

PROGRAMMING ASSIGNMENT -02

Wine Quality Prediction

Email Id: sk3568@njit.edu

GitHub: <https://github.com/sravyakganti/Wine-quality-prediction->

Docker Hub: <https://hub.docker.com/repository/docker/sravyakganti/cs643-programming-assignment-2/general>

Open the AWS Console and log in.

Choose AWS EMR service from the list of AWS services, and then choose EMR on EC2 Clusters.

You'll see the Clusters page. It is evident that there aren't any clusters in use. We must establish a cluster.

Select the "Create cluster" option to start a cluster.

[Amazon EMR](#) > EMR on EC2: Clusters

The screenshot shows the AWS EMR console interface. At the top, there's a header bar with 'Clusters (0) Info', a refresh button, and buttons for 'View details', 'Terminate', 'Clone', and 'Create cluster'. Below this is a search bar labeled 'Find clusters' and a filter dropdown 'Filter clusters by status'. A table with columns 'Cluster ID', 'Cluster name', and 'Status' is shown, but it is empty with the message 'No Clusters' and 'No Clusters to display.'

The cluster can have any name you choose. Additionally, confirm that the EMR version is the most recent one, as indicated by the figure below. Additionally, choose Spark Interactive from the Application package menu.

Amazon EMR > EMR on EC2: Clusters > Create cluster

Create cluster Info

Name and applications - required Info

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

Name

Amazon EMR release Info

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

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
☐ Phoenix 5.1.3
☒ Spark 3.5.0
☐ Tez 0.10.2
 ☐ Flink 1.18.1
☒ Hadoop 3.3.6
☒ JupyterEnterpriseGateway 2.6.0
☐ MXNet 1.9.1
☐ Pig 0.17.0
☐ Sqoop 1.4.7
☐ Trino 435
 ☐ HBase 2.4.17
☒ Hive 3.1.3
☐ JupyterHub 1.5.0
☐ Oozie 5.2.1
☐ Presto 0.284
☐ TensorFlow 2.11.0
☐ Zeppelin 0.10.1

Summary Info

Name and applications - required

Name
winequalitypred

Amazon EMR release
emr-7.1.0

Application bundle
Spark Interactive (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

Provisioning configuration

For instance type, I have selected m5.xlarge. We can choose anything according to our preference.

Uniform instance groups

Primary

Choose EC2 instance type

4 vCore 16 GiB memory EBS only storage
 On-Demand price: - Lowest Spot price: -

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

Core

Choose EC2 instance type

4 vCore 16 GiB memory EBS only storage
 On-Demand price: - Lowest Spot price: -

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

Task 1 of 1

Remove instance group

For cluster scalability and provisioning, set the Instance size for Core to 1 and Task-2 to 3.

▼ **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
Core	m5.xlarge	<input type="text" value="2"/>	<input type="checkbox"/>
Task - 1	m5.xlarge	<input type="text" value="2"/>	<input type="checkbox"/>

Choose the security groups for the Primary, Core, and Task nodes in the EC2 security groups as indicated below.

▼ **EC2 security groups (firewall)**

 **Change notice**
We've updated the names of some security groups to use more inclusive language. For example, groups that included terms like "master" and "slave" now use the terms "primary" and "core" instead.

Primary node

EMR-managed security group
EMR will automatically update the selected group.

ElasticMapReduce-Primary
sg-0d1c7310b5c184d93 ▼

Additional security groups - *optional*
Select up to 4 additional security groups.

Choose additional security groups ▼

Core and task nodes

EMR-managed security group
EMR will automatically update the selected group.

ElasticMapReduce-Core
sg-0d05add22fb00d4e0 ▼

Additional security groups - *optional*
Select up to 4 additional security groups.

Choose additional security groups ▼

To stop the cluster from ending automatically, make sure you choose the option to manually terminate the cluster. But this is not a suggested option.

▼ 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.

Unhealthy node replacement - *new* [Info](#)

- ☒ Turn on
Amazon EMR gracefully stops processes on unhealthy nodes to minimize data loss and job interruptions. It quickly replaces unhealthy nodes with new EC2 instances to keep your jobs running smoothly.
- ☐ Turn off
Amazon EMR adds unhealthy nodes to a denylist while keeping them in the cluster, allowing you continued access for troubleshooting.

To access the cluster via SSH, create an EC2 key pair in AWS. Ensure that you securely store the .pem file associated with the key pair, as it will be required for establishing an SSH connection to the cluster.

▼ Security configuration and EC2 key pair [Info](#)

Choose a security configuration or create a new one that you can reuse with other clusters.

Security configuration

Select your cluster encryption, authentication, and instance metadata service settings.

Amazon EC2 key pair for SSH to the cluster [Info](#)

Choose the IAM Roles accordingly:

▼ Identity and Access Management (IAM) roles - *required* [Info](#)

Choose or create a service role and instance profile for the EC2 instances in your cluster.

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 ▼



After initiating the cluster creation process, you can monitor its progress on the Clusters page of the EMR service. Initially, the cluster's status will be displayed as "Starting," indicating that it is in the process of being provisioned. Over time, the status will transition to "Waiting," signifying that the cluster has been successfully created and is ready for use, as illustrated in the provided image.

✓ Your cluster "winequalitypred" has been successfully created. ✕

[Amazon EMR](#) > EMR on EC2: Clusters

Clusters (1) [Info](#)



[View details](#)

[Terminate](#)

[Clone](#)

[Create cluster](#)

Filter clusters by status ▼

Find clusters

Filter clusters by creation date-time

< 1 > ⚙



Cluster ID ▼

Cluster name ▼

Status

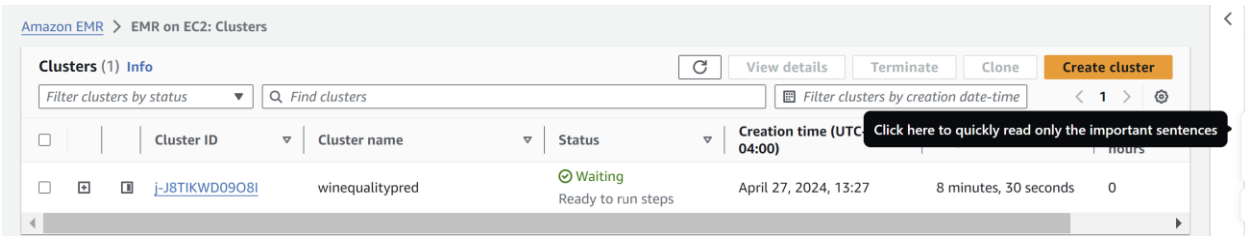


[j-J8TIKWD09O8I](#)

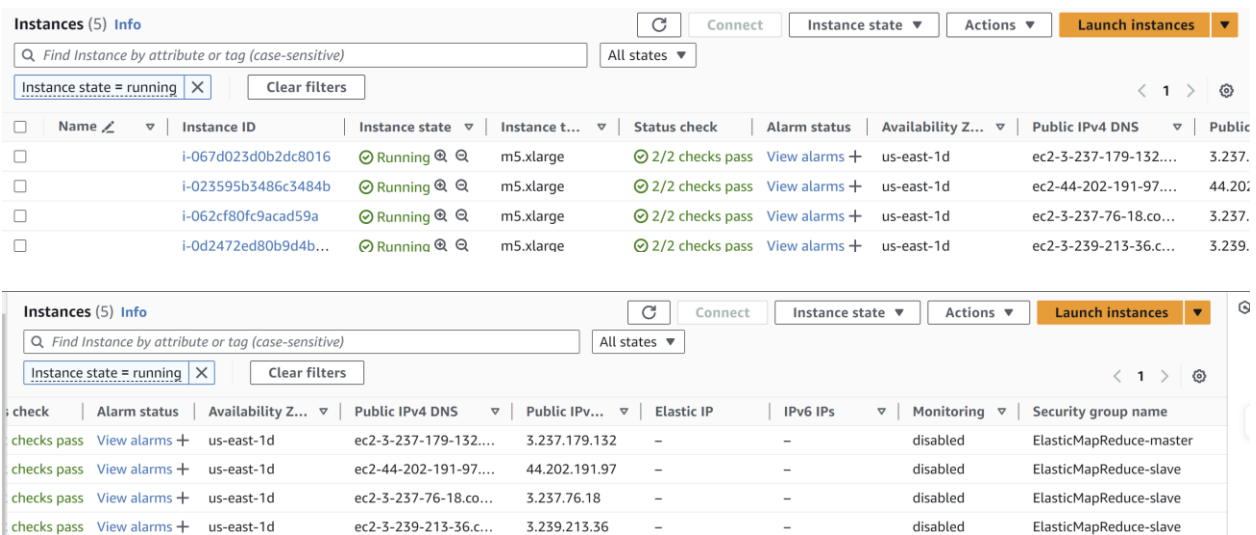
winequalitypred

Starting

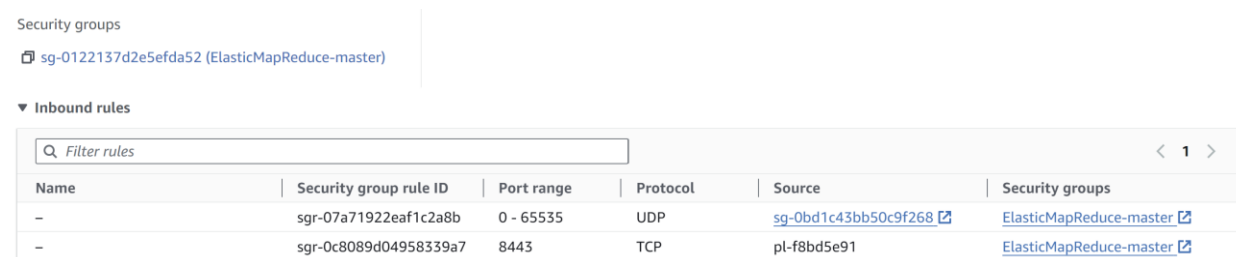
Preparing cluster



Navigate to the EC2 Instances page in the AWS Management Console. On this page, you will observe that a total of 4 EC2 instances have been launched as part of the EMR cluster creation process. Among these instances, one functions as the Master node, responsible for coordinating and managing the cluster operations. The remaining four instances serve as Slave nodes, dedicated to executing distributed tasks and computations within the cluster, as illustrated in the accompanying image.



Within the EC2 service in the AWS Management Console, locate the security group named "ElasticMapReduce-Master" associated with the EMR cluster you created. Once you have identified this security group, click on its corresponding Security Group ID to access its configuration details.



Once you have accessed the configuration details of the "ElasticMapReduce-Master" security group, navigate to the section that displays the inbound network traffic rules. In this section, locate and click on the option that allows you to modify or edit the existing inbound rules.

Inbound rules									
Inbound rules (7)									
<input type="text" value="Search"/> < 1 >									
<input type="checkbox"/>	Name	Security group r...	IP version	Type	Protocol	Port range	Source		
<input type="checkbox"/>	-	sgr-07a71922eaf1c...	-	All UDP	UDP	0 - 65535	sg-0bd1c43l		
<input type="checkbox"/>	-	sgr-0c8089d04958...	-	Custom TCP	TCP	8443	pl-f8bd5e91		

Within the inbound rules configuration section, locate the button or option that allows you to create a new rule. Click on this button, and then specify port numbers 22 and 4040 as the ports to be opened for inbound traffic. Configure any additional settings for these rules according to the provided instructions or screenshot. Once you have added these new rules, click on the "Save rules" button to apply the changes to the security group.

-	SSH	TCP	22	Anywh...	Q	Spark Web Instance	Delete
					0.0.0.0/0 X		
-	Custom TCP	TCP	4040	Anywh...	Q		Delete
					0.0.0.0/0 X		

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Next, navigate to the Amazon Simple Storage Service (S3) within the AWS Management Console. The purpose of this step is to create a new S3 bucket, which will serve as a storage location for the dataset files required.

Once you have accessed the Amazon S3 service, locate and click on the option or button labeled "Create Bucket" to initiate the process of creating a new S3 bucket.

When prompted to provide a name for the new S3 bucket, enter "dataset-programming-assignment-2" as the bucket name. After specifying the bucket name, scroll down to the bottom of the page, and click on the "Create bucket" button to finalize the creation of the S3 bucket with the given name.

Create bucket [Info](#)

Buckets are containers for data stored in S3.

General configuration

AWS Region

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

Bucket type [Info](#)

☒ 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 [Info](#)

dataset-programming-assignment-2

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

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

Format: s3://bucket/prefix

Locate the newly created S3 bucket named "dataset-programming-assignment-2" in the list of buckets displayed and click on its name to access the bucket's contents and configuration options.

[General purpose buckets](#) | [Directory buckets](#)

General purpose buckets (1) [Info](#) [All AWS Regions](#)

[Refresh](#) [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

Name ▲

AWS Region ▼

IAM Access Analyzer

Creation date ▼

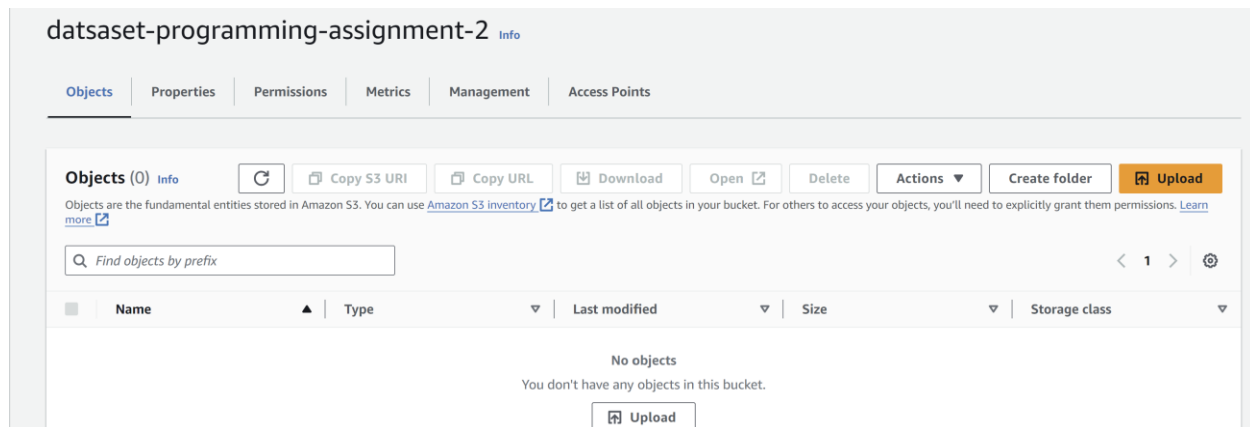
☐ dataset-programming-assignment-2

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

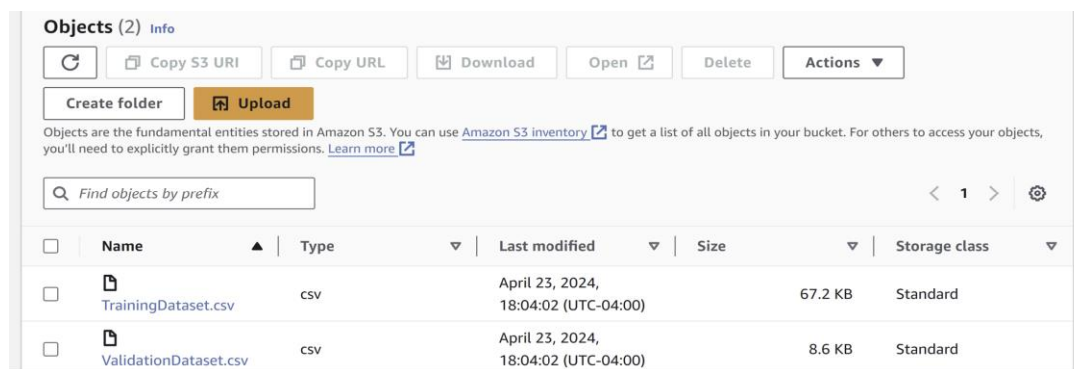
[View analyzer for us-east-1](#)

April 27, 2024, 14:10:42 (UTC-04:00)

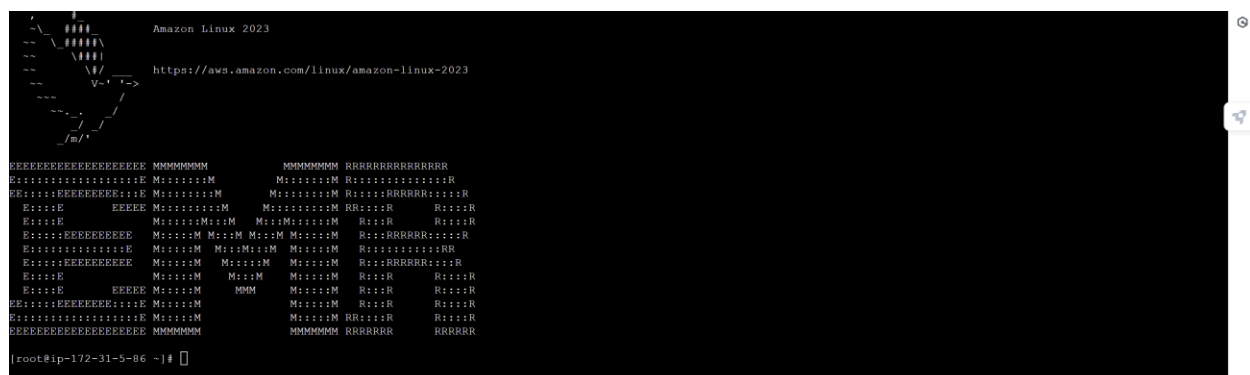
Once you have accessed the contents of the "dataset-programming-assignment-2" S3 bucket, look for the "Upload" button or option, and click on it to initiate the process of uploading files to the bucket.



From the upload interface, locate the option to add or select files for upload. Use this option to browse and choose the .csv files containing the dataset from the assignment. After selecting the desired files, proceed to click on the "Upload" button to initiate the transfer of the dataset files to the "dataset-programming-assignment-2" S3 bucket. Upon successful upload, you should have two .csv files stored in the S3 bucket, named "ValidationDataset.csv" and "TrainingDataset.csv," respectively.



Goto EC2 instances (running), select the Instance ID which is corresponding to the master node. Click on that and select Connect. It will establish the connection as per below fig.



To set up the necessary credentials for accessing AWS services from the Master node EC2 instance, we need to configure the AWS credentials on that instance. Follow these steps:
Open a terminal window and connect to the Master node EC2 instance using SSH.

In the terminal of the Master node, execute the following command to create a new directory named **.aws**: **mkdir .aws**

Next, create an empty file named "credentials" inside the ".aws" directory by running the command: **touch .aws/credentials**

Open the "credentials" file in a text editor by executing: **vi .aws/credentials**

You can now paste your AWS access credentials (access key ID and secret access key) into this file and save the changes. This will allow the Master node instance to authenticate with AWS services using the provided credentials.

```
AWS CLI:
Copy and paste the following
into ~/.aws/credentials

[default]
aws_access_key_id=ASIAQ3EGQ
ESQEGYDCXM
aws_secret_access_key=wIKJsI
ZthQX+e0ugbneobjT53o54wK/TS
kLXvc3
aws_session_token=IQoJb3JpZ
luX2VjEjD////////wEaCXVzL
dlc3QtMiJGMEQCIxhcxzg6dr6K
qileUqOM7xIoTQY6FdZHGgxFzL3
QTAiBKnvi3TqrNa/orrNTKwE7Gs
```

sudo yum update: This command updates all the installed packages on your system to their latest versions using the YUM package manager.

sudo yum install git: This command installs Git, a distributed version control system, on your system using the YUM package manager.

pip install pyspark findspark boto3 numpy pandas scikit-learn datetime

git clone <https://github.com/sravyakganti/Wine-quality-prediction-.git>

By executing this command, Git will download a complete copy of the remote repository, including all its files and revision history, and create a new directory named "CS643_Programming_assignment_2" in your current working directory. This local copy allows you to work on the project files, make changes, and potentially contribute those changes back to the remote repository.

Now, execute the below commands to launch the Spark Application.

spark-submit --master yarn CS643_Programming_assignment_2/WineTraining.py

spark-submit --master yarn CS643_Programming_assignment_2/WineTesting.py > output.txt

```

Apr 26, 2024 8:56:28 PM org.apache.spark.launcher.Log4jHotPatchOption staticJavaAgentOption
WARNING: spark.log4jHotPatch.enabled is set to true, but /usr/share/log4j-cve-2021-44228-hotpatch/jdk17/Log4jHotPatchFat.jar does not exist at the configured location

24/04/26 20:56:32 INFO SparkContext: Running Spark version 3.5.0-amzn-1
24/04/26 20:56:32 INFO SparkContext: OS info Linux, 6.1.84-99.169.amzn2023.x86_64, amd64
24/04/26 20:56:32 INFO SparkContext: Java version 17.0.10
24/04/26 20:56:32 INFO ResourceUtils: =====
24/04/26 20:56:32 INFO ResourceUtils: No custom resources configured for spark.driver.
24/04/26 20:56:32 INFO ResourceUtils: =====
24/04/26 20:56:32 INFO SparkContext: Submitted application: WineQuality Training
24/04/26 20:56:32 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 4, script: , vendor: , memory -> name: memory, amount: 9486, s
cript: , vendor: , offHeap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpus -> name: cpus, amount: 1.0)
24/04/26 20:56:32 INFO ResourceProfile: Limiting resource is cpus at 4 tasks per executor
24/04/26 20:56:32 INFO ResourceProfileManager: Added ResourceProfile id: 0
24/04/26 20:56:32 INFO SecurityManager: Changing view acls to: hadoop
24/04/26 20:56:32 INFO SecurityManager: Changing modify acls to: hadoop
24/04/26 20:56:32 INFO SecurityManager: Changing view acls groups to:
24/04/26 20:56:32 INFO SecurityManager: Changing modify acls groups to:
24/04/26 20:56:32 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: hadoop; groups with view permissions: EMPTY; users with mod
ify permissions: hadoop; groups with modify permissions: EMPTY
24/04/26 20:56:32 INFO SparkEnv: Registering MapOutputTracker
24/04/26 20:56:32 INFO SparkEnv: Registering BlockManagerMaster
24/04/26 20:56:32 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
24/04/26 20:56:32 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
24/04/26 20:56:32 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
24/04/26 20:56:32 INFO DiskBlockManager: Created local directory at /mnt/tmp/blockmgr-19a6d5b6-a51f-4d1b-b115-eb9b8f2239af
24/04/26 20:56:32 INFO MemoryStore: MemoryStore started with capacity 1048.8 MiB
24/04/26 20:56:32 INFO SparkEnv: Registering OutputCommitCoordinator
24/04/26 20:56:32 INFO SubResultCacheManager: Sub-result caches are disabled.
24/04/26 20:56:32 INFO JettyUtils: Start Jetty 0.0.0.0:4040 for SparkUI
24/04/26 20:56:33 INFO Utils: Successfully started service 'SparkUI' on port 4040.
24/04/26 20:56:33 INFO Client: Will allocate AM container, with 896 MB memory including 384 MB overhead
24/04/26 20:56:33 INFO Client: Setting up container launch context for our AM
24/04/26 20:56:33 INFO Client: Connecting to ResourceManager at ip-172-31-69-133.ec2.internal/172.31.69.133:8032
24/04/26 20:56:33 INFO Configuration: resource-types.xml not found
24/04/26 20:56:33 INFO ResourceUtils: Unable to find 'resource-types.xml'.
24/04/26 20:56:33 INFO Client: Verifying our application has not requested more than the maximum memory capability of the cluster (12288 MB per container)
24/04/26 20:56:33 INFO Client: Will allocate AM container, with 896 MB memory including 384 MB overhead
24/04/26 20:56:33 INFO Client: Setting up container launch context for our AM
24/04/26 20:56:33 INFO Client: Setting up the launch environment for our AM container
24/04/26 20:56:33 INFO Client: Preparing resources for our AM container
24/04/26 20:56:33 WARN Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading libraries under SPARK_HOME.
24/04/26 20:56:34 INFO Client: Uploading resource file:/mnt/tmp/spark-4db2f66a-odd0-4955-bde0-c52a65fe4c83/_spark_libs_1278197152423226073.zip -> hdfs://ip-172-31-69-133.ec2.internal:8020
/user/hadoop/.sparkStaging/application_1714161677101_0003/_spark_libs_1278197152423226073.zip
24/04/26 20:56:35 INFO Client: Uploading resource file:/etc/spark/conf.dist/hive-site.xml -> hdfs://ip-172-31-69-133.ec2.internal:8020/user/hadoop/.sparkStaging/application_1714161677101_00
03/hive-site.xml
24/04/26 20:56:35 INFO Client: Uploading resource file:/etc/hudi/conf.dist/hudi-defaults.conf -> hdfs://ip-172-31-69-133.ec2.internal:8020/user/hadoop/.sparkStaging/application_171416167710
1_0003/hudi-defaults.conf
24/04/26 20:56:35 INFO Client: Uploading resource file:/usr/lib/spark/python/lib/pyspark.zip -> hdfs://ip-172-31-69-133.ec2.internal:8020/user/hadoop/.sparkStaging/application_171416167710
1_0003/pyspark.zip
24/04/26 20:56:35 INFO Client: Uploading resource file:/usr/lib/spark/python/lib/py4j-0.10.9.7-src.zip -> hdfs://ip-172-31-69-133.ec2.internal:8020/user/hadoop/.sparkStaging/application_171

```

```

SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
root
|-- ****fixed acidity****: string (nullable = true)
|-- ****volatile acidity****: string (nullable = true)
|-- ****citric acid****: string (nullable = true)
|-- ****residual sugar****: string (nullable = true)
|-- ****chlorides****: string (nullable = true)
|-- ****free sulfur dioxide****: string (nullable = true)
|-- ****total sulfur dioxide****: string (nullable = true)
|-- ****density****: string (nullable = true)
|-- ****pH****: string (nullable = true)
|-- ****sulphates****: string (nullable = true)
|-- ****alcohol****: string (nullable = true)
|-- ****quality****: string (nullable = true)

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|****fixed acidity****|****volatile acidity****|****citric acid****|****residual sugar****|****chlorides****|****free sulfur dioxide****|****total sulfur dioxide****|****density****|****pH****|****sulphates****|****alcohol****|****quality****|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      8.9|      0.53|      0.22|      0.48|      1.8|      0.077|      29|      60|      0.9968| |
|      7.6|      0.39|      0.31|      2.3|      0.082|      23|      71|      0.9982|
|      9.4|      0.61|      0.43|      5|      0.21|      1.6|      0.106|      10|      37|      0.9966|
|      9.1|      0.51|      0.49|      5|      0.11|      2.3|      0.084|      9|      67|      0.9968|
|      8.5|      0.53|      0.41|      5|      0.14|      2.4|      0.085|      21|      40|      0.9968|
|      6.9|      0.63|      0.7|      0.39|      0.16|      1.4|      0.08|      11|      23|      0.9955|
|      6.3|      0.56|      7.6|      0.41|      0.24|      1.8|      0.08|      4|      11|      0.9962|
|      0.59|      7.9|      0.43|      5|      0.21|      1.6|      0.106|      10|      37|      0.9966|
|      0.91|      9.5|      0.71|      0|      1.9|      0.08|      14|      35|      0.9972|
|      7.1|

```

Now, execute the following command to see the results:

grep F1 cat output.txt

The outcomes of the applied machine learning methods, including accuracy and F1 scores, are displayed below.

```
Validation Training Set Metrics
+-----+-----+
|features|label|prediction|
+-----+-----+
|[9.4,0.56,3.51,0.9978,11.0,34.0,0.076,1.9,0.0,0.7,7.4]|15.0|15.0|
|[9.8,0.68,3.2,0.9968,25.0,67.0,0.098,2.6,0.0,0.88,7.8]|15.0|15.0|
|[9.8,0.65,3.26,0.997,15.0,54.0,0.092,2.3,0.04,0.76,7.8]|15.0|15.0|
|[9.8,0.58,3.16,0.998,17.0,60.0,0.075,1.9,0.56,0.28,11.2]|16.0|15.0|
|[9.4,0.56,3.51,0.9978,11.0,34.0,0.076,1.9,0.0,0.7,7.4]|15.0|15.0|
+-----+-----+
only showing top 5 rows

The accuracy of the model is 0.575
F1: 0.5619407071339173

Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
TestingDataSet Metrics
+-----+-----+
|features|label|prediction|
+-----+-----+
|[9.4,0.56,3.51,0.9978,11.0,34.0,0.076,1.9,0.0,0.7,7.4]|15|15.0|
|[9.8,0.68,3.2,0.9968,25.0,67.0,0.098,2.6,0.0,0.88,7.8]|15|15.0|
|[9.8,0.65,3.26,0.997,15.0,54.0,0.092,2.3,0.04,0.76,7.8]|15|15.0|
|[9.8,0.58,3.16,0.998,17.0,60.0,0.075,1.9,0.56,0.28,11.2]|16|15.0|
|[9.4,0.56,3.51,0.9978,11.0,34.0,0.076,1.9,0.0,0.7,7.4]|15|15.0|
+-----+-----+
only showing top 5 rows

The accuracy of the model is 0.6271186440677966
F1: 0.593151718932272
```

DOCKER IMPLEMENTATION –

Steps for Installing Docker:

sudo yum update -y

sudo service docker start

start docker.service

```
[root@ip-172-31-5-86 ~]# sudo yum update -y
Last metadata expiration check: 2:32:04 ago on Sat Apr 27 17:28:59 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-5-86 ~]# sudo yum install -y docker
Last metadata expiration check: 2:32:18 ago on Sat Apr 27 17:28:59 2024.
Package docker-25.0.3-1.amzn2023.0.1.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-5-86 ~]# sudo service docker start
Redirecting to /bin/systemctl start docker.service
[root@ip-172-31-5-86 ~]# sudo service docker status
Redirecting to /bin/systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; preset: disabled)
   Active: active (running) since Sat 2024-04-27 20:01:36 UTC; 51s ago
 TriggeredBy: ● docker.socket
   Docs: https://docs.docker.com
  Process: 104192 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
  Process: 104193 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
 Main PID: 104194 (dockerd)
    Tasks: 10
   Memory: 108.8M
      CPU: 367ms
  CGroup: /system.slice/docker.service
          └─104194 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

Apr 27 20:01:35 ip-172-31-5-86.ec2.internal systemd[1]: Starting docker.service - Docker Application Container Engine...
Apr 27 20:01:36 ip-172-31-5-86.ec2.internal dockerd[104194]: time="2024-04-27T20:01:36.240827520Z" level=info msg="Starting up"
```

```

[root@ip-172-31-5-86 ~]# sudo service docker start
Redirecting to /bin/systemctl start docker.service
[root@ip-172-31-5-86 ~]# sudo service docker status
Redirecting to /bin/systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; preset: disabled)
   Active: active (running) since Sat 2024-04-27 20:01:36 UTC; 51s ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Process: 104192 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
    Process: 104193 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
   Main PID: 104194 (dockerd)
      Tasks: 10
     Memory: 108.8M
        CPU: 367ms
    CGroup: /system.slice/docker.service
            └─104194 /usr/bin/dockerd -B fd:/// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

Apr 27 20:01:35 ip-172-31-5-86.ec2.internal systemd[1]: Starting docker.service - Docker Application Container Engine...
Apr 27 20:01:36 ip-172-31-5-86.ec2.internal dockerd[104194]: time="2024-04-27T20:01:36.240827520Z" level=info msg="Starting up"
Apr 27 20:01:36 ip-172-31-5-86.ec2.internal dockerd[104194]: time="2024-04-27T20:01:36.458407840Z" level=info msg="Loading containers: start."
Apr 27 20:01:36 ip-172-31-5-86.ec2.internal dockerd[104194]: time="2024-04-27T20:01:36.847155017Z" level=info msg="Loading containers: done."
Apr 27 20:01:36 ip-172-31-5-86.ec2.internal dockerd[104194]: time="2024-04-27T20:01:36.910102916Z" level=info msg="Docker daemon" commit=f417435 containerd-snapshotter=false s
Apr 27 20:01:36 ip-172-31-5-86.ec2.internal dockerd[104194]: time="2024-04-27T20:01:36.910535531Z" level=info msg="Daemon has completed initialization"
Apr 27 20:01:36 ip-172-31-5-86.ec2.internal dockerd[104194]: time="2024-04-27T20:01:36.963477443Z" level=info msg="API listen on /run/docker.sock"
Apr 27 20:01:36 ip-172-31-5-86.ec2.internal systemd[1]: Started docker.service - Docker Application Container Engine.

```

To create a Docker container image from the provided Dockerfile, execute the following command in your terminal while in the same directory as the Dockerfile.

sudo docker build -t sravyakganti/cs643-programming-assignment-2 .

Executing the provided command will initiate the process of constructing a Docker image based on the instructions specified in the Dockerfile, and the resulting image will be locally stored and available within your EC2 instance.

After attempting to build the Docker image, you can verify its successful creation by running the following command: **sudo docker image ls**. This command will display a list of all Docker images present on your EC2 instance, allowing you to confirm the existence of the newly built image among the listed images.

```

[root@ip-172-31-5-86 CS643_Programming_assignment_2]# docker image ls
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
sk3568/cs643-programming-assignment-2  latest     2d7b4954c93e  3 minutes ago  2.42GB

```

To upload and store the Docker image you've created on the Docker Hub repository, execute the following instruction in your terminal or command prompt.

Run the following command to launch this Docker image:

sudo docker run -it sravyakganti/cs643-programming-assignment-2

Here, you have the option to utilize your image ID in place of the image name.

sudo docker run -it <IMAGE_ID>

```

TestingDataSet Metrics
+-----+-----+
|features|label|prediction|
+-----+-----+
| [9.4,0.56,3.51,0.9978,11.0,34.0,0.076,1.9,0.0,0.7,7.4] | 15.0 | 15.0 |
| [9.8,0.68,3.2,0.9968,25.0,67.0,0.098,2.6,0.0,0.88,7.8] | 15.0 | 15.0 |
| [9.8,0.65,3.26,0.997,15.0,54.0,0.092,2.3,0.04,0.76,7.8] | 15.0 | 15.0 |
| [9.8,0.58,3.16,0.998,17.0,60.0,0.075,1.9,0.56,0.28,11.2] | 16.0 | 15.0 |
| [9.4,0.56,3.51,0.9978,11.0,34.0,0.076,1.9,0.0,0.7,7.4] | 15.0 | 15.0 |
+-----+-----+
only showing top 5 rows

The accuracy of the model is 0.6271186440677966
F1: 0.593151718932272

```

Login into Docker Hub credentials in terminal

docker login

sudo docker push sravyakganti/cs643-programming-assignment-2


```
[root@ip-172-31-5-86 CS643_Programming_assignment_2]# sudo docker push sravyakganti/cs643-programming-assignment-2
Using default tag: latest
The push refers to repository [docker.io/sravyakganti/cs643-programming-assignment-2]
65a5ca627212: Pushed
5fee2b7a15e4: Pushed
99cf951636f1: Pushed
36cd5c873f47: Pushed
dlc09b8eac72: Pushed
57c651240c9f: Mounted from guoxiaojun2/spark-3.2.2-bin-hadoop3.2
553c43e260d1: Mounted from guoxiaojun2/spark-3.2.2-bin-hadoop3.2
36ef902c4c66: Mounted from guoxiaojun2/spark-3.2.2-bin-hadoop3.2
eal1b88bc1ff8: Mounted from guoxiaojun2/spark-3.2.2-bin-hadoop3.2
632ccc24d10f: Mounted from guoxiaojun2/spark-3.2.2-bin-hadoop3.2
8933d669b084: Mounted from guoxiaojun2/spark-3.2.2-bin-hadoop3.2
367158596a5c: Mounted from guoxiaojun2/spark-3.2.2-bin-hadoop3.2
c9ac6abbc04d: Mounted from guoxiaojun2/spark-3.2.2-bin-hadoop3.2
41caa71c39b5: Mounted from guoxiaojun2/spark-3.2.2-bin-hadoop3.2
97393f8c8163: Mounted from guoxiaojun2/spark-3.2.2-bin-hadoop3.2
d7802b8508af: Mounted from guoxiaojun2/spark-3.2.2-bin-hadoop3.2
e3abdc2e9252: Mounted from guoxiaojun2/spark-3.2.2-bin-hadoop3.2
eafe6e032dbd: Mounted from guoxiaojun2/spark-3.2.2-bin-hadoop3.2
92a4e8a3140f: Mounted from guoxiaojun2/spark-3.2.2-bin-hadoop3.2
latest: digest: sha256:dc0b32b44b6d3574124be025252be6636a45f50bc3655b6f768436f48ef2315b size: 4516
```

Now that you have the docker image downloaded from the DockerHub repository, you may start it by following the directions on the DockerHub website.