CS643 - Programming Assignment 2

Docker link :
https://hub.docker.com/repository/docker/mg946/wine_quality/general

Github Repository link :
https://github.com/MohanKumar946/Wine_Quality_CS643

Wine Quality Prediction Using Spark and Amazon Web Services (AWS)

- The procedure to use AWS services to train ML (Machine Learning) models on multiple parallel EC2(Elastic Compute Cloud) instances using EMR cluster.

- The ML program is written in python using Apache Spark MLlib libraries. The training and prediction programs are configured to run inside a container.

- First we set up an EMR on the AWS academy account. We can give any name to our EMR and add a total of 4 tasks on the EMR. Create a new key value pair and download the .pem file.

- Add appropriate permissions required for EMR,EC2 and autoscaling. Create the EMR cluster.

- Now start running the EMR cluster.
- Check all the EC2 instances that are running in AWS account. There will be 6 EC2 instances running. There will be one master EC2 instance and rest of them will be slave EC2 instances.
- If now if not already added add a new SSH inbound rule in the security of the master instance and add your IP address

Now create a new S3 bucket where we can add all our files that have to run on these EC2 instances.

- Create a new S3 bucket and give a unique name to it.
- Now upload all the files we need to the S3 bucket.
- Upload training.py , prediction.py , TrainingDataset.csv , ValidationDataset.csv
- Make sure the files are visible in the S3 bucket after they are uploaded.

Go the EMR cluster created :

- Check for the option Connect to the Primary node using SSH
- We will get a SSH connection link for terminal

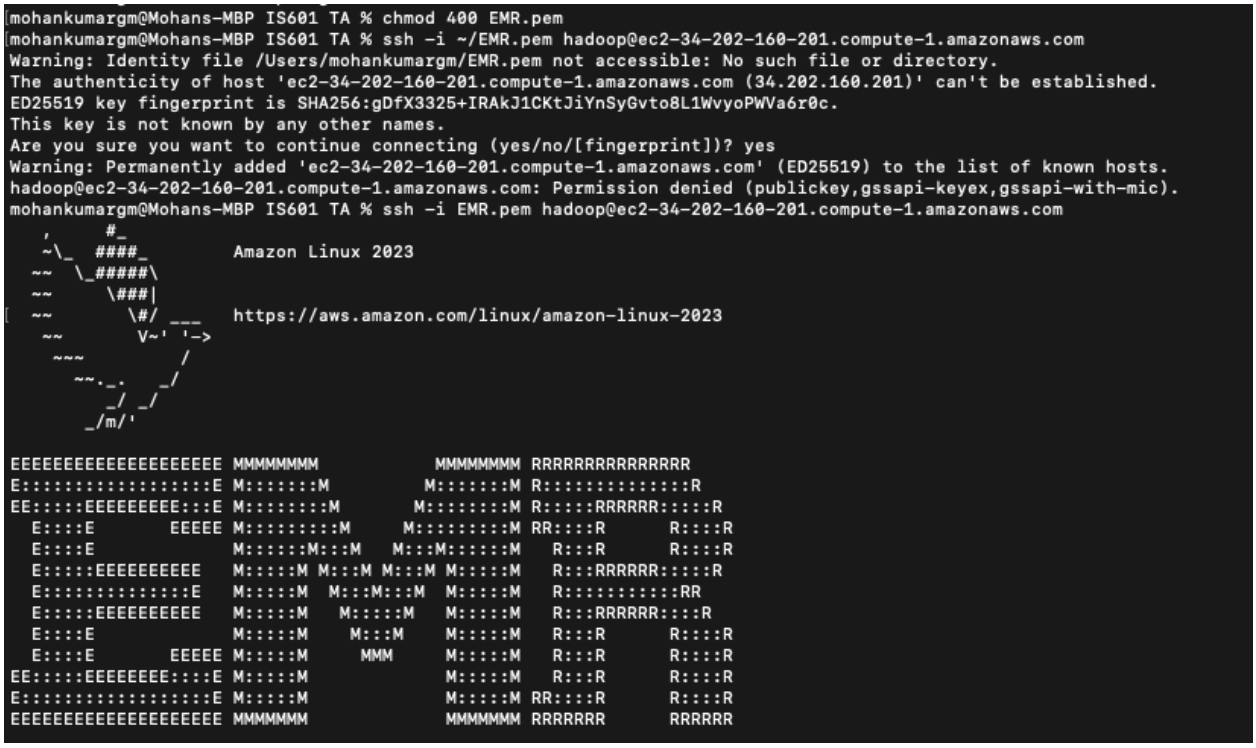
  ssh -i ~/EMR.pem
  hadoop@ec2-34-202-160-201.compute-1.amazonaws.com

- Open a terminal on local where the .pem file is present. Before running the above command we need to change permissions of the .pem file

- Run

  chmod 400 ./EMR.pem

- Now the run SSH connection command



- Successful connection and the hadoop terminal will be connected

- Now connect the S3 bucket with the EMR cluster in the terminal to access its files.

  aws s3 sync s3://emr-ml-bucket/ .

- ‘ls’ command and check if all the files are present.

Setup all python dependencies to run the python files :

- Install numpy dependency

  pip install numpy --user

- Now we have to copy the files to Hadoop file system from the local for both Training.py and Validation.py

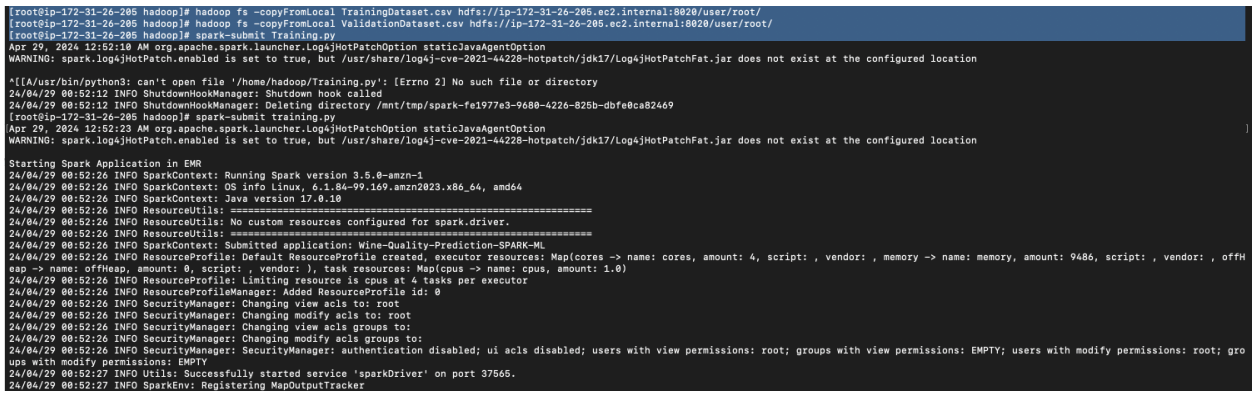
  hadoop fs -copyFromLocal TrainingDataset.csv hdfs://ip-172-31-26-205.ec2.internal:8020/user/root/

  hadoop]# hadoop fs -copyFromLocal ValidationDataset.csv hdfs://ip-172-31-26-205.ec2.internal:8020/user/root/

Once all the dependencies are installed we can run spark submit

- Run the command :

  spark-submit Training.py



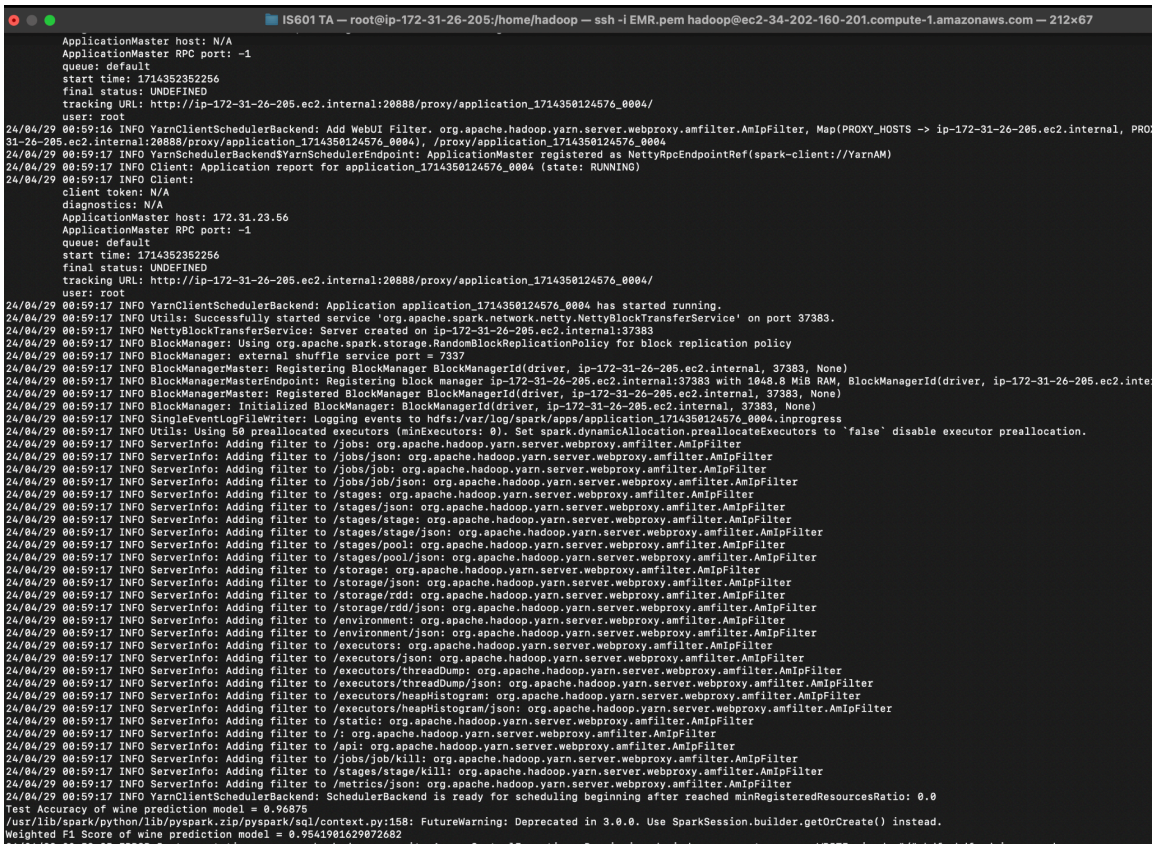
```
24/04/29 00:52:38 INFO ServerInfo: Adding filter to /jobs/job/kill: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/29 00:52:38 INFO ServerInfo: Adding filter to /stages/stage/kill: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/29 00:52:38 INFO ServerInfo: Adding filter to /metrics/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/29 00:52:38 INFO YarnClientSchedulerBackend: SchedulerBackend is ready for scheduling beginning after reached minRegisteredResourcesRatio: 0.0
Reading training CSV file from TrainingDataset.csv
Creating VectorAssembler
Creating StringIndexer
Caching data for faster access
Creating RandomForestClassifier
Creating Pipeline for training
Retraining model on multiple parameters using CrossValidator
Fitting CrossValidator to the training data
Saving the best model to new param `model`
Saving the best model to S3
```

- Once the training.py is completed and a trained model is stored in the S3 bucket. Use ‘ls’ command and find the trained model

- ```
  72-31-26-205 hadoop]# ls
    TrainingDataset.csv  ValidationDataset.csv  prediction.py  trainedmodel  training.py
  ```

- Now run spark command for predicting the data from prediction.py

spark-submit prediction.py


  ```

- We can see a accuracies shown :

  Test Accuracy of wine prediction model = 0.96875
  Weighted F1 Score of wine prediction model = 0.9541901629072682

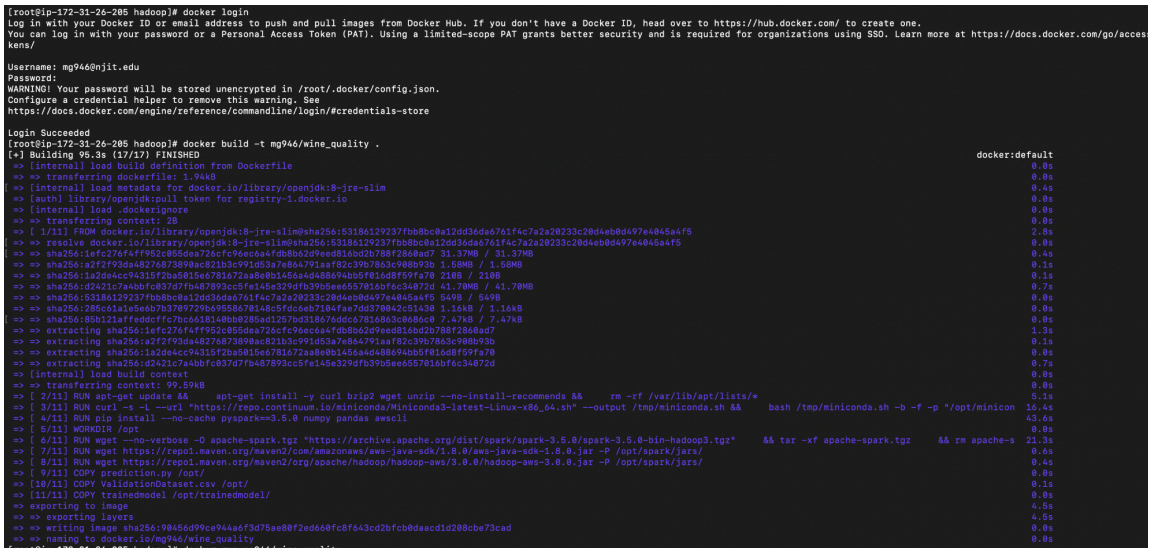
Once our prediction is done we can make our Docker image.

- First create a docker account
- Now in the same root terminal use command :

  docker login

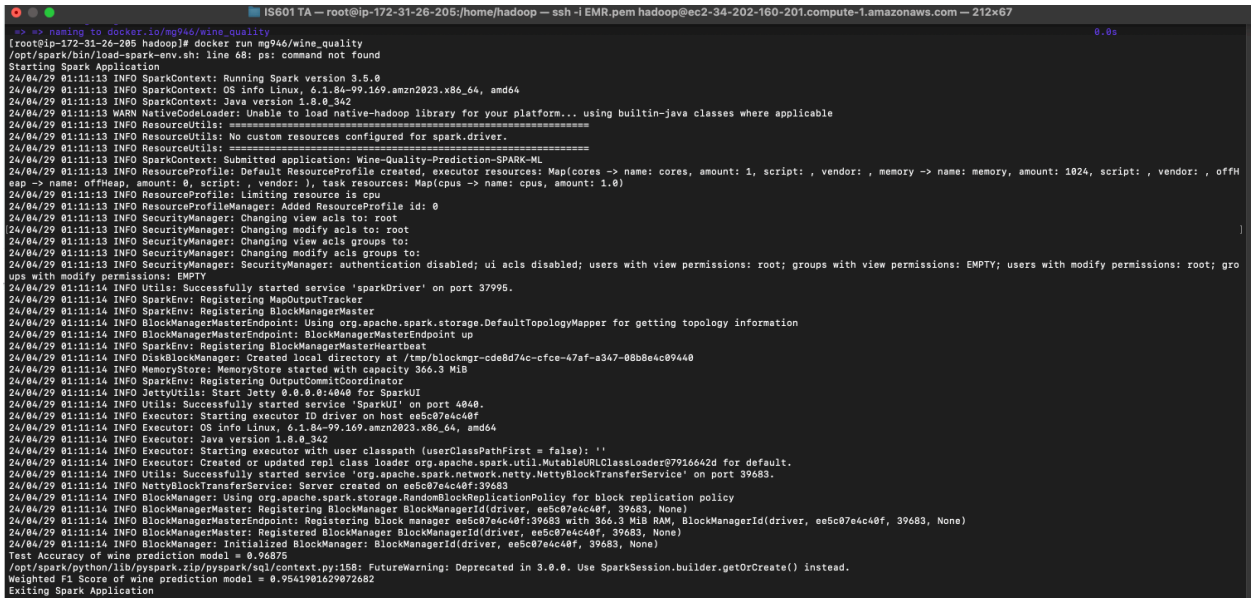
- Enter your docker Username and Password credentials.
- Now we run a command to build the docker file

  docker build -t mg946/wine_quality .



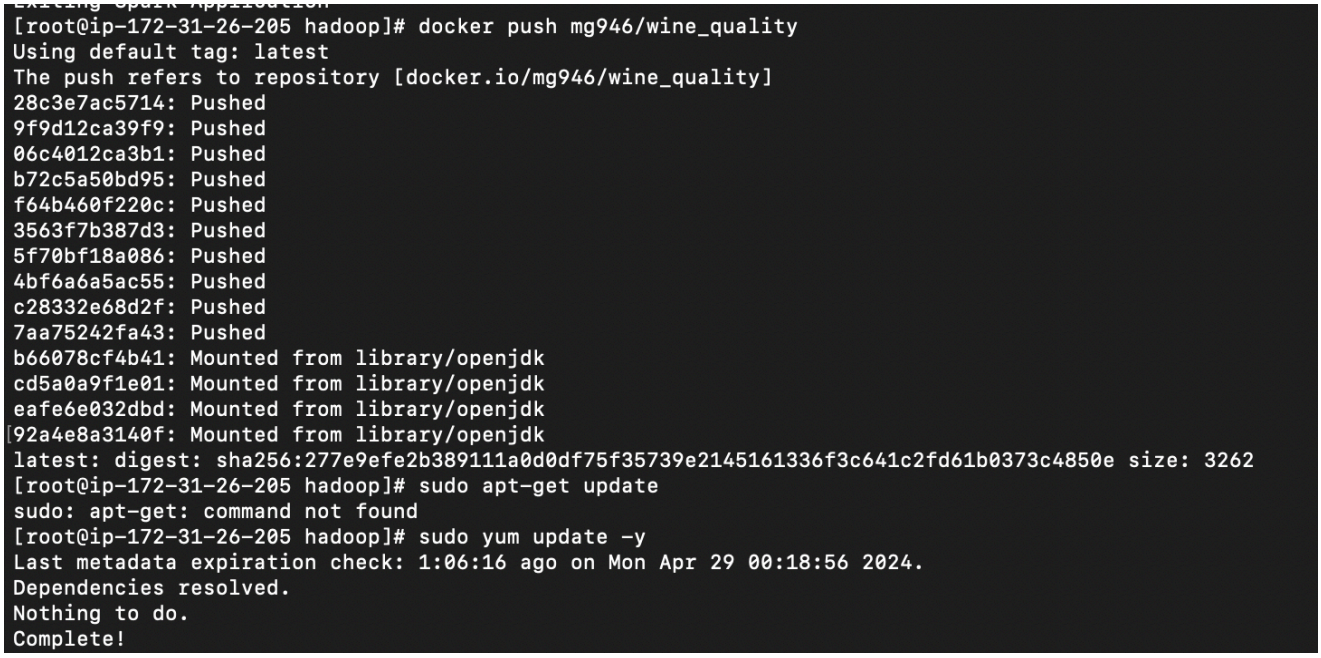
- We check if the Docker file has been created properly by running the file :

docker run mg946/wine_quality



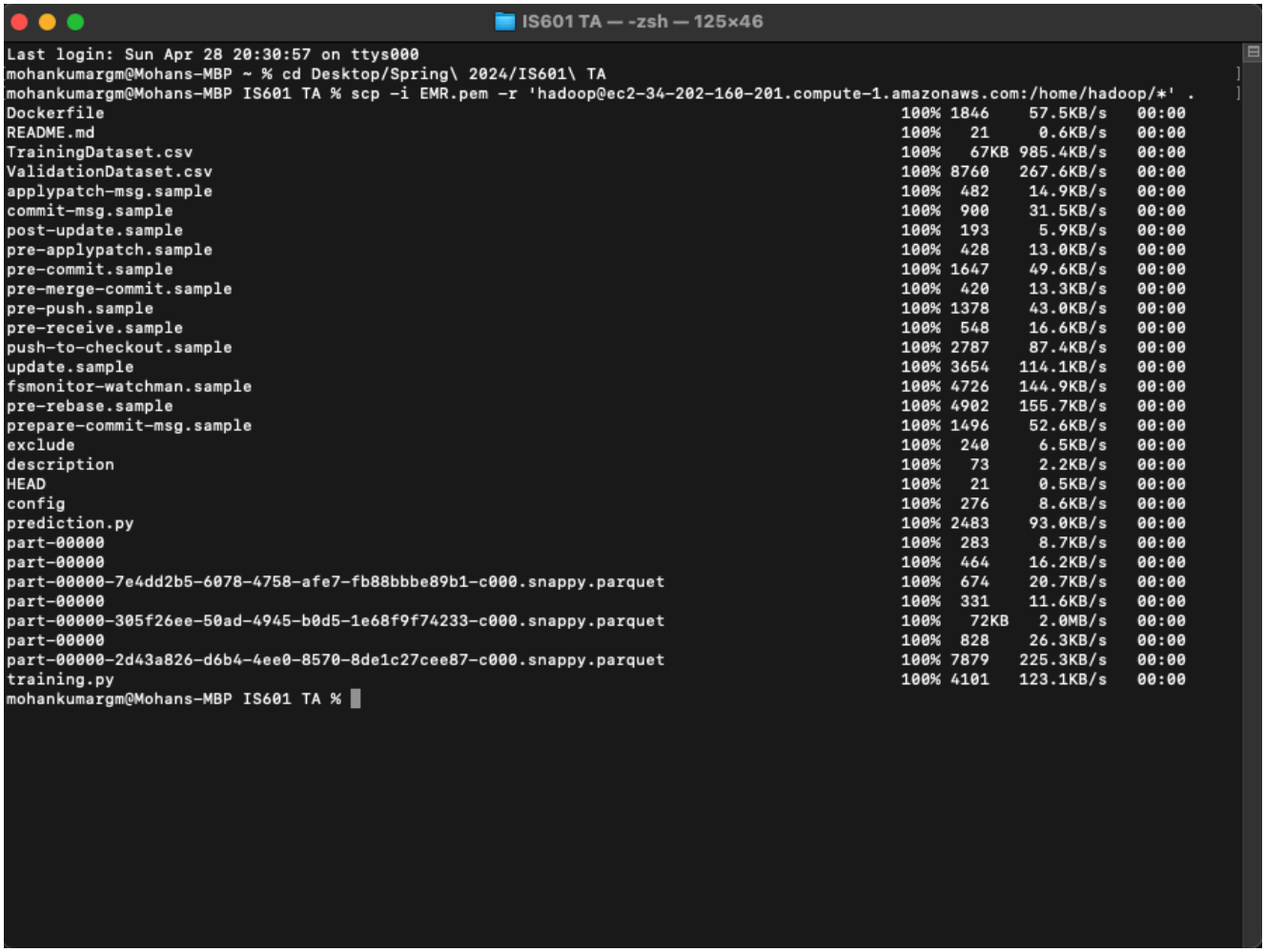
- Once verified push the docker image

docker push mg946/wine_quality

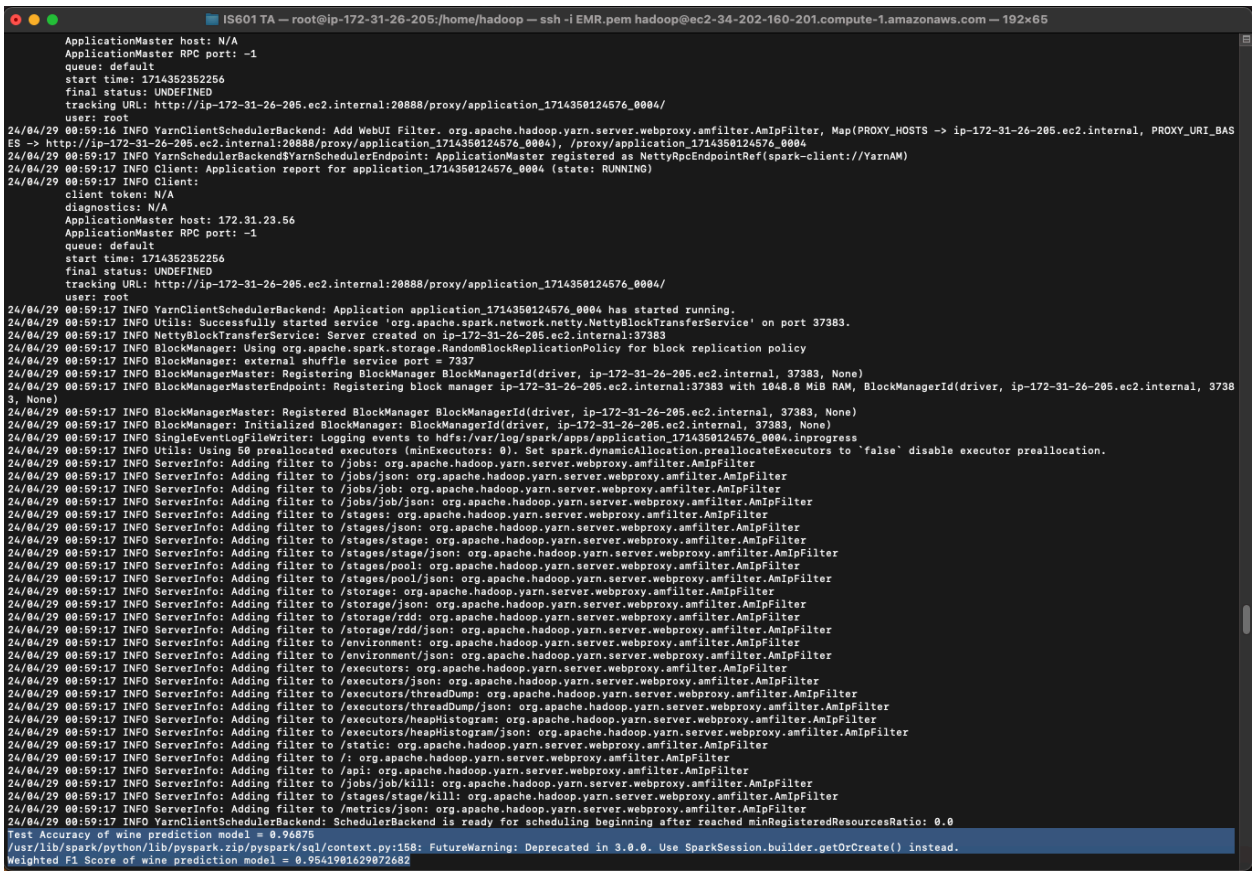


To get all the files present in the Trained Model on local we can run this command :

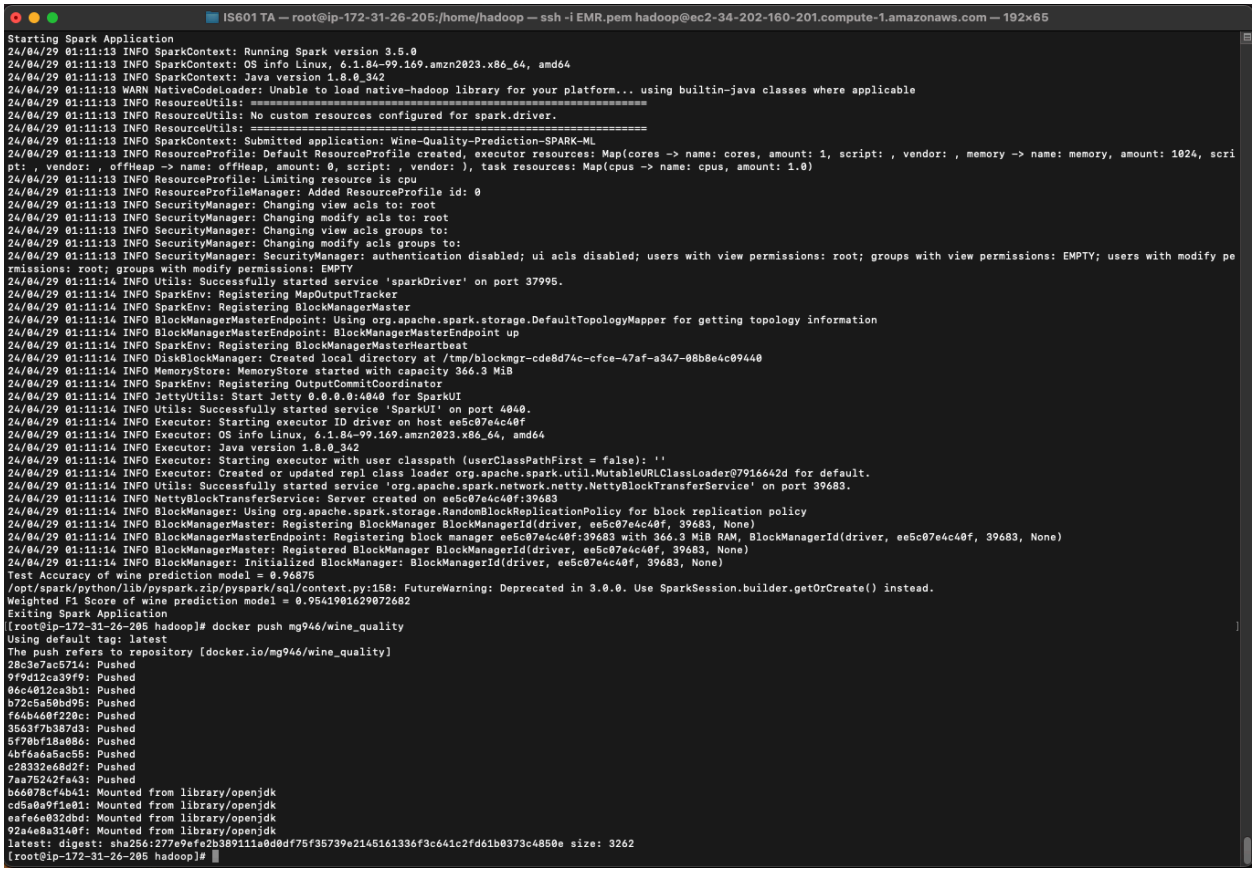
scp -i EMR.pem -r 'hadoop@ec2-34-202-160-201.compute-1.amazonaws.com:/home/hadoop/*' .



Important Screenshots :




IS601 TA — root@ip-172-31-26-205:/home/hadoop — ssh -i EMR.pem hadoop@ec2-34-202-160-201.compute-1.amazonaws.com — 192×65
... 82 more
[root@ip-172-31-26-205 hadoop]# docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/
Username: mg946@njit.edu
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
Login Succeeded
[root@ip-172-31-26-205 hadoop]# docker build -t mg946/wine_quality .
[+] Building 95.3s (17/17) FINISHED docker:default
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 1.94kB 0.0s
=> [internal] load metadata for docker.io/library/openjdk:8-jre-slim 0.4s
=> [auth] library/openjdk:pull token for registry-1.docker.io 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [1/11] FROM docker.io/library/openjdk:8-jre-slim@sha256:53186129237fbb8bc0a12dd36da6761f4c7a2a20233c20d4eb0d497e4045a4f5 2.8s
=> => resolve docker.io/library/openjdk:8-jre-slim@sha256:53186129237fbb8bc0a12dd36da6761f4c7a2a20233c20d4eb0d497e4045a4f5 0.0s
=> => sha256:1efc276f4ff952c055dea726cfc96ec6a4fdb8b62d9eed816bd2b788f2860ad7 31.37MB / 31.37MB 0.4s
=> => sha256:a2f2f93da48276873890ac821b3c991d53a7e864791aaf82c39b7863c908b93b 1.58MB / 1.58MB 0.1s
=> => sha256:1a2de4cc94315f2ba5015e6781672aa8e0b1456a4d488694bb5f016d8f59fa70 210B / 210B 0.1s
=> => sha256:d2421c7a4bbfc037d7fb487893cc5fe145e329dfb39b5ee6557016bf6c34072d 41.70MB / 41.70MB 0.7s
=> => sha256:53186129237fbb8bc0a12dd36da6761f4c7a2a20233c20d4eb0d497e4045a4f5 549B / 549B 0.0s
=> => sha256:285c61a1e5e6b7b3709739e49558b67014c5fdc6eb710afae7dd3700425c51430 1.16kB / 1.16kB 0.0s
=> => sha256:85b121affeddcffc7bc6618140bb0285ad1257bd318676ddc67816863c0686e0 7.47kB / 7.47kB 0.0s
=> => extracting sha256:1efc276f4ff952c055dea726cfc96ec6a4fdb8b62d9eed816bd2b788f2860ad7 1.3s
=> => extracting sha256:a2f2f93da48276873890ac821b3c991d53a7e864791aaf82c39b7863c908b93b 0.1s
=> => extracting sha256:1a2de4cc94315f2ba5015e6781672aa8e0b1456a4d488694bb5f016d8f59fa70 0.0s
=> => extracting sha256:d2421c7a4bbfc037d7fb487893cc5fe145e329dfb39b5ee6557016bf6c34072d 0.7s
=> [internal] load build context 0.0s
=> => transferring context: 99.59kB 0.0s
=> [2/11] RUN apt-get update && apt-get install -y curl bzip2 wget unzip --no-install-recommends && rm -rf /var/lib/apt/lists/* 5.1s
=> [3/11] RUN curl -s -L --url "https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh" --output /tmp/miniconda.sh && bash /tmp/miniconda.sh -b -f -p "/opt/minicon 16.4s
=> [4/11] RUN pip install --no-cache pyspark==3.5.0 numpy pandas awscli 43.6s
=> [5/11] WORKDIR /opt 0.0s
=> [6/11] RUN wget --no-verbose -O apache-spark.tgz "https://archive.apache.org/dist/spark/spark-3.5.0/spark-3.5.0-bin-hadoop3.tgz" && tar -xf apache-spark.tgz && rm apache-s 21.3s
=> [7/11] RUN wget https://repo1.maven.org/maven2/com/amazonaws/aws-java-sdk/1.8.0/aws-java-sdk-1.8.0.jar -P /opt/spark/jars/ 0.6s
=> [8/11] RUN wget https://repo1.maven.org/maven2/org/apache/hadoop/hadoop-aws/3.0.0/hadoop-aws-3.0.0.jar -P /opt/spark/jars/ 0.4s
=> [9/11] COPY prediction.py /opt/ 0.0s
=> [10/11] COPY ValidationDataset.csv /opt/ 0.1s
=> [11/11] COPY trainedmodel /opt/trainedmodel/ 0.0s
=> exporting to image 4.5s
=> => exporting layers 4.5s
=> => writing image sha256:90456d99ce944a6f3d75ae80f2ed660fc8f643cd2bfcb0daacd1d208cbe73cad 0.0s
=> => naming to docker.io/mg946/wine_quality 0.0s
[root@ip-172-31-26-205 hadoop]# docker run mg946/wine_quality
/opt/spark/bin/load-spark-env.sh: line 68: ps: command not found
Starting Spark Application
24/04/29 01:11:13 INFO SparkContext: Running Spark version 3.5.0
24/04/29 01:11:13 INFO SparkContext: OS info Linux, 6.1.84-99.169.amzn2023.x86_64, amd64
24/04/29 01:11:13 INFO SparkContext: Java version 1.8.0_342
24/04/29 01:11:13 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
24/04/29 01:11:13 INFO ResourceUtils: ==
24/04/29 01:11:13 INFO ResourceUtils: No custom resources configured for spark.driver.
24/04/29 01:11:13 INFO ResourceUtils: ==
24/04/29 01:11:13 INFO SparkContext: Submitted application: Wine-Quality-Prediction-SPARK-ML
24/04/29 01:11:13 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 1, script: , vendor: , memory -> name: memory, amount: 1024, script: , vendor: , offHeap -> name: offHeap, amount: 0, script: , vendor:), task resources: Map(cpus -> name: cpus, amount: 1.0)
24/04/29 01:11:13 INFO ResourceProfile: Limiting resource is cpu


IS601 TA — root@ip-172-31-26-205:/home/hadoop — ssh -i EMR.pem hadoop@ec2-34-202-160-201.compute-1.amazonaws.com — 192×65
Starting Spark Application
24/04/29 01:11:13 INFO SparkContext: Running Spark version 3.5.0
24/04/29 01:11:13 INFO SparkContext: OS info Linux, 6.1.84-99.169.amzn2023.x86_64, amd64
24/04/29 01:11:13 INFO SparkContext: Java version 1.8.0_342
24/04/29 01:11:13 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
24/04/29 01:11:13 INFO ResourceUtils: ==
24/04/29 01:11:13 INFO ResourceUtils: No custom resources configured for spark.driver.
24/04/29 01:11:13 INFO ResourceUtils: ==
24/04/29 01:11:13 INFO SparkContext: Submitted application: Wine-Quality-Prediction-SPARK-ML
24/04/29 01:11:13 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 1, script: , vendor: , memory -> name: memory, amount: 1024, script: , vendor: , offHeap -> name: offHeap, amount: 0, script: , vendor:), task resources: Map(cpus -> name: cpus, amount: 1.0)
24/04/29 01:11:13 INFO ResourceProfile: Limiting resource is cpu
24/04/29 01:11:13 INFO ResourceProfileManager: Added ResourceProfile id: 0
24/04/29 01:11:13 INFO SecurityManager: Changing view acls to: root
24/04/29 01:11:13 INFO SecurityManager: Changing modify acls to: root
24/04/29 01:11:13 INFO SecurityManager: Changing view acls groups to:
24/04/29 01:11:13 INFO SecurityManager: Changing modify acls groups to:
24/04/29 01:11:13 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: root; groups with view permissions: EMPTY; users with modify permissions: root; groups with modify permissions: EMPTY
24/04/29 01:11:14 INFO Utils: Successfully started service 'sparkDriver' on port 37995.
24/04/29 01:11:14 INFO SparkEnv: Registering MapOutputTracker
24/04/29 01:11:14 INFO SparkEnv: Registering BlockManagerMaster
24/04/29 01:11:14 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
24/04/29 01:11:14 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
24/04/29 01:11:14 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
24/04/29 01:11:14 INFO DiskBlockManager: Created local directory at /tmp/blockmgr-cde8d74c-cfce-47af-a347-08b8e4c09440
24/04/29 01:11:14 INFO MemoryStore: MemoryStore started with capacity 366.3 MiB
24/04/29 01:11:14 INFO SparkEnv: Registering OutputCommitCoordinator
24/04/29 01:11:14 INFO JettyUtils: Start Jetty 0.0.0.0:4040 for SparkUI
24/04/29 01:11:14 INFO Utils: Successfully started service 'SparkUI' on port 4040.
24/04/29 01:11:14 INFO Executor: Starting executor ID driver on host ee5c07e4c40f
24/04/29 01:11:14 INFO Executor: OS info Linux, 6.1.84-99.169.amzn2023.x86_64, amd64
24/04/29 01:11:14 INFO Executor: Java version 1.8.0_342
24/04/29 01:11:14 INFO Executor: Starting executor with user classpath (userClassPathFirst = false): ''
24/04/29 01:11:14 INFO Executor: Created or updated repl class loader org.apache.spark.util.MutableURLClassLoader@7916642d for default.
24/04/29 01:11:14 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 39683.
24/04/29 01:11:14 INFO NettyBlockTransferService: Server created on ee5c07e4c40f:39683
24/04/29 01:11:14 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
24/04/29 01:11:14 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, ee5c07e4c40f, 39683, None)
24/04/29 01:11:14 INFO BlockManagerMasterEndpoint: Registering block manager ee5c07e4c40f:39683 with 366.3 MiB RAM, BlockManagerId(driver, ee5c07e4c40f, 39683, None)
24/04/29 01:11:14 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, ee5c07e4c40f, 39683, None)
24/04/29 01:11:14 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, ee5c07e4c40f, 39683, None)
Test Accuracy of wine prediction model = 0.96875
/opt/spark/python/lib/pyspark.zip/pyspark/sql/context.py:158: FutureWarning: Deprecated in 3.0.0. Use SparkSession.builder.getOrCreate() instead.
Weighted F1 Score of wine prediction model = 0.9541901629072682
Exiting Spark Application
[root@ip-172-31-26-205 hadoop]# docker push mg946/wine_quality
Using default tag: latest
The push refers to repository [docker.io/mg946/wine_quality]
28c3e7ac5714: Pushed
9f9d12ca39f9: Pushed
0dc4012ca3b1: Pushed
b72c5a50bd95: Pushed
f64b460f220c: Pushed
3563f7b387d3: Pushed
5f70bf18a086: Pushed
4bf6a6a5ac55: Pushed
c28332e68d2f: Pushed
7aa75242fa43: Pushed
b66078cf4b41: Mounted from library/openjdk
cd5a0a9f1e01: Mounted from library/openjdk
eafe6e032dbd: Mounted from library/openjdk
92a4e8a3140f: Mounted from library/openjdk
latest: digest: sha256:277e9efe2b389111a0d0df75f35739e2145161336f3c641c2fd61b0373c4850e size: 3262
[root@ip-172-31-26-205 hadoop]#

```
-D <property=value>               define a value for a given property
-fs <file:///|hdfs://namenode:port> specify default filesystem URL to use, overrides 'fs.defaultFS' property from configurations.
-jt <local|resourcemanager:port>  specify a ResourceManager
-files <file1,...>                specify a comma-separated list of files to be copied to the map reduce cluster
-libjars <jar1,...>               specify a comma-separated list of jar files to be included in the classpath
-archives <archive1,...>          specify a comma-separated list of archives to be unarchived on the compute machines

The general command line syntax is:
command [genericOptions] [commandOptions]

Usage: hadoop fs [generic options] -copyFromLocal [-f] [-p] [-l] [-d] [-t <thread count>] [-q <thread pool queue size>] <localsrc> ... <dst>
[root@ip-172-31-26-205 hadoop]# hadoop fs -copyFromLocal TrainingDataset.csv hdfs://ip-172-31-26-205.ec2.internal:8020/user/root/
[root@ip-172-31-26-205 hadoop]# hadoop fs -copyFromLocal ValidationDataset.csv hdfs://ip-172-31-26-205.ec2.internal:8020/user/root/
[root@ip-172-31-26-205 hadoop]# spark-submit Training.py
Apr 29, 2024 12:52:10 AM 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

^[[A/usr/bin/python3: can't open file '/home/hadoop/Training.py': [Errno 2] No such file or directory
24/04/29 00:52:12 INFO ShutdownHookManager: Shutdown hook called
24/04/29 00:52:12 INFO ShutdownHookManager: Deleting directory /mnt/tmp/spark-fe1977e3-9680-4226-825b-dbfe0ca82469
[root@ip-172-31-26-205 hadoop]# spark-submit training.py
Apr 29, 2024 12:52:23 AM 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

Starting Spark Application in EMR
24/04/29 00:52:26 INFO SparkContext: Running Spark version 3.5.0-amzn-1
24/04/29 00:52:26 INFO SparkContext: OS info Linux, 6.1.84-99.169.amzn2023.x86_64, amd64
24/04/29 00:52:26 INFO SparkContext: Java version 17.0.10
24/04/29 00:52:26 INFO ResourceUtils: ==============================================================
24/04/29 00:52:26 INFO ResourceUtils: No custom resources configured for spark.driver.
24/04/29 00:52:26 INFO ResourceUtils: ==============================================================
24/04/29 00:52:26 INFO SparkContext: Submitted application: Wine-Quality-Prediction-SPARK-ML
24/04/29 00:52:26 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 4, script: , vendor: , memory -> name: memory, amount: 9486, script: , vendor: , offH
eap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpus -> name: cpus, amount: 1.0)
24/04/29 00:52:26 INFO ResourceProfile: Limiting resource is cpus at 4 tasks per executor
24/04/29 00:52:26 INFO ResourceProfileManager: Added ResourceProfile id: 0
24/04/29 00:52:26 INFO SecurityManager: Changing view acls to: root
24/04/29 00:52:26 INFO SecurityManager: Changing modify acls to: root
24/04/29 00:52:26 INFO SecurityManager: Changing view acls groups to:
24/04/29 00:52:26 INFO SecurityManager: Changing modify acls groups to:
24/04/29 00:52:26 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: root; groups with view permissions: EMPTY; users with modify permissions: root; gro
ups with modify permissions: EMPTY
24/04/29 00:52:27 INFO Utils: Successfully started service 'sparkDriver' on port 37565.
24/04/29 00:52:27 INFO SparkEnv: Registering MapOutputTracker
24/04/29 00:52:27 INFO SparkEnv: Registering BlockManagerMaster
24/04/29 00:52:27 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
24/04/29 00:52:27 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
24/04/29 00:52:27 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
24/04/29 00:52:27 INFO DiskBlockManager: Created local directory at /mnt/tmp/blockmgr-5ec1acfa-b7e7-42e4-89b0-65a802cef7bb
24/04/29 00:52:27 INFO MemoryStore: MemoryStore started with capacity 1048.8 MiB
24/04/29 00:52:27 INFO SparkEnv: Registering OutputCommitCoordinator
24/04/29 00:52:27 INFO SubResultCacheManager: Sub-result caches are disabled.
24/04/29 00:52:27 INFO JettyUtils: Start Jetty 0.0.0.0:4040 for SparkUI
24/04/29 00:52:27 INFO Utils: Successfully started service 'SparkUI' on port 4040.
24/04/29 00:52:27 INFO Utils: Using 50 preallocated executors (minExecutors: 0). Set spark.dynamicAllocation.preallocateExecutors to `false` disable executor preallocation.
24/04/29 00:52:27 INFO DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at ip-172-31-26-205.ec2.internal/172.31.26.205:8032
24/04/29 00:52:28 INFO Configuration: resource-types.xml not found
24/04/29 00:52:28 INFO ResourceUtils: Unable to find 'resource-types.xml'.
24/04/29 00:52:28 INFO Client: Verifying our application has not requested more than the maximum memory capability of the cluster (12288 MB per container)
24/04/29 00:52:28 INFO Client: Will allocate AM container, with 896 MB memory including 384 MB overhead
24/04/29 00:52:28 INFO Client: Setting up container launch context for our AM
24/04/29 00:52:28 INFO Client: Setting up the launch environment for our AM container
24/04/29 00:52:28 INFO Client: Preparing resources for our AM container
24/04/29 00:52:28 WARN Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading libraries under SPARK_HOME.
24/04/29 00:52:29 INFO Client: Uploading resource file:/mnt/tmp/spark-b49d6471-182e-4386-a036-df91bb6c0ab1/__spark_libs__2939782902706113350.zip -> hdfs://ip-172-31-26-205.ec2.internal:8020/user/root/.sparkStagin
g/application_1714350124576_0002/__spark_libs__2939782902706113350.zip
24/04/29 00:52:30 INFO Client: Uploading resource file:/etc/spark/conf.dist/hive-site.xml -> hdfs://ip-172-31-26-205.ec2.internal:8020/user/root/.sparkStaging/application_1714350124576_0002/hive-site.xml
```

```
IS601 TA — root@ip-172-31-26-205:/home/hadoop — ssh -i EMR.pem hadoop@ec2-34-202-160-201.compute-1.amazonaws.com — 212×67
 => => naming to docker.io/mg946/wine_quality                                                                                                                                                    0.0s
[root@ip-172-31-26-205 hadoop]# docker run mg946/wine_quality
/opt/spark/bin/load-spark-env.sh: line 68: ps: command not found
Starting Spark Application
24/04/29 01:11:13 INFO SparkContext: Running Spark version 3.5.0
24/04/29 01:11:13 INFO SparkContext: OS info Linux, 6.1.84-99.169.amzn2023.x86_64, amd64
24/04/29 01:11:13 INFO SparkContext: Java version 1.8.0_342
24/04/29 01:11:13 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
24/04/29 01:11:13 INFO ResourceUtils: ==============================================================
24/04/29 01:11:13 INFO ResourceUtils: No custom resources configured for spark.driver.
24/04/29 01:11:13 INFO ResourceUtils: ==============================================================
24/04/29 01:11:13 INFO SparkContext: Submitted application: Wine-Quality-Prediction-SPARK-ML
24/04/29 01:11:13 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 1, script: , vendor: , memory -> name: memory, amount: 1024, script: , vendor: , offH
eap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpus -> name: cpus, amount: 1.0)
24/04/29 01:11:13 INFO ResourceProfile: Limiting resource is cpu
24/04/29 01:11:13 INFO ResourceProfileManager: Added ResourceProfile id: 0
24/04/29 01:11:13 INFO SecurityManager: Changing view acls to: root
24/04/29 01:11:13 INFO SecurityManager: Changing modify acls to: root
24/04/29 01:11:13 INFO SecurityManager: Changing view acls groups to:
24/04/29 01:11:13 INFO SecurityManager: Changing modify acls groups to:
24/04/29 01:11:13 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: root; groups with view permissions: EMPTY; users with modify permissions: root; gro
ups with modify permissions: EMPTY
24/04/29 01:11:14 INFO Utils: Successfully started service 'sparkDriver' on port 37995.
24/04/29 01:11:14 INFO SparkEnv: Registering MapOutputTracker
24/04/29 01:11:14 INFO SparkEnv: Registering BlockManagerMaster
24/04/29 01:11:14 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
24/04/29 01:11:14 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
24/04/29 01:11:14 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
24/04/29 01:11:14 INFO DiskBlockManager: Created local directory at /tmp/blockmgr-cde8d74c-cfce-47af-a347-08b8e4c09440
24/04/29 01:11:14 INFO MemoryStore: MemoryStore started with capacity 366.3 MiB
24/04/29 01:11:14 INFO SparkEnv: Registering OutputCommitCoordinator
24/04/29 01:11:14 INFO JettyUtils: Start Jetty 0.0.0.0:4040 for SparkUI
24/04/29 01:11:14 INFO Utils: Successfully started service 'SparkUI' on port 4040.
24/04/29 01:11:14 INFO Executor: Starting executor ID driver on host ee5c07e4c40f
24/04/29 01:11:14 INFO Executor: OS info Linux, 6.1.84-99.169.amzn2023.x86_64, amd64
24/04/29 01:11:14 INFO Executor: Java version 1.8.0_342
24/04/29 01:11:14 INFO Executor: Starting executor with user classpath (userClassPathFirst = false): ''
24/04/29 01:11:14 INFO Executor: Created or updated repl class loader org.apache.spark.util.MutableURLClassLoader@7916642d for default.
24/04/29 01:11:14 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 39683.
24/04/29 01:11:14 INFO NettyBlockTransferService: Server created on ee5c07e4c40f:39683
24/04/29 01:11:14 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
24/04/29 01:11:14 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, ee5c07e4c40f, 39683, None)
24/04/29 01:11:14 INFO BlockManagerMasterEndpoint: Registering block manager ee5c07e4c40f:39683 with 366.3 MiB RAM, BlockManagerId(driver, ee5c07e4c40f, 39683, None)
24/04/29 01:11:14 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, ee5c07e4c40f, 39683, None)
24/04/29 01:11:14 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, ee5c07e4c40f, 39683, None)
Test Accuracy of wine prediction model = 0.96875
/opt/spark/python/lib/pyspark.zip/pyspark/sql/context.py:158: FutureWarning: Deprecated in 3.0.0. Use SparkSession.builder.getOrCreate() instead.
Weighted F1 Score of wine prediction model = 0.9561901629072682
Exiting Spark Application
```