

CS643 CLOUD PROGRAMMING ASSIGNMENT 2

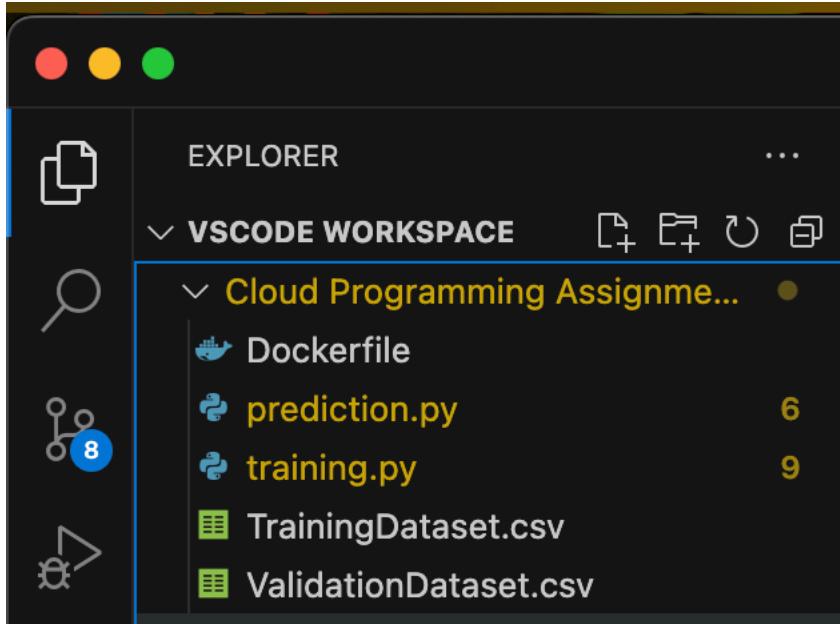
Goal: The purpose of this individual assignment is to learn how to develop parallel machine learning (ML) applications in Amazon AWS cloud platform. Specifically, you will learn: (1) how to use [Apache Spark](#) to train an ML model in parallel on multiple EC2 instances; (2) how to use [Spark's MLlib](#) to develop and use an ML model in the cloud; (3) How to use [Docker](#) to create a container for your ML model to simplify model deployment.

Description: You have to build a wine quality prediction ML model in Spark over AWS. The model must be trained in parallel using 4 EC2 instances. Then, you need to save and load the model in a Spark application that will perform wine quality prediction; this application will run on one EC2 instance. The assignment must be implemented in Java on Ubuntu Linux. The details of the assignment are presented below:

- Input for model training: we share 2 datasets with you for your ML model. Both datasets are available in Canvas, under Programming Assignment 2.
 - TrainingDataset.csv: you will use this dataset to train the model in parallel on multiple EC2 instances.
 - ValidationDataset.csv: you will use this dataset to validate the model and optimize its performance (i.e., select the best values for the model parameters).
- Input for prediction testing: TestDataset.csv. We will use this file, which has a similar structure with the two datasets above, to test the functionality and performance of your prediction application. Your prediction application should take such a file as input. This file is not shared with you, but you can use the validation dataset to make sure your application works.
- Output: The output of your application will be a measure of the prediction performance, specifically the F1 score, which is available in MLlib.
- Model Implementation: You have to develop a Spark application that uses MLlib to train for wine quality prediction using the training dataset. You will use the validation dataset check the performance of your trained model and to potentially tune your ML model parameters for best performance. You should start with a simple linear regression or logistic regression model from MLlib, but you can try multiple ML models to see which one leads to better performance. For classification models, you can use 10 classes (the wine scores are from 1 to 10). Note: there will be extra-credit for the top 5 applications/students in terms of prediction performance (see below under grading).
- Docker container: You have to build a Docker container for your prediction application. In this way, the prediction model can be quickly deployed across many different environments.
- The model training is done in parallel on 4 EC2 instances.
- The prediction with or without Docker is done on a single EC2 instance.

Solution:

File structure of cloud programming assignment 2 in VS CODE



Create an EMR(Elastic Map Reduce Cluster) as shown in the steps below and include 4 task instance groups to run in parallel.

It create one core, master and 4 slave instances.

Core

Master

Task 1

Task 2

Task 3

Task 4

Let everything else be default

create a key pair .pem file

Select IAM roles as specified in the images

AWS Services Search [Option+S] □

Name and applications - required Info

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

Name
Wine_Prediction

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

AmazonCloudWatchAgent 1.300032.2 Flink 1.18.1 HBase 2.4.17

HCatalog 3.1.3 Hadoop 3.3.6 Hive 3.1.3

Hue 4.11.0 JupyterEnterpriseGateway 2.6.0 JupyterHub 1.5.0

Livy 0.8.0 MXNet 1.9.1 Oozie 5.2.1

Phoenix 5.1.3 Pig 0.17.0 Presto 0.284

Spark 3.5.0 Sqoop 1.4.7 TensorFlow 2.11.0

Tez 0.10.2 Trino 435 Zeppelin 0.10.1

ZooKeeper 3.9.1

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, authorization, and instance metadata service settings.

Choose a security configuration

Amazon EC2 key pair for SSH to the cluster Info

EMR_Cluster

▼ **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 ▼ C

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 ▼ C

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
EMR_AutoScaling_DefaultRole ▼ C Create IAM role ↗

Open EMR in AWS and you can see the clusters
 Go to EC2 console and check which instance is master and edit the inbound rules under security tab
 Add rule -> select SSH and MyIP in two drop-downs and then select save.

Move to Cluster Wine_Prediction now and you can see link to
 “connect to the Primary node using SSH”

Move to folder that has key pair in terminal and execute below commands

```
chmod 400 EMR_Cluster.pem
ssh -i EMR_Cluster.pem
hadoop@ec2-18-215-161-107.compute-1.amazonaws.com
```

The screenshot shows the Amazon EMR console with the following details:

- Cluster ID:** j-GV1H851V5VAI
- Cluster configuration:** Instance groups
- Capacity:** 1 Primary | 1 Core | 4 Task
- Applications:** Amazon EMR version emr-7.1.0, Installed applications Hadoop 3.3.6, Hive 3.1.3, JupyterEnterpriseGateway 2.6.0, Livy 0.8.0, Spark 3.5.0
- Cluster management:** Log destination in Amazon S3 aws-logs-058264194150-us-east-1/elasticmapreduce, Persistent application UIs Spark History Server, YARN timeline server, Tez UI
- Status and time:** Status Waiting, Creation time April 27, 2024, 16:59 (UTC-04:00), Elapsed time 1 hour
- Primary node public DNS:** ec2-18-215-161-107.compute-1.amazonaws.com
- Connect options:** Connect to the Primary node using SSH, Connect to the Primary node using SSM

Below the summary, there are tabs for Properties, Bootstrap actions, Instances (Hardware), Steps, Applications, Configurations, Monitoring, Events, and Tags (0).

The modal window provides instructions for connecting via 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](#)

Two tabs are shown: Windows and Mac/Linux. The Mac/Linux tab is selected.

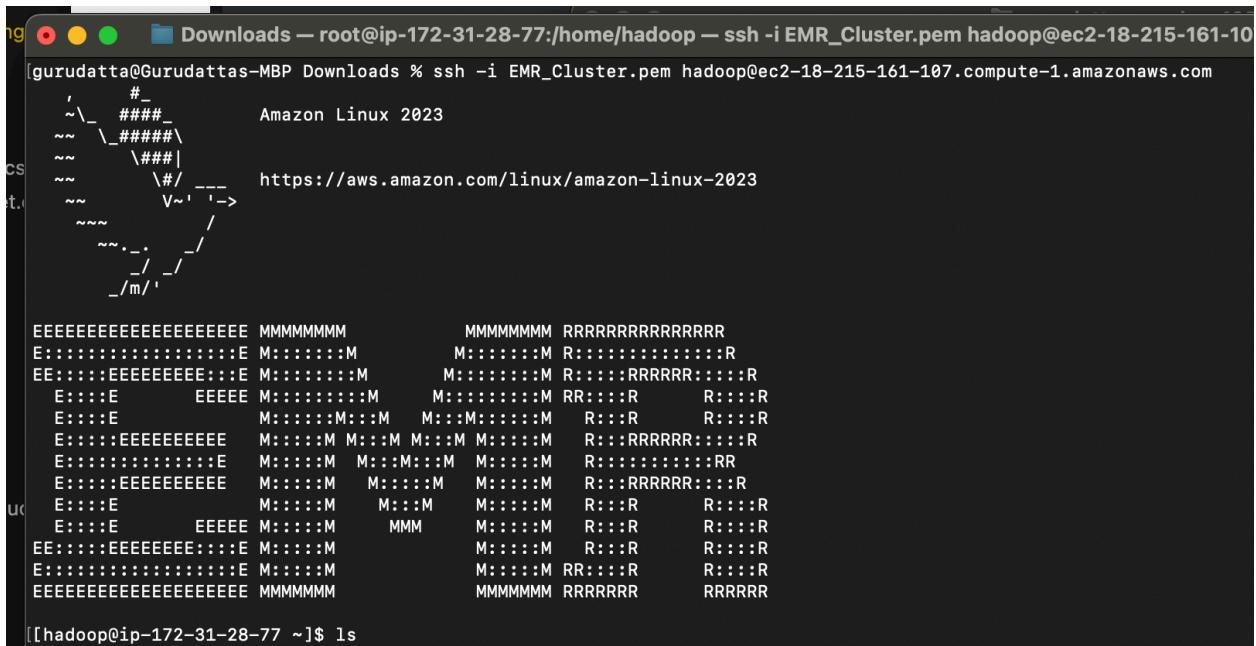
- Open a terminal window. On Mac OS X, choose Applications > Utilities > Terminal. On other Linux distributions, terminal is typically found at Applications > Accessories > Terminal.
- To establish a connection to the primary node, enter the following command. Replace ~/EMR_Cluster.pem with the location and filename of the private key file (.pem) that you used to launch the cluster.

```
ssh -i ~/EMR_Cluster.pem hadoop@ec2-18-215-161-107.compute-1.amazonaws.com
```

- Enter yes to dismiss the security warning.

At the bottom, there is a link to View web interfaces hosted on Amazon EMR clusters and a Close button.

Instances (6) Info										
							Connect	Instance state ▾	Actions ▾	Launch instances ▾
Find Instance by attribute or tag (case-sensitive)										
Instance state = running X Clear filters										
Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP	IPv6 IPs	Monitoring	Security group name	Key name	Last modified	Launch time	Actions
	ec2-52-91-92-72.compute-1.amazonaws.com	52.91.92.72	-	-	disabled	ElasticMapReduce-slave	EMR_Cluster	2024/04/10 10:45:10	2024/04/10 10:45:10	Edit Stop Start Reboot Delete
	ec2-107-20-68-105.compute-1.amazonaws.com	107.20.68.105	-	-	disabled	ElasticMapReduce-slave	EMR_Cluster	2024/04/10 10:45:10	2024/04/10 10:45:10	Edit Stop Start Reboot Delete
	ec2-18-215-161-107.compute-1.amazonaws.com	18.215.161.107	-	-	disabled	ElasticMapReduce-master	EMR_Cluster	2024/04/10 10:45:10	2024/04/10 10:45:10	Edit Stop Start Reboot Delete
	ec2-54-167-79-102.compute-1.amazonaws.com	54.167.79.102	-	-	disabled	ElasticMapReduce-slave	EMR_Cluster	2024/04/10 10:45:10	2024/04/10 10:45:10	Edit Stop Start Reboot Delete
	ec2-54-196-242-15.compute-1.amazonaws.com	54.196.242.15	-	-	disabled	ElasticMapReduce-slave	EMR_Cluster	2024/04/10 10:45:10	2024/04/10 10:45:10	Edit Stop Start Reboot Delete
	ec2-18-208-220-43.compute-1.amazonaws.com	18.208.220.43	-	-	disabled	ElasticMapReduce-slave	EMR_Cluster	2024/04/10 10:45:10	2024/04/10 10:45:10	Edit Stop Start Reboot Delete



```
gurudatta@Gurudattas-MBP Downloads % ssh -i EMR_Cluster.pem hadoop@ec2-18-215-161-107.compute-1.amazonaws.com
[Truncated]

```

Go to S3 Console and create S3 Bucket “pa2-ml-spark” and upload all required files into the bucket

upload all the code files and datasets and docker file as we will be running everything inside EMR cluster master ec2 that we just have created.

The screenshot shows the AWS S3 console with the path [Amazon S3](#) > [Buckets](#) > [pa2-ml-spark](#). The 'pa2-ml-spark' bucket page is displayed, showing the following details:

- Objects (5) [Info](#)**
- Action buttons: Copy S3 URI, Copy URL, Download, Open, Delete, Actions (dropdown), Create folder, Upload.
- A note: "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](#)"
- Search bar:
- Pagination: < 1 > | ⌂
- Table headers: Name, Type, Last modified, Size, Storage class.
- Table data:

Name	Type	Last modified	Size	Storage class
Dockerfile	-	April 27, 2024, 17:29:10 (UTC-04:00)	1.8 KB	Standard
prediction.py	py	April 27, 2024, 17:28:22 (UTC-04:00)	2.4 KB	Standard
training.py	py	April 27, 2024, 16:59:29 (UTC-04:00)	4.0 KB	Standard
TrainingDataset.csv	csv	April 27, 2024, 16:59:29 (UTC-04:00)	67.2 KB	Standard
ValidationDataset.csv	csv	April 27, 2024, 16:59:29 (UTC-04:00)	8.6 KB	Standard

Execution without Docker

Go back to terminal now and type in the below commands

sudo su -> for changing to the root (optional)

aws s3 sync s3://pa2-ml-spark/ . -> fetching all code files into the instance
ls -> cross check if all files are dumped into instance

```
[[hadoop@ip-172-31-28-77 ~]$ ls
[[hadoop@ip-172-31-28-77 ~]$ aws s3 sync s3://pa2-ml-spark/ .
download: s3://pa2-ml-spark/TrainingDataset.csv to ./TrainingDataset.csv
download: s3://pa2-ml-spark/prediction.py to ./prediction.py
download: s3://pa2-ml-spark/Dockerfile to ./Dockerfile
download: s3://pa2-ml-spark/training.py to ./training.py
download: s3://pa2-ml-spark/ValidationDataset.csv to ./ValidationDataset.csv
[[hadoop@ip-172-31-28-77 ~]$ ls
Dockerfile  TrainingDataset.csv  ValidationDataset.csv  prediction.py  training.py
```

As this assignment requires to run the ML code of Wine_Quality prediction using SPARK ML Libraries, we use the below command

Pip install numpy —user -> for installing python numpy into instance
 spark-submit training.py -> running training.py to generate a training model
 In our case it creates a trained model with name spark-model

```

Successfully installed numpy-1.26.4
[[hadoop@ip-172-31-28-77 ~]$ spark-submit training.py
Apr 27, 2024 9:03:15 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 location
Starting Spark Application
24/04/27 21:03:23 INFO SparkContext: Running Spark version 3.5.0-amzn-1
24/04/27 21:03:23 INFO SparkContext: OS info Linux, 6.1.84-99.169.amzn2023.x86_64, amd64
24/04/27 21:03:23 INFO SparkContext: Java version 17.0.10
24/04/27 21:03:23 INFO ResourceUtils: =====
24/04/27 21:03:23 INFO ResourceUtils: No custom resources configured for spark.driver.
24/04/27 21:03:23 INFO ResourceUtils: =====
24/04/27 21:03:23 INFO SparkContext: Submitted application: WineQualityPrediction
24/04/27 21:03:24 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map[cores -> name: cores, amount: 4, script: , vendor: , name: memory, amount: 9486, script: , vendor: , offHeap -> name: offHeap, amount: 0, script: , vendor: ], task resources: Map[cpus -> name: cpus, amount: 4]
24/04/27 21:03:24 INFO ResourceProfile: Limiting resource is cpus at 4 tasks per executor
24/04/27 21:03:24 INFO ResourceProfileManager: Added ResourceProfile id: 0
24/04/27 21:03:24 INFO SecurityManager: Changing view acls to: hadoop
24/04/27 21:03:24 INFO SecurityManager: Changing modify acls to: hadoop
24/04/27 21:03:24 INFO SecurityManager: Changing view acls groups to:
24/04/27 21:03:24 INFO SecurityManager: Changing modify acls groups to:
24/04/27 21:03:24 INFO SecurityManager: SecurManager authentication disabled; ui acls disabled; users with view permissions: hadoop; groups with permissions: EMPTY; users with modify permissions: hadoop; groups with modify permissions: EMPTY
24/04/27 21:03:25 INFO Utils: Successfully started service 'sparkDriver' on port 39131.
24/04/27 21:03:25 INFO SparkEnv: Registering MapOutputTracker
24/04/27 21:03:25 INFO SparkEnv: Registering BlockManagerMaster
24/04/27 21:03:25 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
24/04/27 21:03:25 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
24/04/27 21:03:25 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
24/04/27 21:03:25 INFO DiskBlockManager: Created local directory at /mnt/tmp/blockmgr-00ed3650-9197-4da9-9a93-2a920aa3ab
24/04/27 21:03:25 INFO MemoryStore: MemoryStore started with capacity 1048.8 MiB
24/04/27 21:03:26 INFO SparkEnv: Registering OutputCommitCoordinator
24/04/27 21:03:26 INFO SubResultCacheManager: Sub-result caches are disabled.
24/04/27 21:03:26 INFO JettyUtils: Start Jetty 0.0.0.0:4040 for SparkUI
24/04/27 21:03:26 INFO Utils: Successfully started service 'SparkUI' on port 4040.

```

For any Hadoop related path issues

refer the below

```

hadoop fs -copyFromLocal TrainingDataset.csv hdfs://
ip-172-31-28-77.ec2.internal:8020/user/root/

```

Below images shows the list of files and generated ML model spark-model in hdfs

```

24/04/27 21:07:52 INFO ServerInfo: Adding filter to /metrics/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AMIPFilter
24/04/27 21:07:52 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
[root@ip-172-31-28-77 hadoop]# ls
Dockerfile TrainingDataset.csv ValidationDataset.csv prediction.py training.py
[root@ip-172-31-28-77 hadoop]# hadoop fs -ls
Found 3 items
drwxr-xr-x - root hdfsadmingroup 0 2024-04-27 21:09 .sparkStaging
-rw-r--r-- 1 root hdfsadmingroup 68804 2024-04-27 21:07 TrainingDataset.csv
drwxr-xr-x - root hdfsadmingroup 0 2024-04-27 21:09 spark-model

```

Now push ValidationDataset.csv to hdfs

```
hadoop fs -copyFromLocal ValidationDataset.csv hdfs://
ip-172-31-28-77.ec2.internal:8020/user/root/
```

```
[root@ip-172-31-28-77 hadoop]# hadoop fs -copyFromLocal ValidationDataset.csv hdfs://ip-172-31-28-77.ec2.internal:8020/user/root/
[root@ip-172-31-28-77 hadoop]# spark-submit prediction.py
Apr 27, 2024 9:12:59 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
ation

Starting Spark Application
24/04/27 21:13:02 INFO SparkContext: Running Spark version 3.5.0-amzn-1
24/04/27 21:13:02 INFO SparkContext: OS info Linux, 6.1.84-99.169.amzn2023.x86_64, amd64
24/04/27 21:13:02 INFO SparkContext: Java version 17.0.10
24/04/27 21:13:03 INFO ResourceUtils: =====
24/04/27 21:13:03 INFO ResourceUtils: No custom resources configured for spark.driver.
24/04/27 21:13:03 INFO ResourceUtils: =====
24/04/27 21:13:03 INFO SparkContext: Submitted application: WineQualityPrediction
24/04/27 21:13:03 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 4, script: , ve
ame: memory, amount: 9486, script: , vendor: , offHeap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpu -> name: cpu
24/04/27 21:13:03 INFO ResourceProfile: Limiting resource is cpus at 4 tasks per executor
24/04/27 21:13:03 INFO ResourceProfileManager: Added ResourceProfile id: 0
24/04/27 21:13:03 INFO SecurityManager: Changing view acls to: root
24/04/27 21:13:03 INFO SecurityManager: Changing modify acls to: root
24/04/27 21:13:03 INFO SecurityManager: Changing view acls groups to:
24/04/27 21:13:03 INFO SecurityManager: Changing modify acls groups to:
24/04/27 21:13:03 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: root; groups
ns: EMPTY; users with modify permissions: root; groups with modify permissions: EMPTY
24/04/27 21:13:03 INFO Utils: Successfully started service 'sparkDriver' on port 39923.
24/04/27 21:13:03 INFO SparkEnv: Registering MapOutputTracker
24/04/27 21:13:03 INFO SparkEnv: Registering BlockManagerMaster
24/04/27 21:13:03 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
24/04/27 21:13:03 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
24/04/27 21:13:03 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
```

The below image shows the prediction.py execution with validation dataset and shows up the accuracy score as below

Test Accuracy of wine prediction model = 0.96875

```
/usr/lib/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

```
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /stages: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /stages/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /stages/stage: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /stages/stage/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /stages/pool: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /stages/pool/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /storage: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /storage/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /storage/rdd: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /storage/rdd/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /environment: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /environment/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /executors: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /executors/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /executors/threadDump: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /executors/threadDump/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /executors/heaphistogram: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /executors/heaphistogram/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /static: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /api: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /jobs/job/kill: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /stages/stage/kill: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO ServerInfo: Adding filter to /metrics/json: org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter
24/04/27 21:13:14 INFO YarnClientSchedulerBackend: SchedulerBackend is ready for scheduling beginning after reached minRegisteredResourcesRatio: 0.0
Test Accuracy of wine prediction model = 0.96875
/usr/lib/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
```

The above score shows almost 97% test accuracy of prediction in the trained spark-model and Weighted F1 score is around 95.5%

Execution with Docker

Login to docker from the instance

commands : run all of them below from the project directory where Dockerfile is present

docker login -> enter username and password

docker build -t sg2653/ml-spark . -> username/name_of_image

```
[root@ip-172-31-28-77 hadoop]# docker build -t sg2653/ml-spark .
[+] Building 94.5s (17/17) FINISHED
   => [internal] load build definition from Dockerfile
   => transferring dockerfile: 1.92kB
   => [internal] load metadata for docker.io/library/openjdk:8-jre-slim
   => [auth] library/openjdk:pull token for registry-1.docker.io
   => [internal] load .dockerrcignore
   => transferring context: 2B
   => [internal] load build context
   => transferring context: 99.44kB
   [ 1/11] FROM docker.io/library/openjdk:8-jre-slim@sha256:53186129237fbb8bc0a12dd36da6761f4c7a2a20233c20d4eb0d497e4045a4f5
   => resolve docker.io/library/openjdk:8-jre-slim@sha256:53186129237fbb8bc0a12dd36da6761f4c7a2a20233c20d4eb0d497e4045a4f5
   => sha256:a2f2f93da48276873890ac821b3c991d53a7e864791aaaf82c39b7863c988b93b 1.58MB / 1.58MB
   => sha256:a2d2e4cc94315f2ba5815ed781672aa8e0b1456a4d488694bb5f016d8f59fa70 2.9s
   => sha256:d53186129237fbb8bc0a12dd36da6761f4c7a2a20233c20d4eb0d497e4045a4f5
   => sha256:2d421c7a4bbfc037d7fb87893cc5fe145e329dfb39b5ee6557010bf6c34872d 0.1s
   => sha256:2d421c7a4bbfc037d7fb87893cc5fe145e329dfb39b5ee6557010bf6c34872d 21.0s / 21.0s
   => sha256:2d421c7a4bbfc037d7fb87893cc5fe145e329dfb39b5ee6557010bf6c34872d 41.70MB / 41.70MB
   => sha256:2d421c7a4bbfc037d7fb87893cc5fe145e329dfb39b5ee6557010bf6c34872d 54.9B / 54.9B
   => sha256:2d421c7a4bbfc037d7fb87893cc5fe145e329dfb39b5ee6557010bf6c34872d 1.16KB / 1.16KB
   => sha256:2d421c7a4bbfc037d7fb87893cc5fe145e329dfb39b5ee6557010bf6c34872d 7.47KB / 7.47KB
   => sha256:2d421c7a4bbfc037d7fb87893cc5fe145e329dfb39b5ee6557010bf6c34872d 31.37MB / 31.37MB
   => sha256:2d421c7a4bbfc037d7fb87893cc5fe145e329dfb39b5ee6557010bf6c34872d 1.4s
   => extracting sha256:2d421c7a4bbfc037d7fb87893cc5fe145e329dfb39b5ee6557010bf6c34872d 0.1s
   => extracting sha256:2d421c7a4bbfc037d7fb87893cc5fe145e329dfb39b5ee6557010bf6c34872d 0.0s
   => extracting sha256:2d421c7a4bbfc037d7fb87893cc5fe145e329dfb39b5ee6557010bf6c34872d 0.8s
   => [ 2/11] RUN apt-get update & apt-get install -y curl bzip2 wget unzip --no-install-recommends & rm -rf /var/lib/apt/lists/*
   => [ 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
   => [ 4/11] RUN pip install --no-cache pyspark==3.5.0 numpy pandas awscli
   => [ 5/11] WORKDIR /opt
   => [ 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
   => [ 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/
   => [ 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/
   => [ 9/11] COPY prediction.py /
   => [10/11] COPY ValidationDataset.csv /
   => [11/11] COPY spark-model /
   => exporting to image
   => writing image sha256:9552cb55fa2c4e3e72ee1e4040ff227948143a7da6767d77a947123d34ae6605
   => naming to docker.io/sg2653/ml-spark
```

docker run sg2653/ml-spark

In the above command, we need not mention any .py file name or test dataset as arguments because everything is configured inside Dockerfile which takes care of complete execution. All we need it to be aware of file path setup.

```
[root@ip-172-31-28-77 hadoop]# docker run sg2653/ml-spark
/opt/spark/bin/load-spark-env.sh: line 68: ps: command not found
Starting Spark Application
24/04/27 21:23:08 INFO SparkContext: Running Spark version 3.5.0
24/04/27 21:23:08 INFO SparkContext: OS info Linux, 6.1.84-99.169.amzn2023.x86_64, amd64
24/04/27 21:23:08 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
24/04/27 21:23:08 INFO ResourceUtils: =====
24/04/27 21:23:08 INFO ResourceUtils: No custom resources configured for spark.driver.
24/04/27 21:23:08 =====
24/04/27 21:23:08 INFO SparkContext: Submitted application: WineQualityPrediction
24/04/27 21:23:08 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(cpu -> name: cpus, amount: 1.0)
24/04/27 21:23:08 INFO ResourceProfile: Limiting resource is cpu
24/04/27 21:23:08 INFO ResourceProfileManager: Added ResourceProfile id: 0
24/04/27 21:23:08 INFO SecurityManager: Changing view acls to: root
24/04/27 21:23:08 INFO SecurityManager: Changing modify acls to: root
24/04/27 21:23:08 INFO SecurityManager: Changing view acls groups to:
24/04/27 21:23:08 INFO SecurityManager: Changing modify acls groups to:
24/04/27 21:23:08 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
24/04/27 21:23:08 INFO Utils: Successfully started service 'sparkDriver' on port 40969.
24/04/27 21:23:08 INFO SparkEnv: Registering MapOutputTracker
24/04/27 21:23:08 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
24/04/27 21:23:08 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
24/04/27 21:23:08 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
24/04/27 21:23:08 INFO DiskBlockManager: Created local directory at /tmp/blockmgrr-427e113e-5b26-4568-997f-0f15f79a0cac
24/04/27 21:23:08 INFO MemoryStore: MemoryStore started with capacity 366.3 MiB
24/04/27 21:23:08 INFO SparkEnv: Registering OutputCommitCoordinator
24/04/27 21:23:08 INFO JettyUtils: Start Jetty 0.0.0.0:4040 for SparkUI
24/04/27 21:23:08 INFO Utils: Successfully started service 'SparkUI' on port 4040.
24/04/27 21:23:08 INFO Executor: Starting executor ID driver on host 3e84fe4e7688
24/04/27 21:23:08 INFO Executor: OS info Linux, 6.1.84-99.169.amzn2023.x86_64, amd64
24/04/27 21:23:08 INFO Executor: Java version 1.8.0_342
24/04/27 21:23:08 INFO Executor: Starting executor with user classpath (userClassPathFirst = false): ''
24/04/27 21:23:08 INFO Executor: Created or updated repl class loader org.apache.spark.util.MutableURLClassLoader@0d87ecd7 for default.
24/04/27 21:23:08 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 35589.
24/04/27 21:23:09 INFO NettyBlockTransferService: Server created on 3e84fe4e7688:35589
```

```
24/04/27 21:29:30 INFO SparkContext: Submitted application: WineQualityPrediction
24/04/27 21:29:30 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(cpu -> name: cpus, amount: 1.0)
24/04/27 21:29:30 INFO ResourceProfile: Limiting resource is cpu
24/04/27 21:29:30 INFO ResourceProfileManager: Added ResourceProfile id: 0
24/04/27 21:29:30 INFO SecurityManager: Changing view acls to: root
24/04/27 21:29:30 INFO SecurityManager: Changing modify acls to: root
24/04/27 21:29:30 INFO SecurityManager: Changing view acls groups to:
24/04/27 21:29:30 INFO SecurityManager: Changing modify acls groups to:
24/04/27 21:29:30 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
24/04/27 21:29:30 INFO Utils: Successfully started service 'sparkDriver' on port 39173.
24/04/27 21:29:30 INFO SparkEnv: Registering MapOutputTracker
24/04/27 21:29:30 INFO SparkEnv: Registering BlockManagerMaster
24/04/27 21:29:30 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
24/04/27 21:29:30 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
24/04/27 21:29:30 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
24/04/27 21:29:30 INFO DiskBlockManager: Created local directory at /tmp/blockmgrr-2ed66c75-5ef7-448b-966c-7593974a584b
24/04/27 21:29:30 INFO MemoryStore: MemoryStore started with capacity 366.3 MiB
24/04/27 21:29:30 INFO SparkEnv: Registering OutputCommitCoordinator
24/04/27 21:29:31 INFO JettyUtils: Start Jetty 0.0.0.0:4040 for SparkUI
24/04/27 21:29:31 INFO Utils: Successfully started service 'SparkUI' on port 4040.
24/04/27 21:29:31 INFO Executor: Starting executor ID driver on host 830868cd0485
24/04/27 21:29:31 INFO Executor: OS info Linux, 6.1.84-99.169.amzn2023.x86_64, amd64
24/04/27 21:29:31 INFO Executor: Java version 1.8.0_342
24/04/27 21:29:31 INFO Executor: Starting executor with user classpath (userClassPathFirst = false): ''
24/04/27 21:29:31 INFO Executor: Created or updated repl class loader org.apache.spark.util.MutableURLClassLoader@2153aa7c for default.
24/04/27 21:29:31 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 44325.
24/04/27 21:29:31 INFO NettyBlockTransferService: Server created on 830868cd0485:44325
24/04/27 21:29:31 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
24/04/27 21:29:31 INFO BlockManagerMaster: Registering BlockManagerId(driver, 830868cd0485, 44325, None)
24/04/27 21:29:31 INFO BlockManagerMasterEndpoint: Registering block manager 830868cd0485:44325 with 366.3 MiB RAM, BlockManagerId(driver, 830868cd0485, 44325, None)
24/04/27 21:29:31 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, 830868cd0485, 44325, None)
24/04/27 21:29:31 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, 830868cd0485, 44325, 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
```

Output from Execution with Docker

```
24/04/27 21:29:31 INFO BlockManagerMasterEndpoint: Registering BlockManager 830868cd0485:44325 with 366.3 MiB RAM, BlockManagerId(driver, 830868cd0485, 44325, None)
24/04/27 21:29:31 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, 830868cd0485, 44325, None)
24/04/27 21:29:31 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, 830868cd0485, 44325, 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-172-31-28-77 hadoop]# docker push sg2653/ml-spark
```

```
[root@ip-172-31-28-77 hadoop]# docker push sg2653/ml-spark
Using default tag: latest
The push refers to repository [docker.io/sg2653/ml-spark]
94e33ba8ef7a: Pushed
47b7588ce0c1: Pushed
9780aaf4f1e4: Pushed
99952b8d544f: Pushed
a6acf7297fdc: Pushed
ce712ca78f41: Pushed
5f70bf18a086: Pushed
9e25f407b39d: Pushed
10bc2294d76f: Pushed
4d8c41736756: Pushed
b66078cf4b41: Mounted from library/openjdk
cd5a0a9f1e01: Mounted from library/openjdk
eafe6e032dbd: Mounted from library/openjdk
92a4ee8a3140f: Mounted from library/openjdk
latest: digest: sha256:963e6dabb9c11d0b03fb2bb1cf886cc122077aa8912017c369a354384bda3f size: 3262
```

Now we can push the total docker build to docker using the below command

docker push sg2653/ml-spark

so we can even pull the same any time in any machine and run the total model with any add on files using below command

docker pull sg2653/ml-spark

Docker Images

The screenshot shows the Docker Hub 'Images' page. The 'Hub' tab is selected. A search bar contains 'sg2653'. Below it, a table lists the image details:

Tags	OS	Vulnerabilities	Last pushed	Size
latest		Inactive	1 hour ago	1.08 GB

A 'View in Hub' button is visible next to the image row.

We can directly run in Docker Desktop too and check the output as we got in the instance terminal.

The screenshot shows the Docker Desktop interface. A container named 'cranky_difflie' (sg2653/ml-spark:latest) is running. The 'Logs' tab is selected, displaying the following log output:

```

2024-04-27 18:39:11 24/04/27 22:39:11 INFO SparkEnv: Registering OutputCommitCoordinator
2024-04-27 18:39:11 24/04/27 22:39:11 INFO JettyUtils: Start Jetty 0.0.0:4040 for SparkUI
2024-04-27 18:39:11 24/04/27 22:39:11 INFO Utils: Successfully started service 'SparkUI' on port 4040.
2024-04-27 18:39:11 24/04/27 22:39:11 INFO Executor: Starting executor ID driver on host b7117d97a72d
2024-04-27 18:39:11 24/04/27 22:39:11 INFO Executor: OS info Linux, 6.6.22-linuxkit, amd64
2024-04-27 18:39:11 24/04/27 22:39:11 INFO Executor: Java version 1.8.0_342
2024-04-27 18:39:11 24/04/27 22:39:11 INFO Executor: Starting executor with user classpath (userClassPathFirst = false): ''
2024-04-27 18:39:11 24/04/27 22:39:11 INFO Executor: Created or updated repl class loader org.apache.spark.util.MutableURLClassLoader@6ebb4161 for default.
2024-04-27 18:39:11 24/04/27 22:39:11 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 35827.
2024-04-27 18:39:11 24/04/27 22:39:11 INFO NettyBlockTransferService: Server created on b7117d97a72d:35827
2024-04-27 18:39:11 24/04/27 22:39:11 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
2024-04-27 18:39:11 24/04/27 22:39:11 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, b7117d97a72d, 35827, None)
2024-04-27 18:39:11 24/04/27 22:39:11 INFO BlockManagerMasterEndpoint: Registering block manager b7117d97a72d:35827 with 366.3 MiB RAM, BlockManagerId(driver, b7117d97a72d, 35827, None)
2024-04-27 18:39:11 24/04/27 22:39:11 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, b7117d97a72d, 35827, None)
2024-04-27 18:39:11 24/04/27 22:39:11 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, b7117d97a72d, 35827, None)
2024-04-27 18:39:27 Test Accuracy of wine prediction model = 0.96875
2024-04-27 18:39:27 /opt/spark/python/lib/pyspark.zip/pyspark/sql/context.py:158: FutureWarning: Deprecated in 3.0.0. Use SparkSession.builder.getOrCreate() instead.
2024-04-27 18:39:28 Weighted F1 Score of wine prediction model = 0.9541901629072682
2024-04-27 18:39:35 Exiting Spark Application

```

The Docker Desktop status bar at the bottom shows: Engine running, RAM 2.97 GB, CPU 0.10%, Disk 52.80 GB avail. of 62.67 GB, Signed in, v4.29.0.

Docker Hub : Latest Image Layers

The screenshot shows the Docker Hub interface for the image `sg2653/ml-spark:latest`. The top navigation bar includes the Docker Hub logo, a search bar, and links for Sign In and Sign up. Below the header, the image details are displayed: OS/ARCH is linux/amd64, Compressed Size is 1.01 GB, Last Pushed was an hour ago by [sg2653](#), Type is Image, and Manifest Digest is sha256:963... A manifest digest link is also present.

IMAGE LAYERS

```

1 ADD file ... in /           29.91 MB
2 CMD ["bash"]                 0 B
3 /bin/sh -c set -eux; apt-get 1.51 MB
4 ENV JAVA_HOME=/usr/local/openjdk-8 0 B
5 /bin/sh -c { echo '#!/bin/sh';   210 B
6 ENV PATH=/usr/local/openjdk-8/bin... 0 B
7 ENV LANG=C.UTF-8             0 B
8 ENV JAVA_VERSION=8u342        0 B
9 /bin/sh -c set -eux; arch="$... 39.76 MB
10 ENV PATH=/opt/miniconda3/bin:/usr... 0 B
11 ENV PYSPARK_PYTHON=/opt/miniconda... 0 B
12 RUN /bin/sh -c apt-get update  1.99 MB
13 RUN /bin/sh -c curl -s       202.67 MB
14 RUN /bin/sh -c pip install    364.47 MB
15 ENV SPARK_HOME=/opt/spark      0 B
16 WORKDIR /opt                  32 B
17 RUN /bin/sh -c wget --no-ver... 382.41 MB
18 RUN /bin/sh -c wget https://... 10.69 MB
19 RUN /bin/sh -c wget https://... 278.45 KB
20 COPY prediction.py /opt/ # bu... 1.04 KB
21 COPY ValidationDataset.csv /o... 2.75 KB
22 COPY spark-model /opt/spark-... 70.03 KB
23 CMD ["spark-submit" "/opt/predict... 0 B

```

A tooltip for the first command (ADD) is shown, displaying the full command: `ADD file:0eae0dca665c7044bf242cb1fc92cb8ea744f5af2dd376a558c90bc47349 in /`.

To download files from instance to local machine :

```
scp -i EMR_Cluster.pem -r  
'hadoop@ec2-18-215-161-107.compute-1.amazonaws.com:/home/hadoop/*'  
/path/to/local_machine_folder
```

GitHub Repository link for code :

<https://github.com/sainageshgowrabalaji/academic2>