

CS 643, Cloud Computing – Programming Assignment 2

Mehul Shah
ms298@njit.edu

Weblinks:

- GITHUB Repository: <https://github.com/ms298njit/CS643-PA2-WQP>
- Docker Container: <https://hub.docker.com/repository/docker/ms298/wqp>

Purpose:

The purpose of this individual assignment is to learn how to develop parallel machine learning (ML) applications in Amazon AWS cloud platform. Specifically:

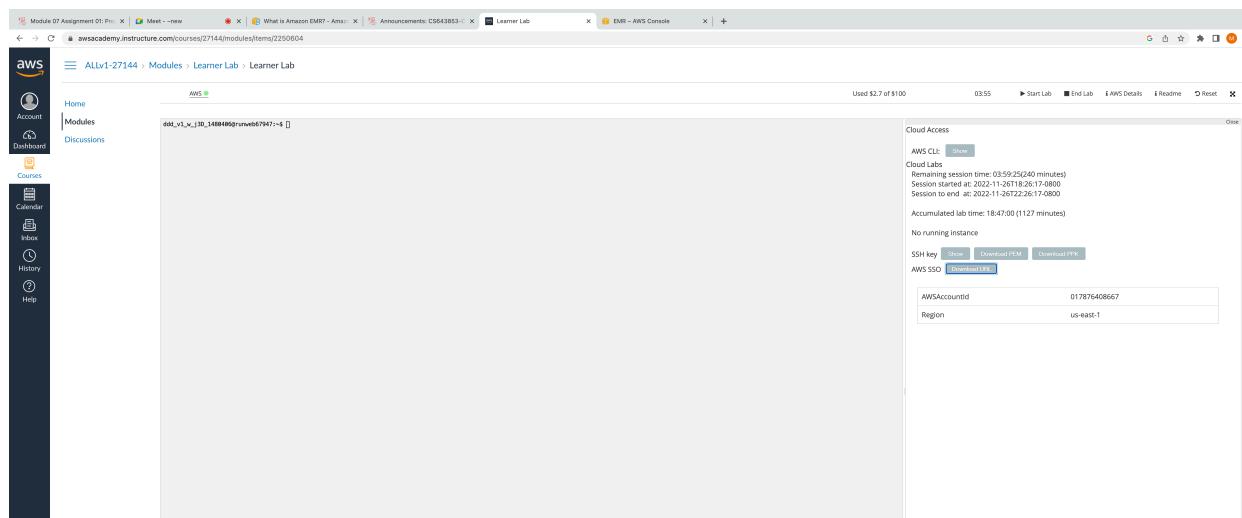
- how to use Apache Spark to train an ML model in parallel on multiple EC2 instances
- how to use Spark's MLlib to develop and use an ML model in the cloud
- how to use Docker to create a container for your ML model to simplify model deployment

Description:

Build a wine quality prediction ML model in Spark over AWS. The model must be trained in parallel using 4 EC2 instances. Then, 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.

Instructions:

1. Create spark cluster using EMR AWS. Log on to AWS academy student account, start learner lab and open EMR-AWS console.



Screenshot of the AWS Management Console Home page:

- Recently visited:** EMR, EC2, VPC, AWS Budgets, S3.
- Welcome to AWS:**
 - Getting started with AWS: Learn the fundamentals and find valuable information to get the most out of AWS.
 - Training and certification: Learn from AWS experts and advance your skills and knowledge.
 - What's new with AWS?: Discover new AWS services, features, and Regions.
- AWS Health:** Open issues: 0 (Past 7 days), Scheduled changes: 0 (Upcoming and past 7 days), Other notifications: 0 (Past 7 days). Go to AWS Health.
- Cost and usage:**
 - Current month costs: \$1.34
 - Forecasted month end costs: \$1.56 (Up 11% over last month)
 - Last month costs: \$1.40
 - Costs shown are unblended. Learn more.
- Build a solution:**
 - Launch a virtual machine (With EC2 (0 min))
 - Start a development project (With CodeStar (5 min))
 - Connect an IoT device (With AWS IoT (5 min))
 - Build using virtual servers (With Lambda (2 min))
 - Deploy a serverless microservice (With API Gateway (2 min))
 - Start migrating to AWS (With AWS Migration Services (2 min))
 - Host a static web app (With AWS Amplify Console (2 min))
- Trusted Advisor:** No recommendations. This could be because you have not run Trusted Advisor checks, or you don't have AWS Business or AWS Enterprise support plans. Go to Trusted Advisor.
- Explore AWS:** Free AWS Training, Try AWS Amplify Hosting for Free.
- Latest announcements:**
 - NOV 22 AWS announces availability of Microsoft SQL Server 2022 images on AWS.
 - NOV 23 Leia Inc. and AWS Partner Insys VT build DRIM solution with near real-time data processing.
 - NOV 23 AWS Nitro System now supports.
- Recent AWS blog posts:**
 - NOV 23 Backup SQL Server databases to Amazon RDS for MySQL.
- Applications:** Region US East (N. Virginia).

Screenshot of the Amazon Elastic MapReduce (EMR) console:

- Left sidebar:**
 - Amazon EMR
 - EMR Studio
 - EMR on EC2
 - Clusters
 - Notebooks
 - Git repositories
 - Security configurations
 - Block public access
 - VPC subnets
 - Events
 - EMR on EKS
 - Virtual clusters
 - Help
 - What's new
- Top banner:** EMR Serverless is now GA. With EMR Serverless, get the benefits of Amazon EMR such as open source compatibility, latest versions and performance optimized runtime for popular frameworks along with easy provisioning, quick job startup, automatic capacity management, and simple cost controls. Get Started with EMR Serverless.
- Welcome to Amazon Elastic MapReduce:**

Amazon Elastic MapReduce (Amazon EMR) is a web service that enables businesses, researchers, data analysts, and developers to easily and cost-effectively process vast amounts of data.

You do not appear to have any clusters. Create one now.
- Create cluster:** Upload, Create, Monitor.
- How Elastic MapReduce Works:**
 - Upload: Upload your data and processing application to S3.
 - Create: Configure and create your cluster by specifying data inputs, outputs, cluster size, security settings, etc.
 - Monitor: Monitor the health and progress of your cluster. Retrieve the output in S3.
- Additional Information:**
 - More about Elastic MapReduce
 - EMR overview
 - FAQ
 - Pricing
 - More Help Using Elastic MapReduce
 - Forum
 - Documentation
 - Developer Guide
 - API Reference
 - EMR on GitHub
 - Help portal

2. Configure Cluster parameters such as: General, Software, Hardware, Security and Access

General Configuration

Cluster name: PA2-WQP

Release: emr-5.36.0

Applications:

- Core Hadoop: Hadoop 2.10.1, Hive 2.3.9, HDFS 4.10.0, MapReduce 2.10.1, Tez 2.3.9, HBase 2.1.1, HDFS 2.10.1, Hive 2.3.9, HDFS 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

Launch mode: Cluster

Software configuration

Release: emr-5.36.0

Applications:

- Core Hadoop: Hadoop 2.10.1, Hive 2.3.9, HDFS 4.10.0, MapReduce 2.10.1, Tez 2.3.9, HBase 2.1.1, HDFS 2.10.1, Hive 2.3.9, HDFS 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

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

Cluster scaling: Scale cluster nodes based on workload

Auto-termination: Enable auto-termination

Terminate cluster when it is idle after 24 hours

Security and access

EC2 key pair: Test1

Permissions: Default

EC2 role: EMR_DefaultRole

EC2 instance profile: EMR_EC2_DefaultRole

Create cluster

Summary

ID: j-2B6JQHUYNT9M
Creation date: 2022-12-08 10:07 (UTC-5)
Elapsed time: 44 minutes

After last step completes: Cluster waits
Termination protection: Off
Type: Standard

Master public DNS: ec2-23-20-4-89.compute-1.amazonaws.com
Connect to the Master Node using SSH

Configuration details

Release label: emr-5.36.0
Hadoop distribution: Amazon
Applications: Spark 2.4.8, Zeppelin 0.10.0
Log URI: s3://aws-log-0178764867-us-east-1/elasticmapreduce/cluster/j-2B6JQHUYNT9M
EMRFS consistent view: Disabled
Custom AMI ID: --
Amazon Linux Release: 2.0.20221103.3

Application user interfaces

Persistent user interfaces: Spark history server, YARN timeline server
On-cluster user interfaces: Enable an SSH Connection interface

Network and hardware

Availability zone: us-east-1b
Subnet ID: subnet-0ab2f0b81099945e0
Master instance: Running 1 m5.xlarge
Core instances: Running 3 m5.xlarge
Task: --
Cluster scaling: Not enabled
Auto-termination: Terminate if idle for 1 day

Security and access

Key name: Test1
EC2 instance profile: EMR_EC2_DefaultRole
EC2 role: EMR_DefaultRole
Visible to all users: All Check
Security groups for Master: sg-0880304cb7847b7b (ElasticMapReduce-master)
Security groups for Core & Task: sg-0d1ac01d530ba5a7 (ElasticMapReduce-Task slave)

Create cluster

View details | Clone | Terminate

Filter: Active clusters | Filter clusters ... | 1 cluster (all loaded) C

ID	Status	Creation time (UTC-8)	Elapsed time	Normalized instance hours
j-2B6JQHUYNT9M	Waiting Cluster ready	2022-12-08 10:07 (UTC-8)	46 minutes	0

3. Edit inbound rules to allow access to the cluster through SSH

Screenshot of the AWS EC2 Management Console showing the security group configuration for the master node.

EC2 > Security Groups > sg-0882d0d4cb4784b7b - ElasticMapReduce-master

Details

Security group name	sg-ElasticMapReduce-master	Security group ID	sg-0882d0d4cb4784b7b	Description	Slave group for Elastic MapReduce created on 2022-11-26T22:17:07Z
Owner	017876408667	Inbound rules count	19 Permission entries	Outbound rules count	1 Permission entry

Inbound rules | Outbound rules | Tags

You can now check network connectivity with Reachability Analyzer

Inbound rules (19)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sgr-047e595d46fe8	IPv4	Custom TCP	TCP	8443	72.21.198.64/29	-
-	sgr-04839179c03655b	IPv4	Custom TCP	TCP	8443	72.21.198.64/29	-
-	sgr-05ab3011c03d4735b	IPv4	Custom TCP	TCP	8443	207.17.172.6/32	-
-	sgr-05d0d9951869e2...	IPv4	Custom TCP	TCP	8443	54.240.217.64/28	-
-	sgr-05a7c18abb87a5f3	-	All TCP	TCP	0 - 65535	sp-0994a9ba7f8991cc...	-
-	sgr-05ffccbb164e0106	-	All UDP	UDP	0 - 65535	sp-0994a9ba7f8991cc...	-
-	sgr-0505f6a768970e40	IPv4	SSH	TCP	22	0.0.0.0/0	-
-	sgr-0309f1a277104044	IPv4	Custom TCP	TCP	8443	54.240.217.82/29	-
-	sgr-0341310ed6c1f	-	All ICMP - IPv4	ICMP	All	sp-0994a9ba7f8991cc...	-
-	sgr-08448a203e715f	-	All ICMP - IPv4	ICMP	All	sp-0994a9ba7f8991cc...	-
-	sgr-03711521d07f1680	IPv4	Custom TCP	TCP	8443	207.17.167.25/32	-
-	sgr-09870550355e9...	IPv4	Custom TCP	TCP	8443	207.17.167.10/32	-
-	sgr-0a6e5eb0fa0f0a6a	IPv4	Custom TCP	TCP	8443	207.17.167.26/32	-
-	sgr-05d5c336ba7095f	IPv4	Custom TCP	TCP	8443	54.240.217.80/29	-
-	sgr-06666389f1a9990f	IPv4	Custom TCP	TCP	8443	54.240.217.16/29	-
-	sgr-0906fb11afe62c72	IPv4	Custom TCP	TCP	8443	54.239.98.0/24	-
-	sgr-b0b4c5d6c2a2d081...	IPv4	Custom TCP	TCP	8443	72.21.17.0/24	-
-	sgr-02d1e88be62801...	-	All TCP	TCP	0 - 65535	sp-0882d0d4cb4784b...	-
-	sgr-0972e2d7345563de	-	All UDP	UDP	0 - 65535	sp-0994a9ba7f8991cc...	-

Feedback: Looking for language selection? Find it in the new Unified Settings.

Screenshot of the AWS EC2 Management Console showing the security group configuration for the slave node.

EC2 > Security Groups > sg-0c9acc81d530a5ae7 - ElasticMapReduce-slave

Details

Security group name	sg-ElasticMapReduce-slave	Security group ID	sg-0c9acc81d530a5ae7	Description	Slave group for Elastic MapReduce created on 2022-11-26T22:17:07Z
Owner	017876408667	Inbound rules count	7 Permission entries	Outbound rules count	1 Permission entry

Inbound rules | Outbound rules | Tags

You can now check network connectivity with Reachability Analyzer

Inbound rules (7)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sgr-005241ef1b0cd0f	-	All UDP	UDP	0 - 65535	sp-0994a9ba7f8991cc...	-
-	sgr-0080122c54470a...	-	All ICMP - IPv4	ICMP	All	sp-0882d0d4cb4784b...	-
-	sgr-012838f561db2e7f	IPv4	SSH	TCP	22	0.0.0.0/0	-
-	sgr-08a044569729b94	-	All ICMP - IPv4	ICMP	All	sp-0994a9ba7f8991cc...	-
-	sgr-03ed5b12711864a	-	All TCP	TCP	0 - 65535	sp-0882d0d4cb4784b...	-
-	sgr-098a87d90c640...	-	All UDP	UDP	0 - 65535	sp-0882d0d4cb4784b...	-
-	sgr-061ee203b50c3b4...	-	All TCP	TCP	0 - 65535	sp-0994a9ba7f8991cc...	-

4. Connect to master node/instance within cluster using command:
 - ssh -i Trial1.pem hadoop@ec2-23-20-4-89.compute-1.amazonaws.com

5. copy files (TrainingDataset.csv, ValidationDataset.csv and WQP-1.0.jar) to EMR local file system and then put it on HDFS
 - scp -i Trial1.pem /Users/mehul/Desktop/NJIT/Fall\ 2022/CS643\ -\ Cloud\ Computing/Programming\ Assignment\ 2/WQP/data/* hadoop@ec2-23-20-4-89.compute-1.amazonaws.com:/home/hadoop
 - scp -i Trial1.pem /Users/mehul/Desktop/NJIT/Fall\ 2022/CS643\ -\ Cloud\ Computing/Programming\ Assignment\ 2/WQP/target/WQP-1.0.jar hadoop@ec2-23-20-4-89.compute-1.amazonaws.com:/home/hadoop
 - hadoop fs -put * .

```
~$ Desktop/NJIT/Fall 2022/CS643 - Cloud Computing/Programming Assignment 2 -- hadoop@ip-172-31-46-84:~ -- ssh -i Trial1.pem hadoop@ec2-23-20-4-89.compute-1.amazonaws.com
[base] Mehul-iMac:Programming Assignment 2 mehul$ scp -i Trial1.pem /Users/mehul/Desktop/NJIT/Fall1_2022/CS643\ -\ Cloud\ Computing/Programming\ Assignment\ 2/WQP/data/* hadoop@ec2-23-20-4-89.compute-1.amazonaws.com:/home/hadoop
TrainingDataset.csv: [Errno 2] No such file or directory
ValidationDataset.csv: [Errno 2] No such file or directory
[base] Mehul-iMac:Programming Assignment 2 mehul$ scp -i Trial1.pem /Users/mehul/Desktop/NJIT/Fall1_2022/CS643\ -\ Cloud\ Computing/Programming\ Assignment\ 2/WQP/target/WQP-1.0.jar hadoop@ec2-23-20-4-89.compute-1.amazonaws.com:/home/hadoop
WQP-1.0.jar: [Errno 2] No such file or directory
[base] Mehul-iMac:Programming Assignment 2 mehul$
```

```
~$ Desktop/NJIT/Fall 2022/CS643 - Cloud Computing/Programming Assignment 2 -- hadoop@ip-172-31-46-84:~ -- ssh -i Trial1.pem hadoop@ec2-23-20-4-89.compute-1.amazonaws.com
[base] Mehul-iMac:Programming Assignment 2 mehul$ ssh -i Trial1.pem hadoop@ec2-23-20-4-89.compute-1.amazonaws.com
Last login: Thu Dec  8 15:51:08 2022
__| _--|_
 _| (   _/   Amazon Linux 2 AMI
___| \_\_|_____|

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

E:EEEEEEEEE~~~~~E~~~~~E~~~~~M~~~~~MM~~~~~RRRRRRRRRRRRR
E:|||||:|||||:|||||:||: M:|||||:M M:|||||:M R:|||||:|||||:R
EE:|||||:EEEEEEEEE:||: E: M:|||||:M M:|||||:M R:|||||:RRRRR:||:R
E:||:E EEEEE M:|||||:M M:|||||:M RR:||:R R:||:R
E:||:E M:|||||:M:||:M M:||:M:|||||:M R:||:R R:||:R
E:|||||:EEEEEEEEE M:|||||:M M:||:M M:|||||:M R:|||||:RRRRR:||:R
E:|||||:M:||:M:||:M M:|||||:M M:|||||:M R:|||||:|||||:RR
E:|||||:EEEEEEEEE M:|||||:M M:|||||:M M:|||||:M R:|||||:RRRRR:||:R
E:||:E M:|||||:M M:|||||:M M:|||||:M R:||:R R:||:R
E:||:E EEEEE M:|||||:M MMM M:|||||:M R:||:R R:||:R
EE:|||||:EEEEEEEEE:||: M:|||||:M M:|||||:M R:||:R R:||:R
E:|||||:M:||:M:||:M M:|||||:M M:|||||:M R:||:R R:||:R
E:|||||:EEEEEEEEE:||: M:|||||:M M:|||||:M R:||:R R:||:R
EEEEEEEEE~~~~~E~~~~~E~~~~~M~~~~~MM~~~~~RRRRRRRR RRRRRR

[hadoop@ip-172-31-46-84 ~]$ ls
TrainingDataset.csv ValidationDataset.csv WQP-1.0.jar
[hadoop@ip-172-31-46-84 ~]$ hadoop fs -put *
[hadoop@ip-172-31-46-84 ~]$ hadoop fs -put *
put: 'TrainingDataset.csv': File exists
put: 'ValidationDataset.csv': File exists
put: 'WQP-1.0.jar': File exists
[hadoop@ip-172-31-46-84 ~]$
```

6. Create and run spark application to train ML model in parallel on 4 EC2 instances.

- Read TrainingDataset.csv, ValidationDataset.csv from HDFS.
 - Create trained ML model onto HDFS.
 - Run WQP-1.0.Jar file with three arguments: 1st Argument = TrainingDataset.csv, 2nd Argument = ValidationDataset.csv, 3rd Argument = trained ML model path
 - spark-submit --deploy-mode cluster --class winequality.MLmodelTrainer WQP-1.0.jar hdfs://ip-172-31-46-84.ec2.internal:8020/user/hadoop/TrainingDataset.csv hdfs://ip-172-31-46-84.ec2.internal:8020/user/hadoop/ValidationDataset.csv hdfs://ip-172-31-46-84.ec2.internal:8020/user/hadoop/model/
 - hadoop fs -ls



Lerner Lab | EMR – AWS Console | Wine Quality Model - Spark Job | Wine Quality Model application UI

Jobs Stages Storage Environment Executors SQL

Spark Jobs (7)

User Task Log
Total Updates: 23
Scheduling Mode: FIFO
Completed Jobs: 18

Event Timeline
Completed Jobs (18)

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
17	parquet at LinearRegression.scala:747	2022/12/08 16:10:25	0.6 s	1/1 (1 skipped)	1/1 (1 skipped)
16	parquet at LinearRegression.scala:747	2022/12/08 16:10:24	24 ms	1/1	1/1
15	parquet at LinearRegression.scala:747	2022/12/08 16:10:24	24 ms	1/1	1/1
14	runJob at SparkHadoopWriter.scala:78	2022/12/08 16:10:24	0.2 s	1/1	1/1
13	treeAggregate at RegressionMetrics.scala:57	2022/12/08 16:10:23	0.7 s	2/2 (1 skipped)	2/3 (2 skipped)
12	show at MLModelTrainer.java:57	2022/12/08 16:10:23	0.2 s	1/1 (1 skipped)	20/29 (1 skipped)
11	show at MLModelTrainer.java:57	2022/12/08 16:10:23	91 ms	1/1 (1 skipped)	4/4 (1 skipped)
10	show at MLModelTrainer.java:57	2022/12/08 16:10:22	0.8 s	2/2	2/2
9	csv at MLModelTrainer.java:89	2022/12/08 16:10:21	45 ms	1/1	1/1
8	csv at MLModelTrainer.java:89	2022/12/08 16:10:21	35 ms	1/1	1/1
7	count at LinearRegression.scala:953	2022/12/08 16:10:21	49 ms	1/1 (2 skipped)	1/1 (0/1 skipped)
6	count at LinearRegression.scala:953	2022/12/08 16:10:21	0.5 s	1/1 (1 skipped)	200/29 (1 skipped)
5	sum at RegressionMetrics.scala:71	2022/12/08 16:10:20	0.4 s	1/1 (1 skipped)	200/29 (1 skipped)
4	treeAggregate at RegressionMetrics.scala:57	2022/12/08 16:10:19	0.8 s	2/2 (1 skipped)	2/3 (2 skipped)
3	treeAggregate at WeightedLeastSquares.scala:105	2022/12/08 16:10:15	3 s	2/2 (1 skipped)	2/3 (2 skipped)
2	first at LinearRegression.scala:322	2022/12/08 16:10:14	1 s	2/2	2/2
1	csv at MLModelTrainer.java:89	2022/12/08 16:10:12	1 s	1/1	1/1
0	csv at MLModelTrainer.java:89	2022/12/08 16:10:08	4 s	1/1	1/1

Lerner Lab | EMR – AWS Console | Wine Quality Model - Stages | Wine Quality Model application UI

Stages for All Jobs

Completed Stages: 23
Skipped Stages: 10

Completed Stages (23)

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
32	parquet at LinearRegression.scala:747	+2022/12/08 16:10:25	0.6 s	1/1	2026.0 B	186.0 B		
30	parquet at LinearRegression.scala:747	+2022/12/08 16:10:24	15 ms	1/1			186.0 B	
29	runJob at SparkHadoopWriter.scala:78	+2022/12/08 16:10:24	45 ms	1/1		510.0 B		
28	runJob at SparkHadoopWriter.scala:78	+2022/12/08 16:10:24	0.1 s	1/1		202.0 B		
27	treeAggregate at RegressionMetrics.scala:57	+2022/12/08 16:10:24	84 ms	13/13			79.7 kB	
26	treeAggregate at RegressionMetrics.scala:57	+2022/12/08 16:10:23	0.6 s	200/290	6.2 KB		12.2 kB	79.7 kB
24	show at MLModelTrainer.java:57	+2022/12/08 16:10:23	0.2 s		20/20		2.2 kB	
22	show at MLModelTrainer.java:57	+2022/12/08 16:10:23	80 ms		4/4		198.0 B	
20	show at MLModelTrainer.java:57	+2022/12/08 16:10:23	94 ms		1/1		201.0 B	
19	show at MLModelTrainer.java:57	+2022/12/08 16:10:22	0.7 s		1/1		8.3 kB	
18	csv at MLModelTrainer.java:89	+2022/12/08 16:10:21	37 ms		1/1		8.3 kB	
17	csv at MLModelTrainer.java:89	+2022/12/08 16:10:21	32 ms		1/1		8.3 kB	
16	count at LinearRegression.scala:953	+2022/12/08 16:10:21	45 ms		1/1		11.5 kB	
13	count at LinearRegression.scala:953	+2022/12/08 16:10:21	0.4 s		200/290	217.6 kB		11.5 kB
11	sum at RegressionMetrics.scala:71	+2022/12/08 16:10:20	0.4 s		200/290	235.8 kB		
9	treeAggregate at RegressionMetrics.scala:57	+2022/12/08 16:10:20	99 ms		13/13		97.2 kB	
8	treeAggregate at RegressionMetrics.scala:57	+2022/12/08 16:10:19	0.7 s		200/290	226.1 kB		97.2 kB
6	treeAggregate at WeightedLeastSquares.scala:105	+2022/12/08 16:10:19	0.1 s		13/13		207.5 kB	
5	treeAggregate at WeightedLeastSquares.scala:105	+2022/12/08 16:10:15	3 s		200/290	1040.0 B	81.5 kB	207.6 kB
3	first at LinearRegression.scala:322	+2022/12/08 16:10:15	0.5 s		1/1		380.0 B	
2	first at LinearRegression.scala:322	+2022/12/08 16:10:14	0.5 s		1/1		65.8 kB	81.9 kB
1	csv at MLModelTrainer.java:89	+2022/12/08 16:10:12	1 s		1/1		65.8 kB	
0	csv at MLModelTrainer.java:89	+2022/12/08 16:10:08	2 s		1/1		64.0 kB	

Skipped Stages (10)

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
31	parquet at LinearRegression.scala:747	+details	Unknown	0/1				
25	show at MLModelTrainer.java:57	+details	Unknown	0/1				
23	show at MLModelTrainer.java:57	+details	Unknown	0/1				
21	show at MLModelTrainer.java:57	+details	Unknown	0/1				
15	count at LinearRegression.scala:953	+details	Unknown	0/200				
14	first at LinearRegression.scala:322	+details	Unknown	0/1				
12	first at LinearRegression.scala:322	+details	Unknown	0/1				
10	first at LinearRegression.scala:322	+details	Unknown	0/1				
7	first at LinearRegression.scala:322	+details	Unknown	0/1				
4	first at LinearRegression.scala:322	+details	Unknown	0/1				



Lerner Lab | EMR – AWS Console | Wine Quality Model - Executors | Wine Quality Model application UI

Executors

Show Additional Metrics

Summary

RDD Block	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Blocklisted
Active[0]	0.0 B / 11.3 GB	0.0 B	8	0	0	1076	1076	48 s (5 s)	931.2 kB	504.5 kB	0 B	0
Dead[0]	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0 ms (0 ms)	0 B	0 B	0 B	0
Total[0]	0.0 B / 11.3 GB	0.0 B	8	0	0	1076	1076	48 s (3 s)	931.2 kB	504.5 kB	0 B	0

Executors

Show 20 entries

Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Logs
driver	ip-172-31-32-142.ec2.internal:40029	Active	0	0.0 B / 1.1 GB	0.0 B	0	0	0	0	0	0 ms (0 ms)	0 B	0 B	0 B	stout, driver
1	ip-172-31-38-42.ec2.internal:32923	Active	0	0.0 B / 5.1 GB	0.0 B	4	0	0	199	199	21 s (2 s)	151.5 kB	82 kB	69.5 kB	stout, storer
2	ip-172-31-44-143.ec2.internal:36589	Active	0	0.0 B / 5.1 GB	0.0 B	4	0	0	877	877	24 s (0.9 s)	779.8 kB	422.5 kB	435 kB	stout, storer

Showing 1 to 3 of 3 entries

Previous | Next

7. Perform wine quality prediction model without Docker on a single EC2 instance.

- spark-submit --class winequality.WineQualityPrediction WQP-1.0.jar hdfs://ip-172-31-46-84.ec2.internal:8020/user/hadoop/ValidationDataset.csv hdfs://ip-172-31-46-84.ec2.internal:8020/user/hadoop/model/

```
-/Desktop/NJITFall 2022/CS643 - Cloud Computing/Programming Assignment 2 — hadoop@ip-172-31-46-84:~ ssh -i Trial1.pem hadoop@ec2-23-20-4-89.compute-1.amazonaws.com -/Desktop/NJIT
[hadoop@ip-172-31-46-84 ~]$ spark-submit --class winequality.WineQualityPrediction WQP-1.0.jar hdfs://ip-172-31-46-84.ec2.internal:8020/user/hadoop/ValidationDataset.csv hdfs://ip-172-31-46-84.ec2.internal:8020/user/hadoop/model/
22/12/08 16:21:11 INFO GPLNativeModelLoader: Loaded native gpl library
22/12/08 16:21:11 INFO LzoDecoded: Successfully loaded & initialized native-lzo library [hadoop-lzo rev 8493e2b7cf53ff5f739d6b1532457f2c6cd495e8]
=====
features|quality| prediction
[16.7, 0.6, 0.17, 2.3,...] | 6 | 5.776933061582862
[18.3, 0.42, 0.38, 2,...] | 6 | 5.826299281747663
[18.0, 0.48, 0.24, 2,...] | 6 | 5.482588722334893
[19.0, 0.5, 0.2, 2.4,...] | 6 | 5.482588722334893
[15.0, 1.02, 0.84, 1,...] | 4 | 4.677494723976885
[19.3, 0.715, 0.24, 2,...] | 5 | 5.499135180743839
[17.8, 0.46, 0.26, 1,...] | 6 | 5.318677113469292
[10.5, 0.5, 0.2, 2.1,...] | 6 | 5.482588722334893
[19.3, 0.39, 0.44, 2,...] | 5 | 6.029804263395965
[17.6, 0.9, 0.86, 2.5,...] | 5 | 4.994329279481949
[14.0, 0.5, 0.2, 2.1,...] | 6 | 5.482588722334893
[18.1, 0.545, 0.18, 1,...] | 6 | 5.158687763016828
[17.4, 0.66, 0.0, 1.8,...] | 5 | 5.015281577188312
[17.0, 0.62, 0.0, 1.8,...] | 6 | 5.015281577188312
[10.5, 0.5, 0.5, 2.1,...] | 5 | 5.829840592615384
[11.0, 0.32, 0.55, 2,...] | 7 | 6.23843699111377
[18.5, 0.51, 0.64, 2,...] | 7 | 6.117838445888642
[12.0, 0.5, 0.2, 2.1,...] | 6 | 5.482588722334893
[13.0, 0.38, 0.49, 2,...] | 5 | 5.76212925888276486
[18.0, 0.33, 0.53, 2,...] | 6 | 5.542981595484144
=====
only showing top 20 rows
Mean absolute error:0.5294072418885698
[hadoop@ip-172-31-46-84 ~]$
```

8. Perform wine quality prediction model with Docker on a single EC2 instance.

- clone GitHub repository using command:
git clone https://github.com/ms298njit/CS643-PA2-WQP.git
- Install docker on the EC2 instance
- Link to docker image using command:
docker pull ms298/wqp
- run command to start the spark cluster:
docker-compose up -d
- Copy files to HDFS
docker cp data/model spark-master:/opt/workspace && docker cp data/ValidationDataset.csv spark-master:/opt/workspace/ ValidationDataset.csv
- Once the cluster is up and running, run command to build ML model and generate output:
docker run --rm -it --network winequality -v hadoop-distributed-file-system:/opt/workspace --name WQP-test --link spark-master:spark-master ms298/wqp