instance Connect  
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

☐ Connect using EC2 Instance Connect Endpoint  
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address

44.195.62.179

Username

Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, root.

Q root

✕

12. Configure security group inbound rules to allow SSH access from your IP address.

Details

Status and alarms [New](#)

Monitoring

Security

Networking

Storage

Ta

▼ Security details

IAM Role

EMR EC2 DefaultRole [↗](#)

Owner ID

654654352224

SSH ▼

TCP

22

My IP ▼

107.122.189.143/32 ✕

13. Connect to the primary node using SSH by copying and running the provided SSH command in your terminal.

### Connect to the primary node using SSH

You can connect to the Amazon EMR primary node using SSH to perform actions like running interactive queries, examining log files, submit Linux commands, and view web interfaces hosted on Amazon EMR clusters. [Learn more](#)

Windows

Mac/Linux

1. Open a terminal window. On Mac OS X, choose Applications > Utilities > Terminal. On other Linux distributions, terminal is typically found at Applications > Accessories > Terminal.

2. To establish a connection to the primary node, enter the following command. Replace `~/assignment2.pem` with the location and filename of the private key file (.pem) that you used to launch the cluster.

```
ssh -i ~/assignment2.pem hadoop@ec2-44-195-62-179.compute-1.amazonaws.com
```

3. Enter yes to dismiss the security warning.

```
Downloads — -zsh — 133x24
praneethkommu@Sais-MacBook-Air Downloads % ssh -i assignment2.pem hadoop@ec2-44-195-62-179.compute-1.amazonaws.com
```

[illegible]

## S3 Bucket Creation


14. Navigate to the AWS Console and search for "S3".
15. Click on "Create Bucket" and assign a name to your bucket.


General purpose buckets

Directory buckets

General purpose buckets (2) [Info](#)

All AWS Regions



 Copy ARN

Empty

Delete

Create bucket

Buckets are containers for data stored in S3.

16. After successful creation, navigate into the bucket and upload the required files (working code and datasets).

## General configuration

AWS Region

US East (N. Virginia) us-east-1

Bucket type

☒ General purpose

Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

☐ Directory - New

Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name

awsbucketwine

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

## ► Advanced settings

After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

Cancel

Create bucket

- Click on the created bucket and click on upload files and then upload the required files (Working code and datasets)

awsbucketwine [Info](#)

Objects

Properties

Permissions

Metrics

Management

Access Points

Objects (0) [Info](#)



Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

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](#)

Find objects by prefix

< 1 > ⚙

Name	Type	Last modified	Size	Storage class
------	------	---------------	------	---------------

No objects

You don't have any objects in this bucket.

Upload

- Now Upload the required files.

## Without Docker Execution:

- When the upload process completes, navigate to the EC2-connected terminal and execute the following command:

**spark-submit s3://awsbucketwine/Wine\_Quality\_Training\_Spark.py** (Training code)

- Once the training code has completed execution, proceed to execute the testing code using the same command:

**spark-submit s3://awsbucketwine/prediction.py**(Testing code)

```
[[hadoop@ip-172-31-10-106 ~]]$ spark-submit s3://awsbucketwine/Wine_Quality_Training_Spark.py
Apr 27, 2024 1:13:05 AM org.apache.spark.launcher.Log4jHotPatchOption staticJavaAgentOption
WARNING: spark.log4jHotPatch.enabled is set to true, but /usr/share/log4j-cve-2021-44228-hotpatch/jdk17/

Starting Spark Application
24/04/27 01:13:09 INFO SparkContext: Running Spark version 3.5.0-amzn-1
24/04/27 01:13:09 INFO SparkContext: OS info Linux, 6.1.84-99.169.amzn2023.x86_64, amd64
24/04/27 01:13:09 INFO SparkContext: Java version 17.0.10
24/04/27 01:13:09 INFO ResourceUtils: =====
24/04/27 01:13:09 INFO ResourceUtils: No custom resources configured for spark.driver.
24/04/27 01:13:09 INFO ResourceUtils: =====
24/04/27 01:13:09 INFO SparkContext: Submitted application: WineQualityPrediction
24/04/27 01:13:09 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpus -> name: cpus, amount: 4)
24/04/27 01:13:09 INFO ResourceProfile: Limiting resource is cpus at 4 tasks per executor
24/04/27 01:13:09 INFO ResourceProfileManager: Added ResourceProfile id: 0
24/04/27 01:13:09 INFO SecurityManager: Changing view acls to: hadoop
24/04/27 01:13:09 INFO SecurityManager: Changing modify acls to: hadoop
24/04/27 01:13:09 INFO SecurityManager: Changing view acls groups to:
24/04/27 01:13:09 INFO SecurityManager: Changing modify acls groups to:
```



```

Reading training CSV file from s3://awsbucketwine/TrainingDataset.csv
Reading validation CSV file from s3://awsbucketwine/ValidationDataset.csv
Creating VectorAssembler
Creating StringIndexer
Caching data for faster access
Creating RandomForestClassifier
Creating Pipeline for training
Transforming data using the trained model
Evaluating the trained model on the validation set
Test Accuracy of wine prediction model = 0.99375
/usr/lib/spark/python/lib/pyspark.zip/pyspark/sql/context.py:158: FutureWarning: Deprecat
Weighted f1 score of wine prediction model = 0.9933730158730157
Retraining model on multiple parameters using CrossValidator
Fitting CrossValidator to the training data
24/04/27 01:14:26 ERROR TransportResponseHandler: Still have 1 requests outstanding when
Saving the best model to new param `model`
Test Accuracy of wine prediction model (after CrossValidation) = 0.96875
Weighted f1 score of wine prediction model (after CrossValidation) = 0.9541901629072682
Saving the best model to S3

```

21. Upon successful execution, the testing results will be obtained.

## Execution with Docker

22. Set up a Docker repository.
23. Configure the Dockerfile according to your requirements.
24. Make the necessary modifications to your code.
25. Open a new terminal to check locally.
26. Build an image in Docker using the command: **docker build -t train\_wine .**

```

praneethkommu@Sais-MacBook-Air Wine % docker build -t train_wine .
[+] Building 1.1s (20/20) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.26kB
=> [internal] load metadata for docker.io/library/centos:7
=> [auth] library/centos:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [ 1/14] FROM docker.io/library/centos:7@sha256:be65f488b7764ad3
=> [internal] load build context
=> => transferring context: 4.63kB
=> CACHED [ 2/14] RUN yum -y update && yum -y install python3 pyth
=> CACHED [ 3/14] RUN python3 -V
=> CACHED [ 4/14] RUN pip3 install --upgrade pip
=> CACHED [ 5/14] RUN pip3 install awscli
=> CACHED [ 6/14] RUN pip3 install numpy pandas
=> CACHED [ 7/14] WORKDIR /opt
=> CACHED [ 8/14] RUN wget --no-verbose -O apache-spark.tgz "https

```



27. Link the build to the Docker repository with: **docker build praneethdocker1/assignment2:praneeth**
28. Push the image to Docker with: **docker push praneethdocker1/assignment2:praneeth**

```
praneethkommu@Sais-MacBook-Air Wine % docker push praneethdocker1/assignment2:praneeth
The push refers to repository [docker.io/praneethdocker1/assignment2]
a532ca2e8309: Pushed
3aa477d27306: Pushed
85d266bdc8fe: Pushed
84a098af3fd4: Pushed
bcb28d9f9c2a: Pushed
d741d4e12584: Pushed
15ea834d4c4c: Pushed
5f70bf18a086: Pushed
fec425c5a135: Pushed
342f15d88e26: Pushed
833ecbd56391: Pushed
8b25501926a0: Pushed
b409d03e7edb: Pushed
65f23ff12f4d: Mounted from library/centos
praneeth: digest: sha256:e2bb9d3af6d15f04441ba5035a873e679489a3eca69f3a66956cb154e169c30e size: 3262
```

29. To execute, use the command: **docker run train\_wine .**
30. After successful execution, verify the code locally.
31. Navigate to the EC2-connected terminal for cloud testing.
32. Use the command **sudo su** to switch to the root user.

```
[[hadoop@ip-172-31-10-106 ~]$ sudo su
```

```
EEEEEEEEEEEEEEEEEEEE MMMMMMM      MMMMMMM RRRRRRRRRRRRRR
E::::::::::::::::::::E M::::::::M      M::::::::M R:::::::::R
EE::::::::EEEEEEEE::::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 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 MMMMMMM      MMMMMMM RRRRRR      RRRRR
```

```
[[root@ip-172-31-10-106 hadoop]# sudo systemctl start docker
[[root@ip-172-31-10-106 hadoop]# sudo systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
```

33. Start Docker with: **sudo systemctl start docker .**
34. Enable Docker with: **sudo systemctl enable docker .**
35. Pull the image from docker using the command: **docker pull praneethdocker1/assignment2:praneeth**

```
[[root@ip-172-31-10-106 hadoop]# docker pull praneethdocker1/assignment2:praneeth
praneeth: Pulling from praneethdocker1/assignment2
6717b8ec66cd: Pull complete
04fbb41f633a: Pull complete
3a3951e056be: Pull complete
f14b9a56231e: Pull complete
095935b81d71: Pull complete
be2d354dcfd0: Pull complete
4f4fb700ef54: Pull complete
1f2b3b500e9e: Pull complete
dd61aa7aa6f9: Pull complete
e6432b6a7ef0: Pull complete
e98c6d43515b: Pull complete
ca404c38a84a: Pull complete
a72f83e06635: Pull complete
369a152b3ce9: Pull complete
Digest: sha256:e2bb9d3af6d15f04441ba5035a873e679489a3eca69f3a66956cb154e169c30e
Status: Downloaded newer image for praneethdocker1/assignment2:praneeth
```

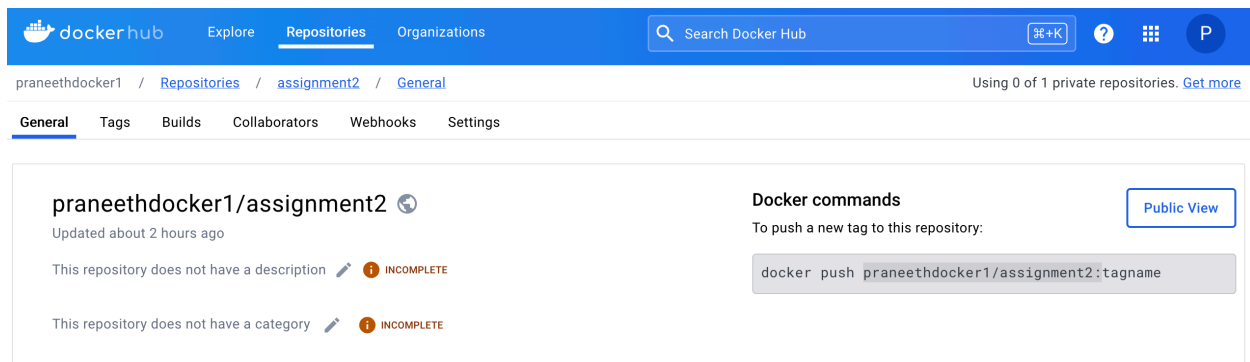
36. For running the image, use the command: **docker run praneethdocker1/assignment2:praneeth**

```
Reading training CSV file from TrainingDataset.csv
Reading validation CSV file from ValidationDataset.csv
Creating VectorAssembler
Creating StringIndexer
Caching data for faster access
Creating RandomForestClassifier
Creating Pipeline for training
Transforming data using the trained model
Evaluating the trained model on the validation set
Test Accuracy of wine prediction model = 0.99375
/opt/spark/python/lib/pyspark.zip/pyspark/sql/context.py:160: FutureWarning: Deprecated e() instead.
Weighted f1 score of wine prediction model = 0.9933730158730157
Retraining model on multiple parameters using CrossValidator
Fitting CrossValidator to the training data
Saving the best model to new param 'model'
Test Accuracy of wine prediction model (after CrossValidation) = 0.96875
/opt/spark/python/lib/pyspark.zip/pyspark/sql/context.py:160: FutureWarning: Deprecated e() instead.
Weighted f1 score of wine prediction model (after CrossValidation) = 0.9541901629072682
```

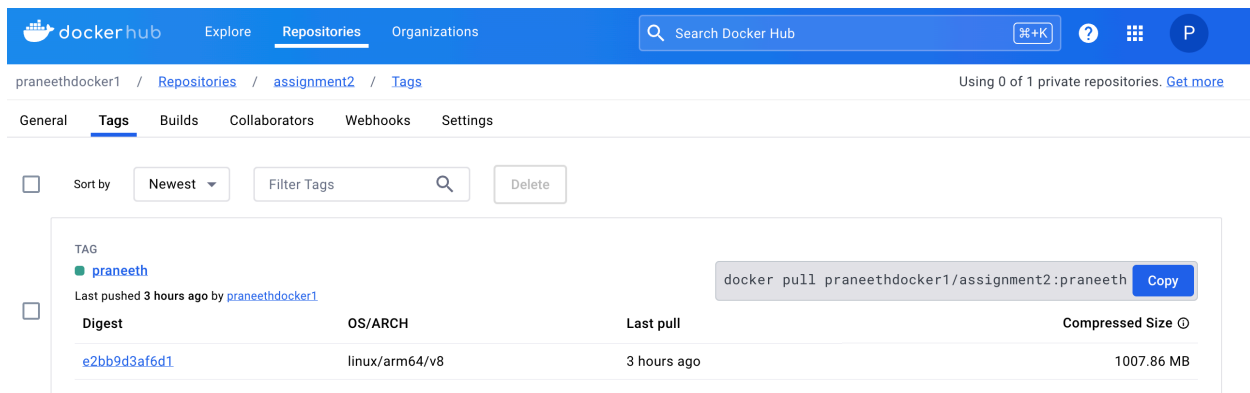
37. After the successful execution, we will see the testing results.

## Docker:

<https://hub.docker.com/repository/docker/praneethdocker1/assignment2/general>



The screenshot shows the Docker Hub repository page for `praneethdocker1/assignment2`. The page is under the 'General' tab. It indicates the repository was updated about 2 hours ago. There are two 'INCOMPLETE' status messages: 'This repository does not have a description' and 'This repository does not have a category'. On the right, under 'Docker commands', it provides the command `docker push praneethdocker1/assignment2:tagname` to push a new tag. A 'Public View' button is also present.



The screenshot shows the Docker Hub repository page for `praneethdocker1/assignment2` under the 'Tags' tab. It displays a list of tags with columns for TAG, Digest, OS/ARCH, Last pull, and Compressed Size. The tag `praneeth` is highlighted, showing a digest of `e2bb9d3af6d1` for the `linux/arm64/v8` architecture, pulled 3 hours ago, with a size of 1007.86 MB. A 'docker pull' command is shown: `docker pull praneethdocker1/assignment2:praneeth`.

TAG	Digest	OS/ARCH	Last pull	Compressed Size
<code>praneeth</code>	<code>e2bb9d3af6d1</code>	<code>linux/arm64/v8</code>	3 hours ago	1007.86 MB