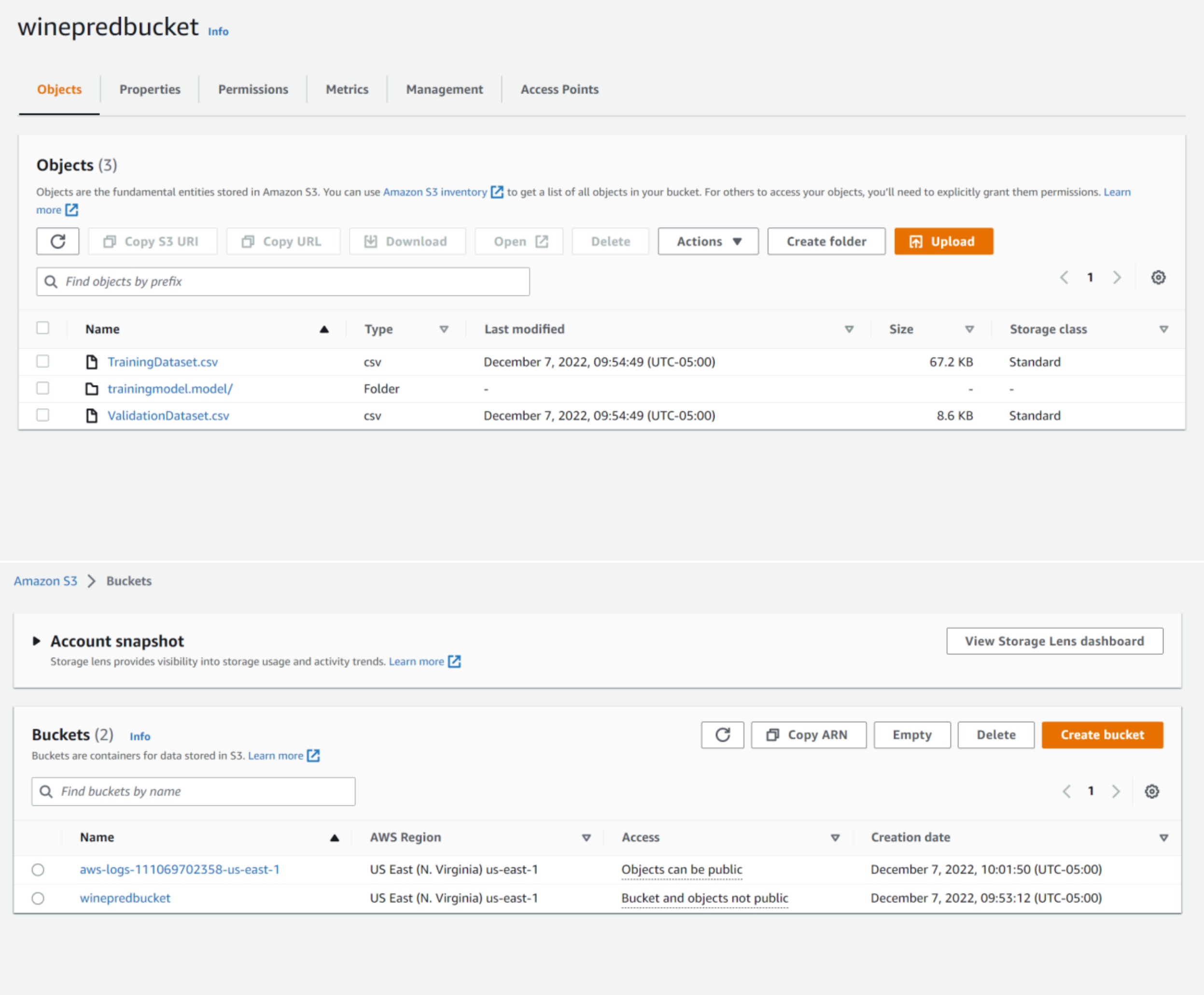
CS 643, CLOUD COMPUTING
PROGRAMMING ASSIGNMENT 2

GitHub Link: https://github.com/keerthikalla/kk224-programming-assignment-2

Docker Repository: https://hub.docker.com/repository/docker/keerthikalla123/winequalitypred

Input for model training:

Create an S3 bucket to upload the training and validation datasets



Objects (3)

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

Name	Type	Last modified	Size	Storage class
TrainingDataset.csv	csv	December 7, 2022, 09:54:49 (UTC-05:00)	67.2 KB	Standard
trainingmodel.model/	Folder	-	-	-
ValidationDataset.csv	csv	December 7, 2022, 09:54:49 (UTC-05:00)	8.6 KB	Standard

Account snapshot

Storage lens provides visibility into storage usage and activity trends. Learn more

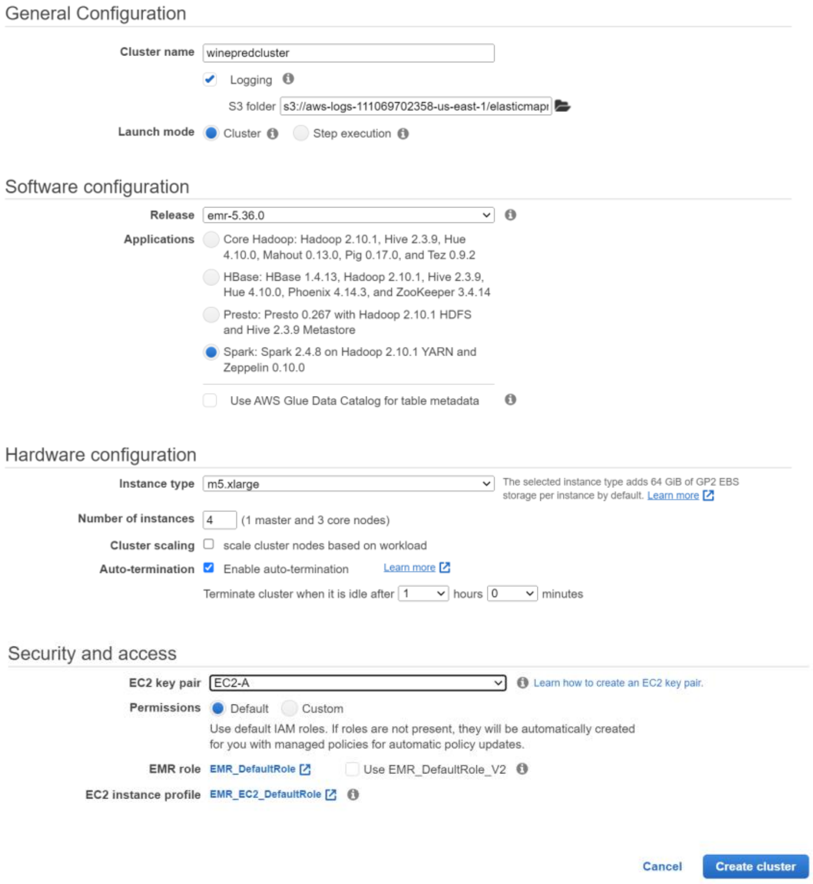
Buckets (2) Info

Buckets are containers for data stored in S3. Learn more

Name	AWS Region	Access	Creation date
aws-logs-111069702358-us-east-1	US East (N. Virginia) us-east-1	Objects can be public	December 7, 2022, 10:01:50 (UTC-05:00)
winepredbucket	US East (N. Virginia) us-east-1	Bucket and objects not public	December 7, 2022, 09:53:12 (UTC-05:00)

Setting up 4 parallel EC2 instances:

1. Launch an EMR cluster with following specifications:



General Configuration

Cluster name: wineepredcluster

☑ Logging

S3 folder: s3://aws-logs-111069702358-us-east-1/elasticmapr

Launch mode: ● Cluster ○ Step execution

Software configuration

Release: emr-5.36.0

Applications:
○ Core Hadoop: Hadoop 2.10.1, Hive 2.3.9, Hue 4.10.0, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.2
○ HBase: HBase 1.4.13, Hadoop 2.10.1, Hive 2.3.9, Hue 4.10.0, Phoenix 4.14.3, and ZooKeeper 3.4.14
○ Presto: Presto 0.267 with Hadoop 2.10.1 HDFS and Hive 2.3.9 Metastore
● Spark: Spark 2.4.8 on Hadoop 2.10.1 YARN and Zeppelin 0.10.0

□ Use AWS Glue Data Catalog for table metadata

Hardware configuration

Instance type: m5.xlarge — The selected instance type adds 64 GiB of GP2 EBS storage per instance by default. Learn more

Number of instances: 4 (1 master and 3 core nodes)

Cluster scaling: □ scale cluster nodes based on workload

Auto-termination: ☑ Enable auto-termination Learn more

Terminate cluster when it is idle after 1 hours 0 minutes

Security and access

EC2 key pair: EC2-A — Learn how to create an EC2 key pair.

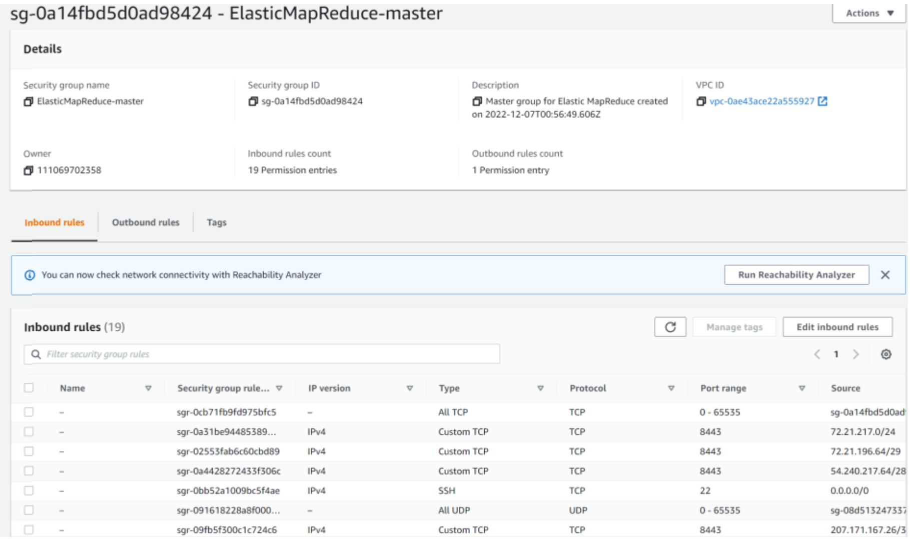
Permissions: ● Default ○ Custom

Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR role: EMR_DefaultRole □ Use EMR_DefaultRole_V2

EC2 instance profile: EMR_EC2_DefaultRole

2. Go to security groups of EMR-Master to edit the inbound rules to allow ssh to be done.



sg-0a14fbd5d0ad98424 - ElasticMapReduce-master

Details

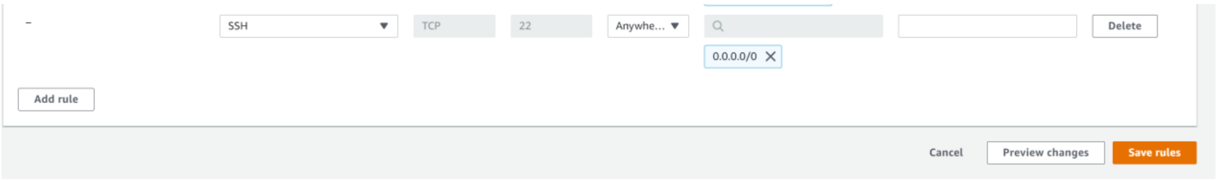
Security group name ElasticMapReduce-master	Security group ID sg-0a14fbd5d0ad98424	Description Master group for Elastic MapReduce created on 2022-12-07T00:56:49.606Z	VPC ID vpc-0ae43ace22a555927
Owner 111069702358	Inbound rules count 19 Permission entries	Outbound rules count 1 Permission entry	

You can now check network connectivity with Reachability Analyzer

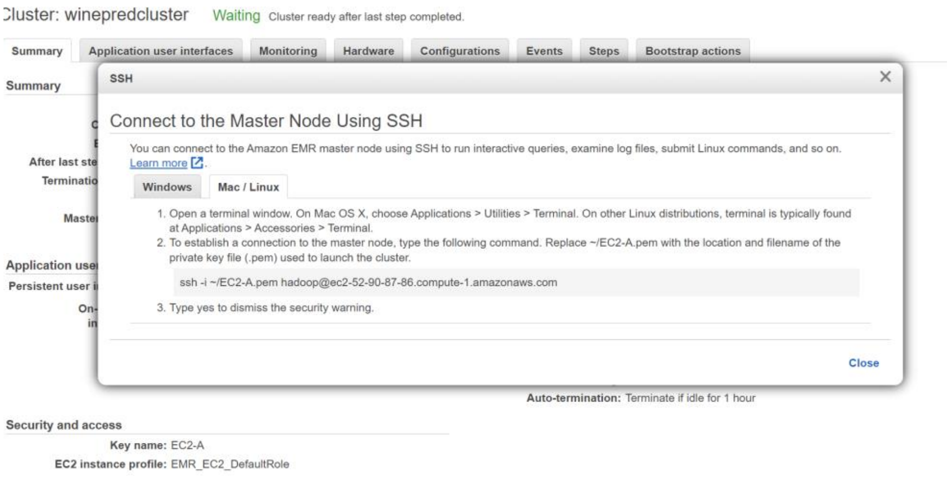
Inbound rules (19)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source
–	sgr-0cb71fb9fd975bfc5	–	All TCP	TCP	0 - 65535	sg-0a14fbd5d0ad
–	sgr-0a31be94485389...	IPv4	Custom TCP	TCP	8443	72.21.217.0/24
–	sgr-02553fab6c60cbd89	IPv4	Custom TCP	TCP	8443	72.21.196.64/29
–	sgr-0a4428272433f306c	IPv4	Custom TCP	TCP	8443	54.240.217.64/28
–	sgr-0bb52a1009bc5f4ae	IPv4	SSH	TCP	22	0.0.0.0/0
–	sgr-091618228a8f000...	–	All UDP	UDP	0 - 65535	sg-08d51324733
–	sgr-09fb5f300c1c724c6	IPv4	Custom TCP	TCP	8443	207.171.167.26/3

3. Click on edit inbound rules and add the following rule and click save.



4. Connect to the Master Node using SSH as follows:



Cluster: winepredcluster Waiting Cluster ready after last step completed.

Connect to the Master Node Using SSH

You can connect to the Amazon EMR master node using SSH to run interactive queries, examine log files, submit Linux commands, and so on. Learn more.

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 master node, type the following command. Replace ~/EC2-A.pem with the location and filename of the private key file (.pem) used to launch the cluster.

```
ssh -i ~/EC2-A.pem hadoop@ec2-52-90-87-86.compute-1.amazonaws.com
```

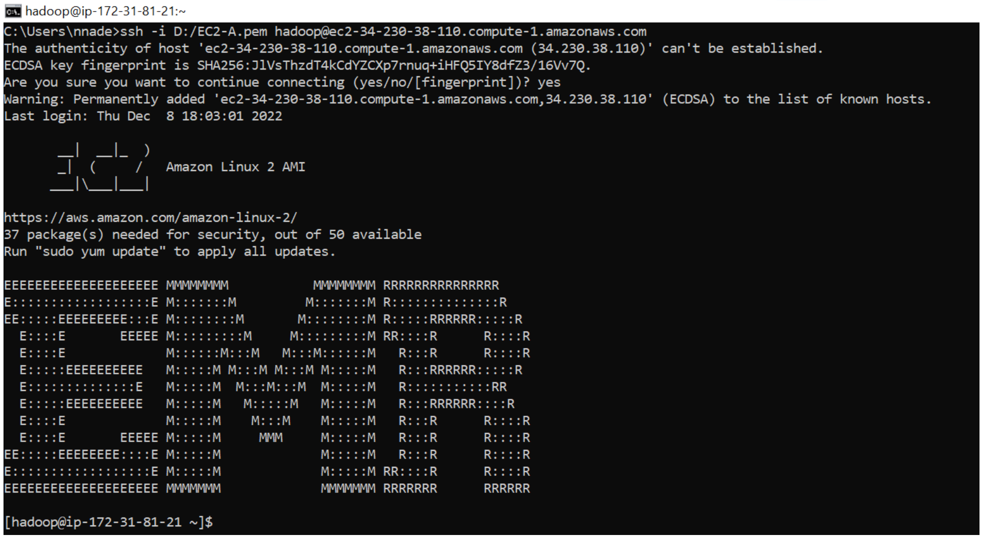
3. Type yes to dismiss the security warning.

Auto-termination: Terminate if idle for 1 hour

Security and access

Key name: EC2-A

EC2 instance profile: EMR_EC2_DefaultRole



hadoop@ip-172-31-81-21:~

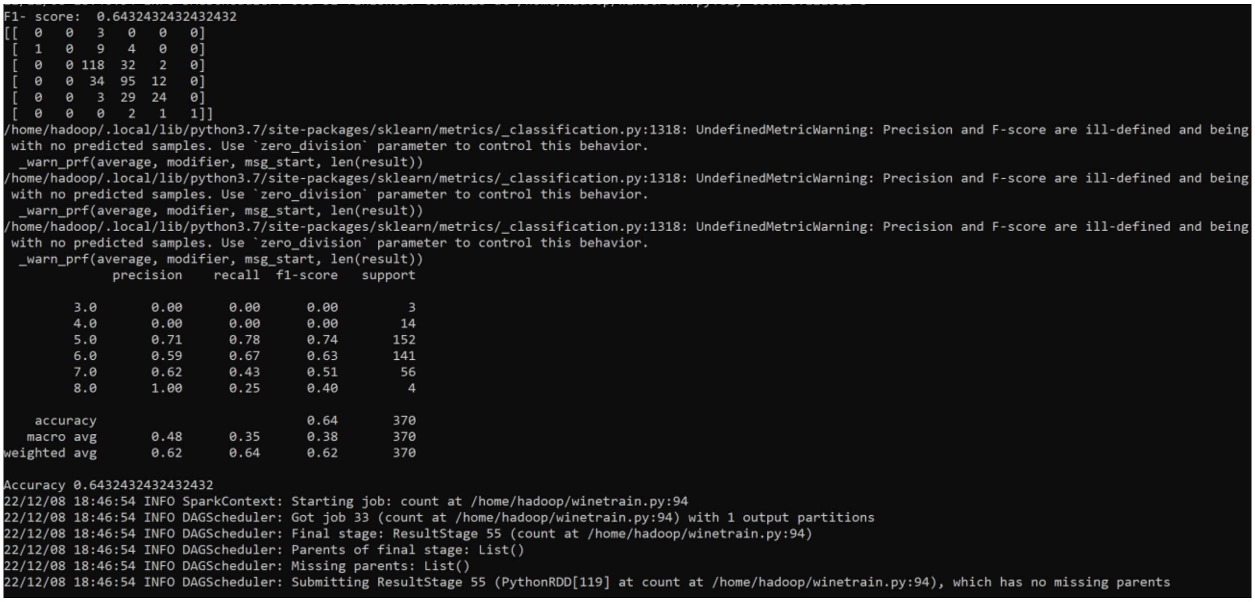
```
C:\Users\nnade>ssh -i D:/EC2-A.pem hadoop@ec2-34-230-38-110.compute-1.amazonaws.com
The authenticity of host 'ec2-34-230-38-110.compute-1.amazonaws.com (34.230.38.110)' can't be established.
ECDSA key fingerprint is SHA256:JlVsThzdT4kCdYZCXp7rnuq+iHFQ5IY8dfZ3/16Vv7Q.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-230-38-110.compute-1.amazonaws.com,34.230.38.110' (ECDSA) to the list of known hosts.
Last login: Thu Dec  8 18:03:01 2022

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
37 package(s) needed for security, out of 50 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::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 RRRRRRR      RRRRRR

[hadoop@ip-172-31-81-21 ~]$
```

5. Create a .py file using the command “nano train.py”.

Then write your code, save, and close by using the following command: Shift O + Enter + Shift X.

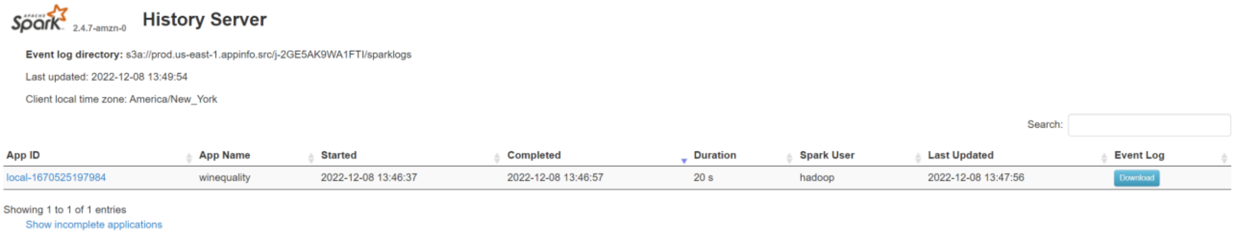
6. Install all necessary libraries using the “pip install <libraryname>” command.

7. Run the code using ‘”spark-submit train.py”



```
F1- score:  0.6432432432432432
[[  0   0   3   0   0   0]
 [  1   0   9   4   0   0]
 [  0   0 118  32   2   0]
 [  0   0  34  95  12   0]
 [  0   0   3  29  24   0]
 [  0   0   0   2   1   1]]
/home/hadoop/.local/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being
 with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/hadoop/.local/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being
 with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/hadoop/.local/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being
 with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

         3.0       0.00      0.00      0.00         3
         4.0       0.00      0.00      0.00        14
         5.0       0.71      0.78      0.74       152
         6.0       0.59      0.67      0.63       141
         7.0       0.62      0.43      0.51        56
         8.0       1.00      0.25      0.40         4

    accuracy                           0.64       370
   macro avg       0.48      0.35      0.38       370
weighted avg       0.62      0.64      0.62       370

Accuracy 0.6432432432432432
22/12/08 18:46:54 INFO SparkContext: Starting job: count at /home/hadoop/winetrain.py:94
22/12/08 18:46:54 INFO DAGScheduler: Got job 33 (count at /home/hadoop/winetrain.py:94) with 1 output partitions
22/12/08 18:46:54 INFO DAGScheduler: Final stage: ResultStage 55 (count at /home/hadoop/winetrain.py:94)
22/12/08 18:46:54 INFO DAGScheduler: Parents of final stage: List()
22/12/08 18:46:54 INFO DAGScheduler: Missing parents: List()
22/12/08 18:46:54 INFO DAGScheduler: Submitting ResultStage 55 (PythonRDD[119] at count at /home/hadoop/winetrain.py:94), which has no missing parents
```

8. Go to the application interface and select spark history server.



History Server

Event log directory: s3a://prod.us-east-1.appinfo.src/j-2GE5AK9WA1FTI/sparklogs

Last updated: 2022-12-08 13:49:54

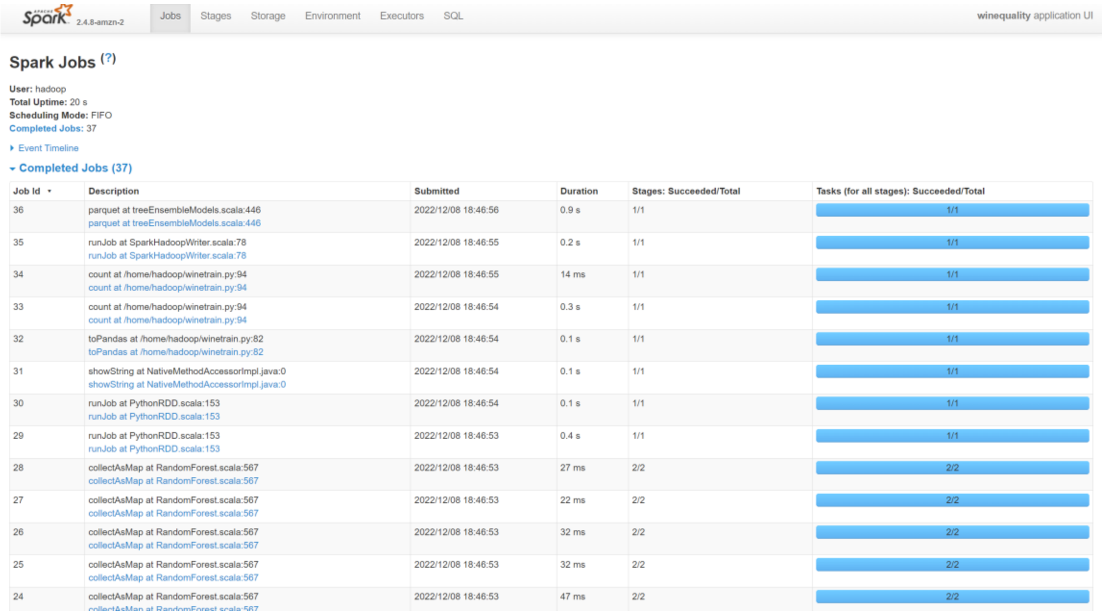
Client local time zone: America/New_York

App ID	App Name	Started	Completed	Duration	Spark User	Last Updated	Event Log
local-1670525197984	winequality	2022-12-08 13:46:37	2022-12-08 13:46:57	20 s	hadoop	2022-12-08 13:47:56	Download

Showing 1 to 1 of 1 entries

Show incomplete applications

9. Check status of your job.



Spark Jobs (?)

User: hadoop
Total Uptime: 20 s
Scheduling Mode: FIFO
Completed Jobs: 37

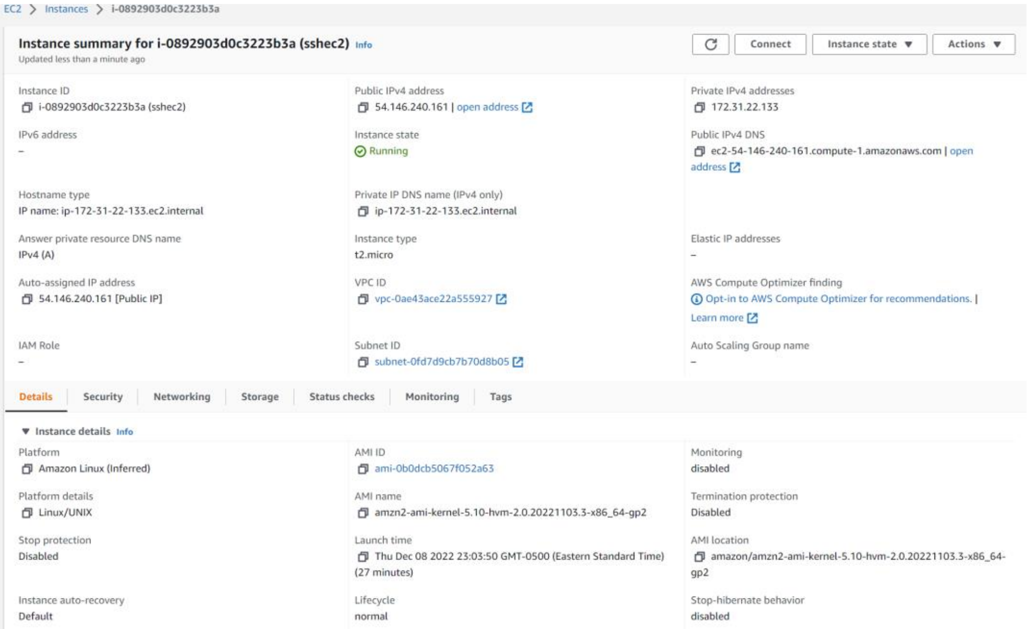
▸ Event Timeline

▾ Completed Jobs (37)

Job Id ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
36	parquet at treeEnsembleModels.scala:446 parquet at treeEnsembleModels.scala:446	2022/12/08 18:46:56	0.9 s	1/1	1/1
35	runJob at SparkHadoopWriter.scala:78 runJob at SparkHadoopWriter.scala:78	2022/12/08 18:46:55	0.2 s	1/1	1/1
34	count at /home/hadoop/winetrain.py:94 count at /home/hadoop/winetrain.py:94	2022/12/08 18:46:55	14 ms	1/1	1/1
33	count at /home/hadoop/winetrain.py:94 count at /home/hadoop/winetrain.py:94	2022/12/08 18:46:54	0.3 s	1/1	1/1
32	toPandas at /home/hadoop/winetrain.py:82 toPandas at /home/hadoop/winetrain.py:82	2022/12/08 18:46:54	0.1 s	1/1	1/1
31	showString at NativeMethodAccessorImpl.java:0 showString at NativeMethodAccessorImpl.java:0	2022/12/08 18:46:54	0.1 s	1/1	1/1
30	runJob at PythonRDD.scala:153 runJob at PythonRDD.scala:153	2022/12/08 18:46:54	0.1 s	1/1	1/1
29	runJob at PythonRDD.scala:153 runJob at PythonRDD.scala:153	2022/12/08 18:46:53	0.4 s	1/1	1/1
28	collectAsMap at RandomForest.scala:567 collectAsMap at RandomForest.scala:567	2022/12/08 18:46:53	27 ms	2/2	2/2
27	collectAsMap at RandomForest.scala:567 collectAsMap at RandomForest.scala:567	2022/12/08 18:46:53	22 ms	2/2	2/2
26	collectAsMap at RandomForest.scala:567 collectAsMap at RandomForest.scala:567	2022/12/08 18:46:53	32 ms	2/2	2/2
25	collectAsMap at RandomForest.scala:567 collectAsMap at RandomForest.scala:567	2022/12/08 18:46:53	32 ms	2/2	2/2
24	collectAsMap at RandomForest.scala:567 collectAsMap at RandomForest.scala:567	2022/12/08 18:46:53	47 ms	2/2	2/2

<u>Developing a Spark Application (Model Implementation):</u>

1. Launch an EC2 instance as follows



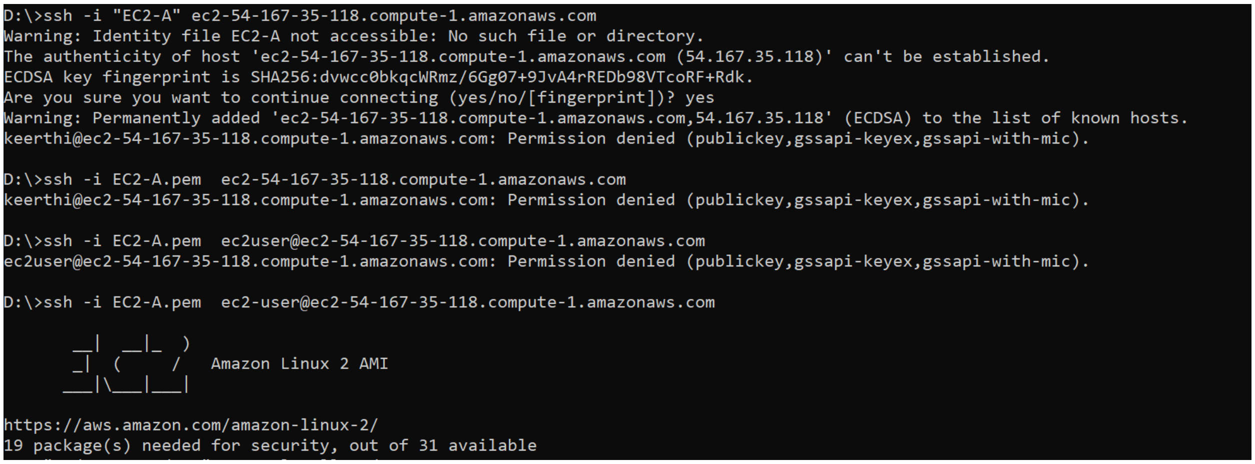
Instance summary for i-0892903d0c3223b3a (sshec2) Info
Updated less than a minute ago

Instance ID: i-0892903d0c3223b3a (sshec2)
Public IPv4 address: 54.146.240.161 | open address
Private IPv4 addresses: 172.31.22.133
IPv6 address: –
Instance state: Running
Public IPv4 DNS: ec2-54-146-240-161.compute-1.amazonaws.com | open address
Hostname type: IP name: ip-172-31-22-133.ec2.internal
Private IP DNS name (IPv4 only): ip-172-31-22-133.ec2.internal
Answer private resource DNS name: IPv4 (A)
Instance type: t2.micro
Elastic IP addresses: –
Auto-assigned IP address: 54.146.240.161 [Public IP]
VPC ID: vpc-0ae43ace22a555927
AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role: –
Subnet ID: subnet-0fd7d9cb7b70d8b05
Auto Scaling Group name: –

▼ Instance details Info

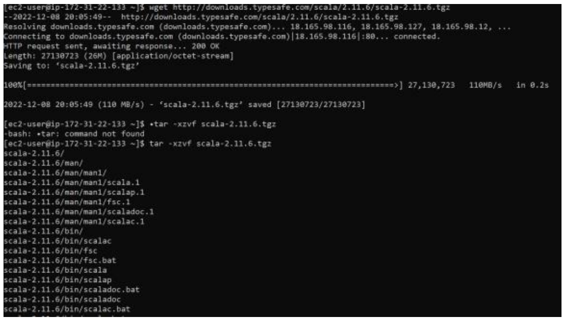
Platform: Amazon Linux (Inferred)
AMI ID: ami-0b0dcb5067f052a63
Monitoring: disabled
Platform details: Linux/UNIX
AMI name: amzn2-ami-kernel-5.10-hvm-2.0.20221103.3-x86_64-gp2
Termination protection: Disabled
Stop protection: Disabled
Launch time: Thu Dec 08 2022 23:03:50 GMT-0500 (Eastern Standard Time) (27 minutes)
AMI location: amazon/amzn2-ami-kernel-5.10-hvm-2.0.20221103.3-x86_64-gp2
Instance auto-recovery: Default
Lifecycle: normal
Stop-hibernate behavior: disabled

2. Connect to your new instance from the local terminal



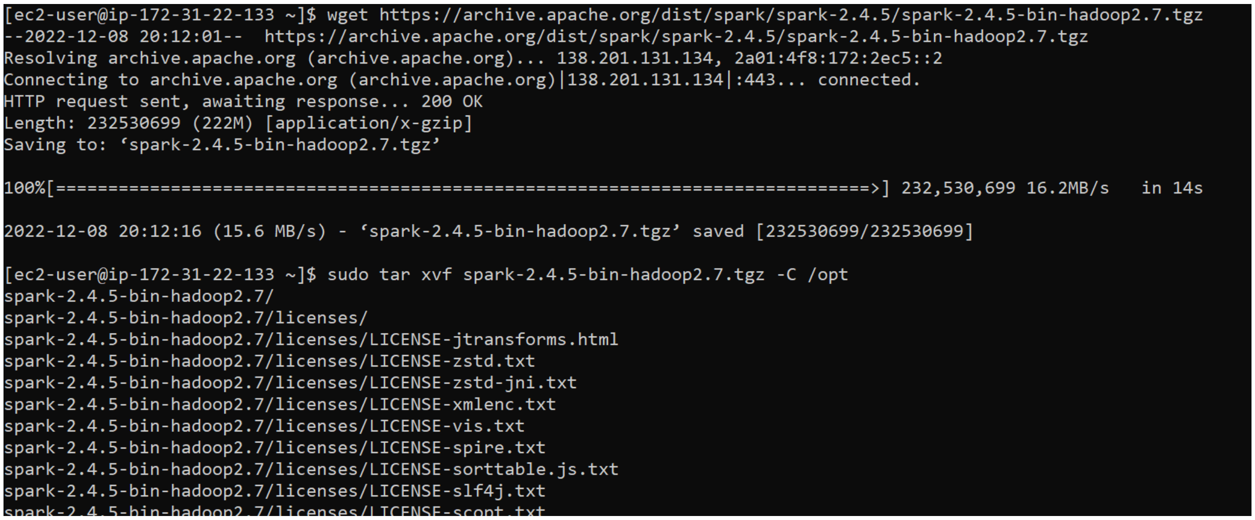
```
D:\>ssh -i "EC2-A" ec2-54-167-35-118.compute-1.amazonaws.com
Warning: Identity file EC2-A not accessible: No such file or directory.
The authenticity of host 'ec2-54-167-35-118.compute-1.amazonaws.com (54.167.35.118)' can't be established.
ECDSA key fingerprint is SHA256:dvwcc0bkqcWRmz/6Gg07+9JvA4rREDb98VTcoRF+Rdk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-167-35-118.compute-1.amazonaws.com,54.167.35.118' (ECDSA) to the list of known hosts.
keerthi@ec2-54-167-35-118.compute-1.amazonaws.com: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).

D:\>ssh -i EC2-A.pem  ec2-54-167-35-118.compute-1.amazonaws.com
keerthi@ec2-54-167-35-118.compute-1.amazonaws.com: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).

D:\>ssh -i EC2-A.pem  ec2user@ec2-54-167-35-118.compute-1.amazonaws.com
ec2user@ec2-54-167-35-118.compute-1.amazonaws.com: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).

D:\>ssh -i EC2-A.pem  ec2-user@ec2-54-167-35-118.compute-1.amazonaws.com

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
19 package(s) needed for security, out of 31 available
```

3. Install Scala



```
[ec2-user@ip-172-31-22-133 ~]$ wget http://downloads.typesafe.com/scala/2.11.6/scala-2.11.6.tgz
--2022-12-08 20:05:49--  http://downloads.typesafe.com/scala/2.11.6/scala-2.11.6.tgz
Resolving downloads.typesafe.com (downloads.typesafe.com)... 18.165.98.116, 18.165.98.127, 18.165.98.12, ...
Connecting to downloads.typesafe.com (downloads.typesafe.com)|18.165.98.116|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 27130723 (26M) [application/octet-stream]
Saving to: ‘scala-2.11.6.tgz’

100%[==========================================================================>] 27,130,723   110MB/s   in 0.2s

2022-12-08 20:05:49 (110 MB/s) - ‘scala-2.11.6.tgz’ saved [27130723/27130723]

[ec2-user@ip-172-31-22-133 ~]$ •tar -xzvf scala-2.11.6.tgz
-bash: •tar: command not found
[ec2-user@ip-172-31-22-133 ~]$ tar -xzvf scala-2.11.6.tgz
scala-2.11.6/
scala-2.11.6/man/
scala-2.11.6/man/man1/
scala-2.11.6/man/man1/scala.1
scala-2.11.6/man/man1/scalap.1
scala-2.11.6/man/man1/fsc.1
scala-2.11.6/man/man1/scaladoc.1
scala-2.11.6/man/man1/scalac.1
scala-2.11.6/bin/
scala-2.11.6/bin/scalac
scala-2.11.6/bin/fsc
scala-2.11.6/bin/fsc.bat
scala-2.11.6/bin/scala
scala-2.11.6/bin/scalap
scala-2.11.6/bin/scaladoc.bat
scala-2.11.6/bin/scaladoc
scala-2.11.6/bin/scalac.bat
```

4. Install Spark

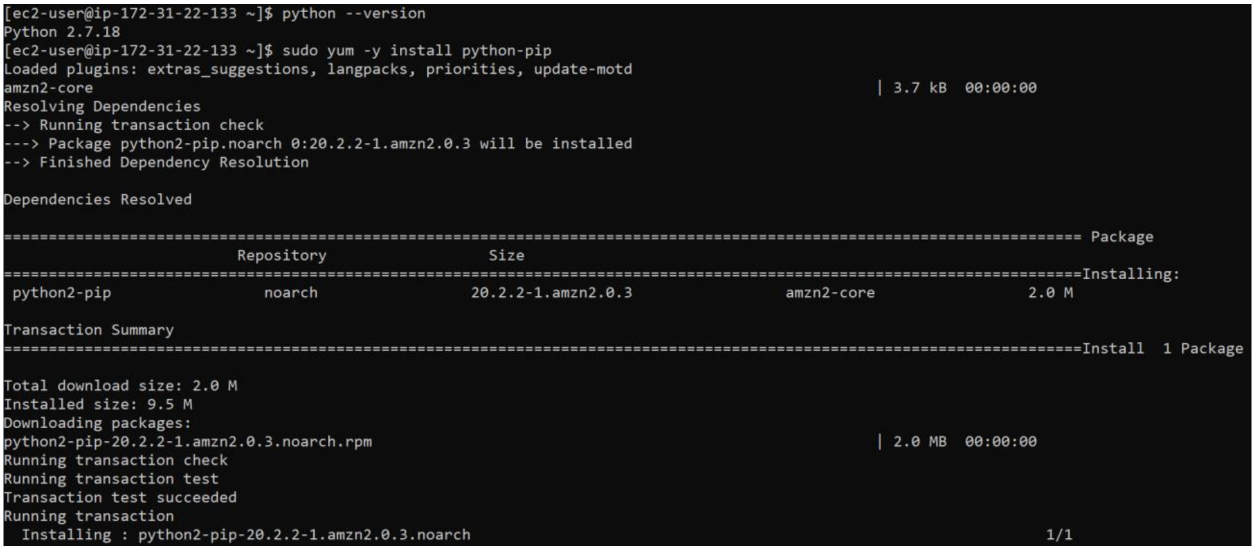


```
[ec2-user@ip-172-31-22-133 ~]$ wget https://archive.apache.org/dist/spark/spark-2.4.5/spark-2.4.5-bin-hadoop2.7.tgz
--2022-12-08 20:12:01--  https://archive.apache.org/dist/spark/spark-2.4.5/spark-2.4.5-bin-hadoop2.7.tgz
Resolving archive.apache.org (archive.apache.org)... 138.201.131.134, 2a01:4f8:172:2ec5::2
Connecting to archive.apache.org (archive.apache.org)|138.201.131.134|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 232530699 (222M) [application/x-gzip]
Saving to: ‘spark-2.4.5-bin-hadoop2.7.tgz’

100%[==========================================================================>] 232,530,699 16.2MB/s   in 14s

2022-12-08 20:12:16 (15.6 MB/s) - ‘spark-2.4.5-bin-hadoop2.7.tgz’ saved [232530699/232530699]

[ec2-user@ip-172-31-22-133 ~]$ sudo tar xvf spark-2.4.5-bin-hadoop2.7.tgz -C /opt
spark-2.4.5-bin-hadoop2.7/
spark-2.4.5-bin-hadoop2.7/licenses/
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-jtransforms.html
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-zstd.txt
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-zstd-jni.txt
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-xmlenc.txt
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-vis.txt
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-spire.txt
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-sorttable.js.txt
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-slf4j.txt
spark-2.4.5-bin-hadoop2.7/licenses/LICENSE-scopt.txt
```

5. Check Java -version. In case it doesn’t exist, follow the instructions provided in this link.

https://techviewleo.com/install-java-openjdk-on-amazon-linux-system/

6. Check python version



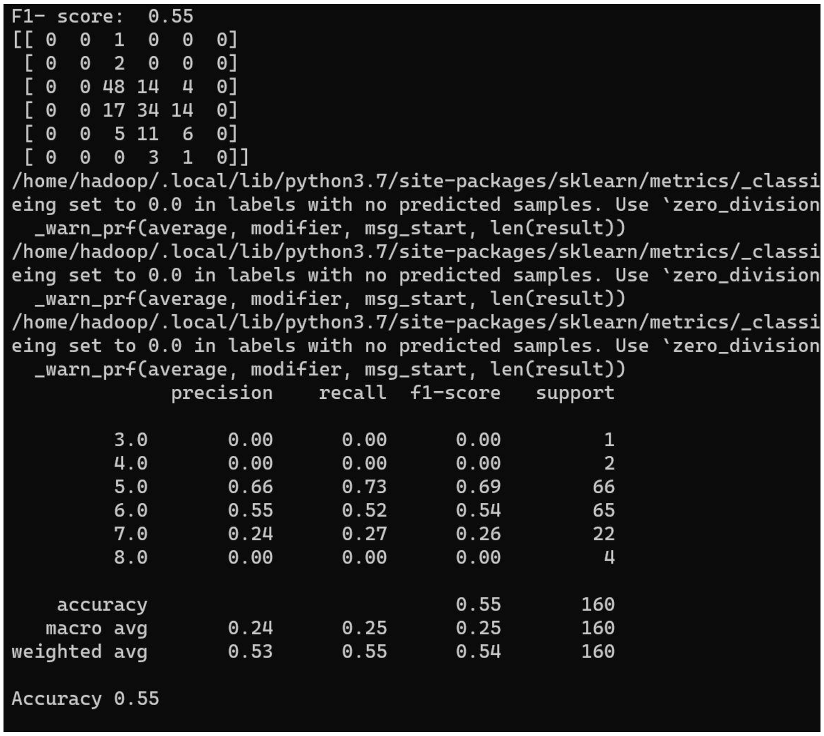
```
[ec2-user@ip-172-31-22-133 ~]$ python --version
Python 2.7.18
[ec2-user@ip-172-31-22-133 ~]$ sudo yum -y install python-pip
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                                                       | 3.7 kB  00:00:00
Resolving Dependencies
--> Running transaction check
---> Package python2-pip.noarch 0:20.2.2-1.amzn2.0.3 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

======================================================================================================================== Package
                          Repository                  Size
========================================================================================================================Installing:
 python2-pip                 noarch                 20.2.2-1.amzn2.0.3                 amzn2-core                 2.0 M

Transaction Summary
========================================================================================================================Install  1 Package

Total download size: 2.0 M
Installed size: 9.5 M
Downloading packages:
python2-pip-20.2.2-1.amzn2.0.3.noarch.rpm                                                        | 2.0 MB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : python2-pip-20.2.2-1.amzn2.0.3.noarch                                                                 1/1
```

7. Create a .py file using the command “nano test.py”.

Then write your code, save, and close by using the following command: Shift O + Enter + Shift X.

8. Install all necessary libraries using the “pip install <libraryname>” command.

9. Run the code using ‘”spark-submit test.py”



```
F1- score:  0.55
[[ 0  0  1  0  0  0]
 [ 0  0  2  0  0  0]
 [ 0  0 48 14  4  0]
 [ 0  0 17 34 14  0]
 [ 0  0  5 11  6  0]
 [ 0  0  0  3  1  0]]
/home/hadoop/.local/lib/python3.7/site-packages/sklearn/metrics/_classi
eing set to 0.0 in labels with no predicted samples. Use `zero_division
  _warn_prf(average, modifier, msg_start, len(result))
/home/hadoop/.local/lib/python3.7/site-packages/sklearn/metrics/_classi
eing set to 0.0 in labels with no predicted samples. Use `zero_division
  _warn_prf(average, modifier, msg_start, len(result))
/home/hadoop/.local/lib/python3.7/site-packages/sklearn/metrics/_classi
eing set to 0.0 in labels with no predicted samples. Use `zero_division
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

         3.0       0.00      0.00      0.00         1
         4.0       0.00      0.00      0.00         2
         5.0       0.66      0.73      0.69        66
         6.0       0.55      0.52      0.54        65
         7.0       0.24      0.27      0.26        22
         8.0       0.00      0.00      0.00         4

    accuracy                           0.55       160
   macro avg       0.24      0.25      0.25       160
weighted avg       0.53      0.55      0.54       160

Accuracy 0.55
```

<u>Predicting using Docker Images:</u>

Launch your ec2-instance and then step-up docker as follows:

1. Go to your Docker repository

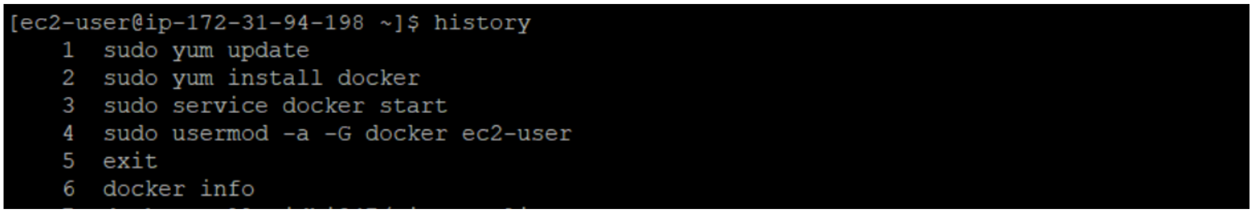
2. Pull the image to Docker hub repository by using the following command

"docker pull keerthikalla123/winequalitypred:tag"

3. Run the image using the following command

"docker run -p 4000:80 keerthikalla123/winequalitypred:tag"

Accuracy and F1 score will be displayed accordingly.



```
[ec2-user@ip-172-31-94-198 ~]$ history
    1  sudo yum update
    2  sudo yum install docker
    3  sudo service docker start
    4  sudo usermod -a -G docker ec2-user
    5  exit
    6  docker info
```