

Spark based LINEAR REGRESSION Algorithm & Sample Linear Data set in HDFS Cluster

Hadoop multi node , Spark cluster



Presented to
Animesh Chaturvedi

Agenda

- Teammates
- About Hadoop
- Hadoop Setup
- About Spark
- Spark Setup
- Algorithm
- Executing using Spark on the HDFS cluster

Teammates

- J . Anil Kumar - 21BDS024
- R . Vinay Kumar - 21BDS056
- Sidharth Kaushik - 21BDS064



Hadoop:

Hadoop is a distributed computing platform that uses clusters of commodity hardware to store and process large amounts of data. It is based on the Hadoop Distributed File System (HDFS) and the MapReduce programming model. Hadoop enables scalable, reliable, and efficient processing of big data.

Single - Node :

Hadoop single node refers to running Hadoop on a single machine, rather than a cluster. It is useful for development, testing and learning purposes. Setting up Hadoop single node involves installing Hadoop and configuring it to run on a local machine, with all Hadoop components running on a single node.

Multi - Node:

Hadoop multi-node refers to a setup where multiple machines work together in a Hadoop cluster to store and process large datasets. This setup enables parallel processing, fault tolerance, and high availability of data. The multi-node configuration includes master and worker nodes, with each node performing specific roles in the Hadoop ecosystem.

HDFS Setup



Configure Hadoop :

Modify the configuration in following "XML" files :

- core-site.xml
- hdfs-site.xml
- yarn-site.xml
- mapred-site.xml

Setup Java path :

Setup the java path in the following files:

- hadoop-env.sh
 - yarn-env.sh
 - mapred-env.sh
-

Files

```
hdfs-site.xml
~/hadoop-3.3.4/etc/hadoop

1 <?xml version="1.0" encoding="UTF-8"?>
2 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3 <!--
4   Licensed under the Apache License, Version 2.0 (the "License");
5   you may not use this file except in compliance with the License.
6   You may obtain a copy of the License at
7
8       http://www.apache.org/licenses/LICENSE-2.0
9
10  Unless required by applicable law or agreed to in writing, software
11  distributed under the License is distributed on an "AS IS" BASIS,
12  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13  See the License for the specific language governing permissions and
14  limitations under the License. See accompanying LICENSE file.
15 -->
16
17 <!-- Put site-specific property overrides in this file. -->
18
19 <configuration>
20   <property>
21     <name>dfs.replication</name>
22     <value>1</value>
23   </property>
24
25   <property>
26     <name>dfs.namenode.name.dir</name>
27     <value>/home/hadoop/hadoop-3.3.4/etc/hadoop/name/namenode</value>
28   </property>
29   <property>
30     <name>dfs.datanode.data.dir</name>
31     <value>/home/hadoop/hadoop-3.3.4/etc/hadoop/data/datanode</value>
32   </property>
33 </configuration>
```

```
core-site.xml
~/hadoop-3.3.4/etc/hadoop

hdfs-site.xml
core-site.xml

1 <?xml version="1.0" encoding="UTF-8"?>
2 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3 <!--
4   Licensed under the Apache License, Version 2.0 (the "License");
5   you may not use this file except in compliance with the License.
6   You may obtain a copy of the License at
7
8       http://www.apache.org/licenses/LICENSE-2.0
9
10  Unless required by applicable law or agreed to in writing, software
11  distributed under the License is distributed on an "AS IS" BASIS,
12  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13  See the License for the specific language governing permissions and
14  limitations under the License. See accompanying LICENSE file.
15 -->
16
17 <!-- Put site-specific property overrides in this file. -->
18
19 <configuration>
20   <property>
21     <name>fs.defaultFS</name>
22     <value>hdfs://master:9000</value>
23   </property>
24   <name>hadoop.proxyuser.dataflair.groups</name> <value>*</value>
25   </property>
26   <property>
27     <name>hadoop.proxyuser.dataflair.hosts</name> <value>*</value>
28   </property>
29   <property>
30     <name>hadoop.proxyuser.server.hosts</name> <value>*</value>
31   </property>
32   <property>
33     <name>hadoop.proxyuser.server.groups</name> <value>*</value>
34   </property>
35
```

```
mapred-site.xml
~/hadoop-3.3.4/etc/hadoop

hdfs-site.xml
core-site.xml
mapred-site.xml

1 <?xml version="1.0"?>
2 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3 <!--
4   Licensed under the Apache License, Version 2.0 (the "License");
5   you may not use this file except in compliance with the License.
6   You may obtain a copy of the License at
7
8       http://www.apache.org/licenses/LICENSE-2.0
9
10  Unless required by applicable law or agreed to in writing, software
11  distributed under the License is distributed on an "AS IS" BASIS,
12  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13  See the License for the specific language governing permissions and
14  limitations under the License. See accompanying LICENSE file.
15 -->
16
17 <!-- Put site-specific property overrides in this file. -->
18
19 <configuration>
20   <property>
21     <name>mapreduce.framework.name</name> <value>yarn</value>
22   </property>
23   <property>
24     <name>mapreduce.application.classpath</name>
25     <value>${HADOOP_MAPRED_HOME}/share/hadoop/mapreduce/*:${HADOOP_MAPRED_HOME}/share/hadoop/mapreduce/
26       lib/*</value>
27   </property>
28
29   <property>
30     <name>mapreduce.framework.name</name>
31     <value>yarn</value>
32   </property>
33 </configuration>
```

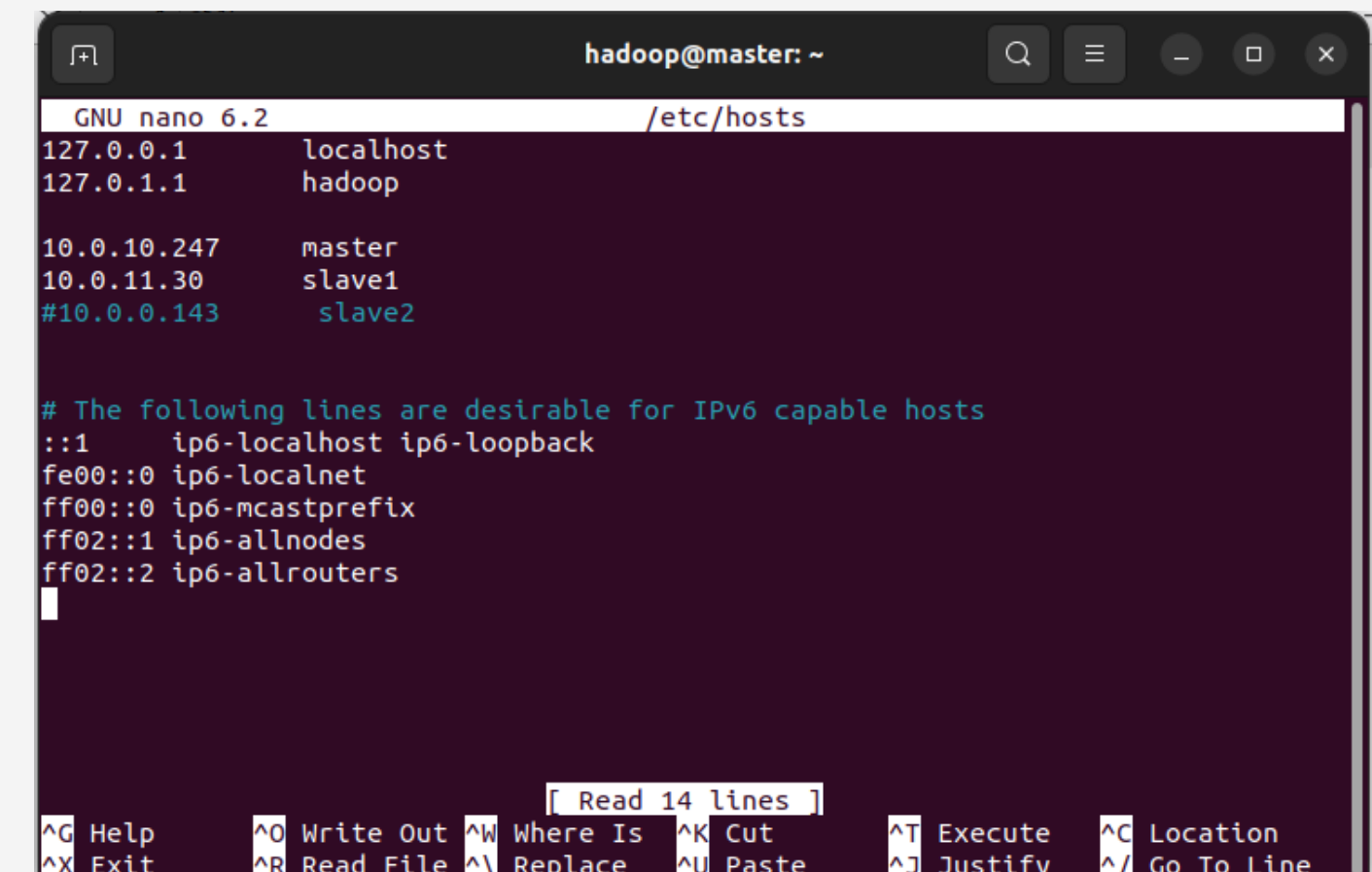
```
yarn-site.xml
~/hadoop-3.3.4/etc/hadoop

1 <?xml version="1.0"?>
2 <!--
3   Licensed under the Apache License, Version 2.0 (the "License");
4   you may not use this file except in compliance with the License.
5   You may obtain a copy of the License at
6
7       http://www.apache.org/licenses/LICENSE-2.0
8
9   Unless required by applicable law or agreed to in writing, software
10  distributed under the License is distributed on an "AS IS" BASIS,
11  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
12  See the License for the specific language governing permissions and
13  limitations under the License. See accompanying LICENSE file.
14 -->
15 <configuration>
16
17 <!-- Site specific YARN configuration properties -->
18   <property>
19     <name>yarn.nodemanager.aux-services</name>
20     <value>mapreduce_shuffle</value>
21   </property>
22   <property>
23     <name>yarn.nodemanager.env-whitelist</name>
24     <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PRE
25       END_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
26   </property>
27
28   <property>
29     <name>yarn.nodemanager.aux-services</name> <value>mapreduce_shuffle</value>
30   </property>
31   <property>
32     <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
33     <value>org.apache.hadoop.mapred.ShuffleHandler</value>
34   </property>
35   <property>
36     <description>The hostname of the RM.</description>
37     <name>yarn.resourcemanager.hostname</name>
38     <value>master</value>
39   </property>
40   <property>
41     <description>The address of the applications manager interface in the RM.</
42     description>
43     <name>yarn.resourcemanager.address</name>
44     <value>master:8032</value>
45   </property>
46 </configuration>
```

HDFS Setup

➤ Add IP address of nodes

Add all IP addresses of nodes under
/home/username/etc/hosts



The screenshot shows a terminal window titled 'hadoop@master: ~' with the nano 6.2 editor open to the file '/etc/hosts'. The file contains the following entries:

```
127.0.0.1    localhost
127.0.1.1    hadoop

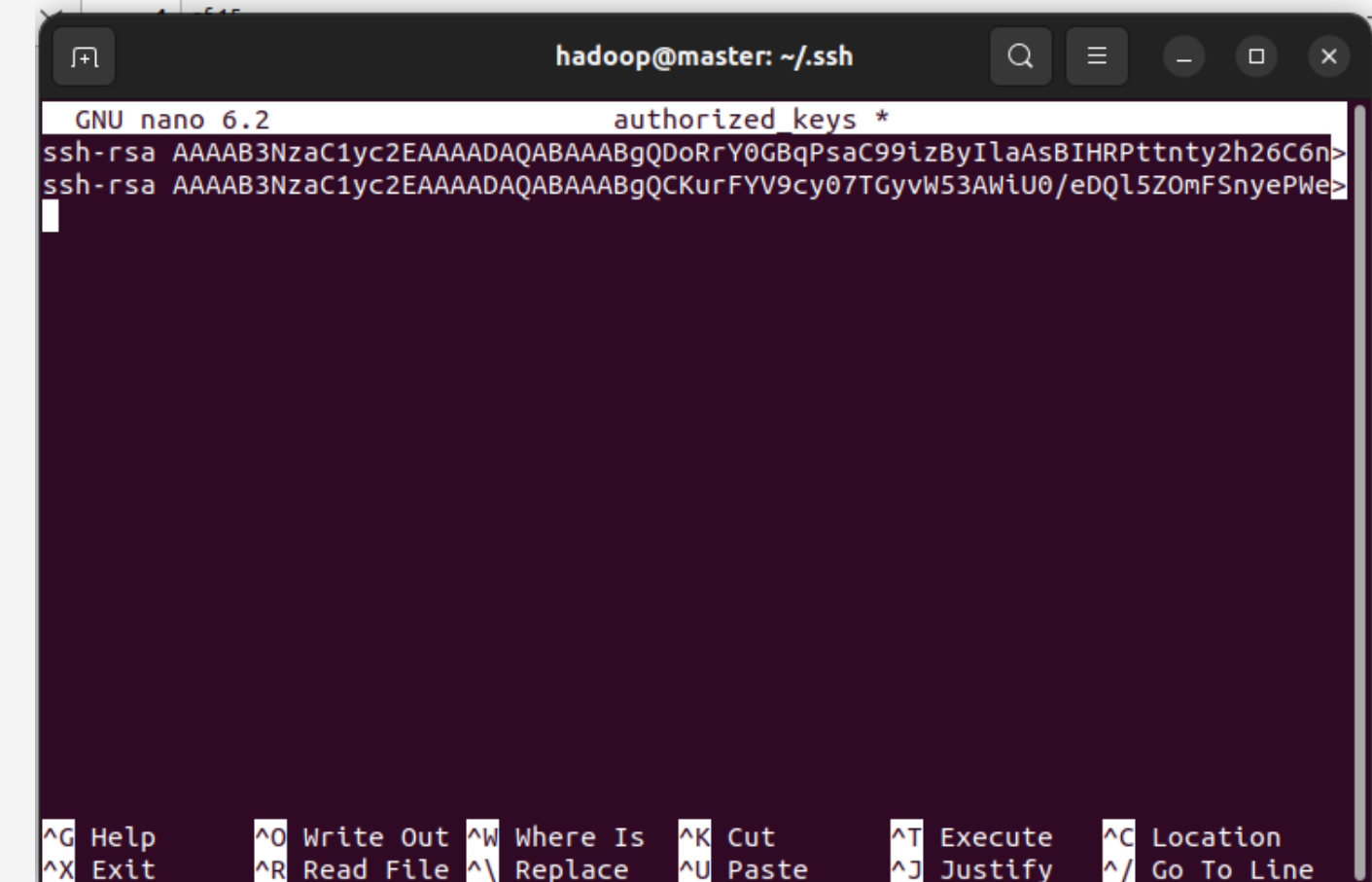
10.0.10.247   master
10.0.11.30    slave1
#10.0.0.143   slave2

# The following lines are desirable for IPv6 capable hosts
::1          ip6-localhost ip6-loopback
fe00::0      ip6-localnet
ff00::0      ip6-mcastprefix
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
```

The bottom of the screen shows the nano editor's command palette with options like Help, Write Out, Where Is, Cut, Execute, Location, Exit, Read File, Replace, Paste, Justify, and Go To Line.

➤ Setup SSH Keys of nodes

SSH keys generated by each system should
add under
/home/username/.ssh/authorized_keys



The screenshot shows a terminal window titled 'hadoop@master: ~/.ssh' with the nano 6.2 editor open to the file 'authorized_keys *'. The file contains two SSH keys:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQDoRrY0GBqPsaC99izByIlaAsBIHRPtntty2h26C6n>
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQCKurFYV9cy07TGyvW53AWiU0/eDQl5Z0mFSnyePWe>
```

The bottom of the screen shows the nano editor's command palette with options like Help, Write Out, Where Is, Cut, Execute, Location, Exit, Read File, Replace, Paste, Justify, and Go To Line.

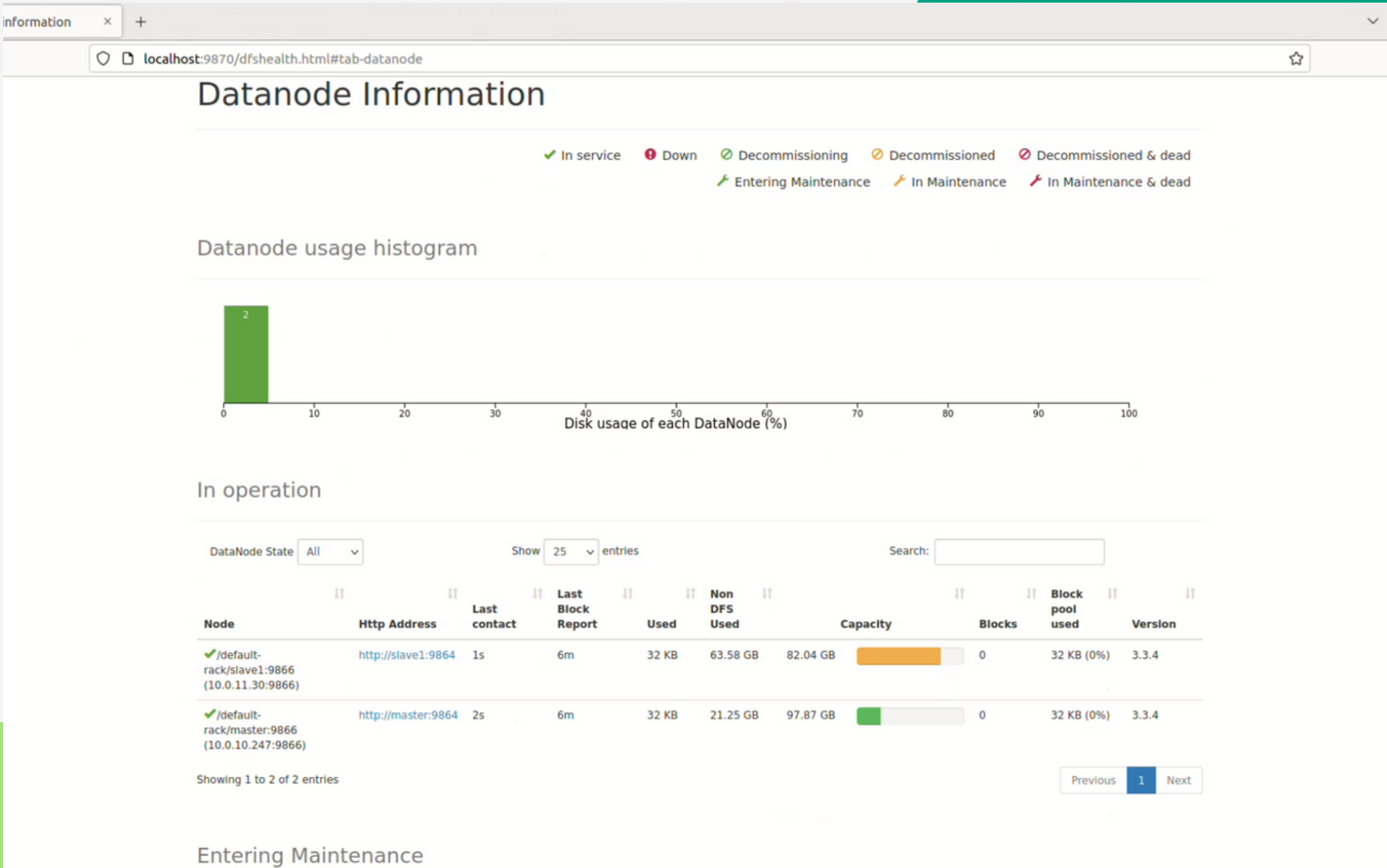
- **Start the Hadoop daemons**

Start the Hadoop deamos on the master node using this command:
"start-all.sh"

- **Verify the cluster**

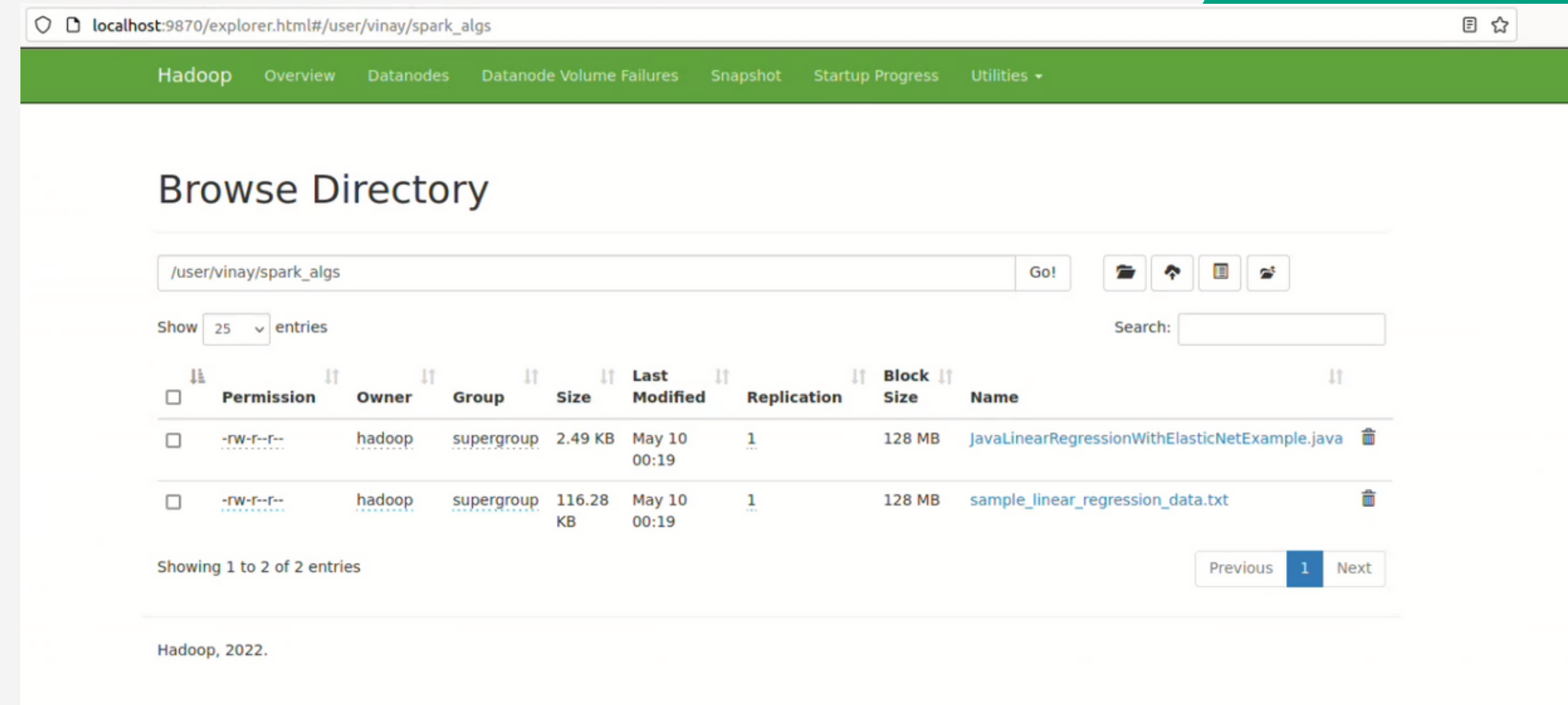
Verify the hadoop cluster is running by checking the Hadoop web page, which can be accessed at localhost

http://<master-node-IP_address>:9870



- Upload Data set to hadoop cluster

Dataset storage can be seen here
http://<master-node-IP_address>:9870



- Location of Dataset in Datanode

```
hadoop@master: ~/hadoop-3.3.4/etc/hadoop
GNU nano 6.2      hdfs-site.xml
-->
<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>

<property>
<name>dfs.namenode.name.dir</name>
<value>/home/hadoop/hadoop-3.3.4/etc/hadoop/name/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>/home/hadoop/hadoop-3.3.4/etc/hadoop/data/datanode</value>
</property>
</configuration>
^G Help      ^O Write Out ^W Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste    ^J Justify  ^_ Go To Line
```

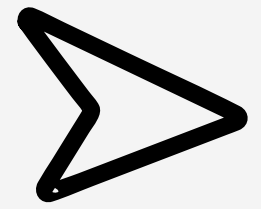
Spark:

Apache Spark is an open-source distributed computing system designed for big data processing and analytics. It provides a unified platform for batch processing, real-time streaming, graph processing, and machine learning workloads. Spark uses in-memory processing for faster data processing and provides APIs in Java, Python, Scala, and R. Spark runs on top of Hadoop Distributed File System (HDFS) and other storage systems like Amazon S3, and provides integration with various data sources such as Hive, Cassandra, and Kafka. Spark has gained popularity due to its ease of use, scalability, and performance improvements over traditional big data processing systems.

Spark cluster :

A Spark cluster is a group of computers working together to process large-scale data workloads. In a Spark cluster, the workload is distributed among the cluster nodes, enabling parallel processing for faster data processing and analytics.

Spark Setup



Configure spark


- Spark-env.sh
- Spark-defaults.conf

```
hadoop@master: /opt/spark-3.3.2-bin-hadoop3/conf
GNU nano 6.2 spark-env.sh
#!/usr/bin/env bash
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export HADOOP_CONF_DIR=/home/hadoop/hadoop-3.3.4/etc/hadoop
export YARN_CONF_DIR=/home/hadoop/hadoop-3.3.4/etc/hadoop
export SPARK_MASTER_HOST=master
export SPARK_WORKER_MEMORY=6g
#
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the license is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
# This file is sourced when running various Spark programs.
# Copy it as spark-env.sh and edit that to configure Spark for your site.
#
# Options read when launching programs locally with
# ./bin/run-example or ./bin/spark-submit
# - HADOOP_CONF_DIR, to point Spark towards Hadoop configuration files
# - SPARK_LOCAL_IP, to set the IP address Spark binds to on this node
# - SPARK_PUBLIC_DNS, to set the public dns name of the driver program
#
# Options read by executors and drivers running inside the cluster
# - SPARK_LOCAL_IP, to set the IP address Spark binds to on this node
# - SPARK_PUBLIC_DNS, to set the public DNS name of the driver program
# - SPARK_LOCAL_DIRS, storage directories to use on this node for shuffle and RDD data
# - MESOS_NATIVE_JAVA_LIBRARY, to point to your libmesos.so if you use Mesos
#
# Options read in any mode
# - SPARK_CONF_DIR, Alternate conf dir. (Default: $(SPARK_HOME)/conf)
# - SPARK_EXECUTOR_CORES, Number of cores for the executors (Default: 1).
# - SPARK_EXECUTOR_MEMORY, Memory per Executor (e.g. 1000M, 2G) (Default: 1G)
# - SPARK_DRIVER_MEMORY, Memory for Driver (e.g. 1000M, 2G) (Default: 1G)
#
# Options read in any cluster manager using HDFS
# - HADOOP_CONF_DIR, to point Spark towards Hadoop configuration files
#
# Options read in YARN client/cluster mode
# - YARN_CONF_DIR, to point Spark towards YARN configuration files when you use YARN
#
# Options for the daemons used in the standalone deploy mode
^G Help      ^O Write Out ^M Where Is  ^K Cut       ^T Execute   ^C Location  ^U Undo     ^H Set Mark  ^I To Bracket ^Q Previous  ^B Back     ^_ Prev Word
^X Exit      ^R Read File ^L Replace   ^U Paste     ^D Justify   ^/ Go To Line ^E Redo     ^C Copy      ^Q Where Was ^K Next     ^F Forward  ^_ Next Word
```

```
hadoop@master: /opt/spark-3.3.2-bin-hadoop3/conf
GNU nano 6.2 spark-defaults.conf
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the license is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
# Default system properties included when running spark-submit.
# This is useful for setting default environmental settings.
spark.master yarn

spark.master spark://master:7077
# Example:
# spark.master spark://master:7077
# spark.eventLog.enabled true
# spark.eventLog.dir hdfs://namenode:8021/directory
# spark.serializer org.apache.spark.serializer.KryoSerializer
# spark.driver.memory 5g
# spark.executor.extraJavaOptions -XX:+PrintGCDetails -Dkey=value -Dnumbers="one two three"
```

SPARK



3.3.2

Spark Master at spark://master:7077

URL: spark://master:7077

Alive Workers: 2

Cores in use: 16 Total, 0 Used

Memory in use: 20.9 GiB Total, 0.0 B Used

Resources in use:

Applications: 0 Running, 0 Completed

Drivers: 0 Running, 0 Completed

Status: ALIVE

Workers (2)

Worker Id	Address	State	Cores	Memory	Resources
worker-20230509184703-10.0.11.30-46321	10.0.11.30:46321	ALIVE	8 (0 Used)	14.3 GiB (0.0 B Used)	
worker-20230510001703-10.0.10.247-35929	10.0.10.247:35929	ALIVE	8 (0 Used)	6.6 GiB (0.0 B Used)	

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

Data Set:

Sample_Linear_Regression

A sample linear regression dataset typically contains a set of data points, where each data point has one or more independent variables and a dependent variable. The goal of linear regression is to create a linear model that can predict the value of the dependent variable based on the values of the independent variables.

```
sample_linear_regression_data[1]
File Edit View
|-9.490009878824548 1:0.4551273600657362 2:0.36644694351969087 3:-0.38256108933468047 4:-0.4458430198517267 5:0.33109790358914726 6:0.8067445293443565 7:-0.2624341731773887
8:-0.44850386111659524 9:-0.07269284838169332 10:0.5658035575800715
0.2577820163584905 1:0.8386555657374337 2:-0.1270180511534269 3:0.499812362510895 4:-0.22686625128130267 5:-0.6452430441812433 6:0.18869982177936828 7:-0.5804648622673358
8:0.651931743775642 9:-0.6555641246242951 10:0.17485476357259122
-4.438869807456516 1:0.5025608135349202 2:0.14208069682973434 3:0.16004976900412138 4:0.505019897181302 5:-0.9371635223468384 6:-0.2841601610457427 7:0.6355938616712786
8:-0.1646249064941625 9:0.9480713629917628 10:0.42681251564645817
-19.782762789614537 1:-0.0388509668871313 2:-0.4166870051763918 3:0.8997202693189332 4:0.6409836467726933 5:0.273289095712564 6:-0.26175701211620517 7:-0.2794902492677298
8:-0.1306778297187794 9:-0.08536581111046115 10:-0.05462315824828923
-7.966593841555266 1:-0.06195495876886281 2:0.6546448480299902 3:-0.6979368909424835 4:0.6677324708883314 5:-0.07938725467767771 6:-0.43885601665437957 7:-0.608071585153688
8:-0.6414531182501653 9:0.7313735926547045 10:-0.026818676347611925
-7.896274316726144 1:-0.15805658673794265 2:0.26573958270655806 3:0.3997172901343442 4:-0.3693430998846541 5:0.14324061105995334 6:-0.25797542063247825 7:0.7436291919296774
8:0.6114618853239959 9:0.2324273700703574 10:-0.25128128782199144
-8.464803554195287 1:0.39449745853945895 2:0.817229160415142 3:-0.6077058562362969 4:0.6182496334554788 5:0.2558665508269453 6:-0.07320145794330979 7:-0.38884168866510227
8:0.07981886851873865 9:0.27022202891277614 10:-0.7474843534024693
2.1214592666251364 1:-0.005346215048158909 2:-0.9453716674280683 3:-0.9270309666195007 4:-0.032312290091389695 5:0.31010676221964206 6:-0.20846743965751569 7:0.8803449313707621
8:-0.23077831216541722 9:0.29246395759528565 10:0.5409312755478819
1.0720117616524107 1:0.7880855916368177 2:0.19767407429003536 3:0.9520689432368168 4:-0.845829774129496 5:0.5502413918543512 6:-0.44235539500246457 7:0.7984106594591154
8:-0.2523277127589152 9:-0.1373808897290778 10:-0.3353514432305029
-13.772441561702871 1:-0.3697050572653644 2:-0.11452811582755928 3:-0.807098168238352 4:0.4903066124307711 5:-0.6582805242342049 6:0.6107814398427647 7:-0.7204208094262783
8:-0.8141063661170889 9:-0.9459402662357332 10:0.09666938346350307
-5.082010756207233 1:-0.43560342773870375 2:0.9349906440170221 3:0.8090021580031235 4:-0.3121157071110545 5:-0.9718883630945336 6:0.6191882496201251 7:0.0429886073795116
8:0.670311110015402 9:0.16692329718223786 10:0.37649213869502973
7.887786536531237 1:0.11276440263810383 2:-0.7684997525607482 3:0.1770172737885798 4:0.7902845707138706 5:0.2529503304079441 6:-0.23483801763662826 7:0.8072501895004851
8:0.6673992021927047 9:-0.4796127376677324 10:0.9244724404994455
14.323146365332388 1:-0.2049276879687938 2:0.1470694373531216 3:-0.48366999792166787 4:0.643491115907358 5:0.3183669486383729 6:0.22821350958477082 7:-0.023605251086149304
8:-0.2770587742156372 9:0.47596326458377436 10:0.7107229819632654
-20.057482615789212 1:-0.3205057828114841 2:0.51605972926996 3:0.45215640988181516 4:0.01712446974606241 5:0.5508198371849293 6:-0.2478254241316491 7:0.7256483175955235 8:0.39418662792516
9:-0.6797384914236382 10:0.6001217520150142
-0.8995693247765151 1:0.4508991072414843 2:0.589749448443134 3:0.6464818311502738 4:0.7005669004769028 5:0.9699584106930381 6:-0.7417466269908464 7:0.22818964839784495
8:0.08574936236270037 9:-0.6945765138377225 10:0.06915201979238828
-19.16829262296376 1:0.09798746565879424 2:-0.34288007110901964 3:0.440249350802451 4:-0.22440768392359534 5:-0.9695067570891225 6:-0.7942032659310758 7:-0.792286205517398
8:-0.6535487038528798 9:0.7952676470618951 10:-0.1622831617066689
5.601801561245534 1:0.6949189734965766 2:-0.32697929564739403 3:-0.15359663581829275 4:-0.8951865090520432 5:0.2057889391931318 6:-0.6676656789571533 7:-0.03553655732400762
8:0.14550349954571096 9:0.034600542078191854 10:0.4223352065067103
-3.2256352187273354 1:0.35278245969741096 2:0.7022211035026023 3:0.5686638754605697 4:-0.4202155290448111 5:-0.26102723928249216 6:0.010688215941416779 7:-0.4311544807877927
8:0.9500151672991208 9:0.14380635780710693 10:-0.7549354840975826
1.5299675726687754 1:-0.13079299081883855 2:0.0983382230287082 3:0.15347083875928424 4:0.45507300685816965 5:0.1921083467305864 6:0.6361110540492223 7:0.7675261182370992
8:-0.2543488202081907 9:0.2927051050236915 10:0.680182444769418
-0.250102447941961 1:-0.8062832278617296 2:0.8266289890474885 3:0.22684501241708888 4:0.1726291966578266 5:-0.6778773666126594 6:0.9993906921393696 7:0.1789490173139363
8:0.5584053824232391 9:0.03495894704368174 10:-0.8505720014852347
12.792267926563595 1:-0.008461200645088818 2:-0.648273596036564 3:-0.005334477339629995 4:0.3781469006858833 5:0.30565234666790686 6:-0.2822867492866177 7:0.10175120738413801
```


Algorithm:

Linear Regression

```
1  /*
2   * Licensed to the Apache Software Foundation (ASF) under one or more
3   * contributor license agreements. See the NOTICE file distributed with
4   * this work for additional information regarding copyright ownership.
5   * The ASF licenses this file to You under the Apache License, Version 2.0
6   * (the "License"); you may not use this file except in compliance with
7   * the License. You may obtain a copy of the License at
8   *
9   * http://www.apache.org/licenses/LICENSE-2.0
10  *
11  * Unless required by applicable law or agreed to in writing, software
12  * distributed under the License is distributed on an "AS IS" BASIS,
13  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14  * See the License for the specific language governing permissions and
15  * limitations under the License.
16  */
17
18  //package org.apache.spark.examples.ml;
19
20  // $example on$
21  import org.apache.spark.ml.regression.LinearRegression;
22  import org.apache.spark.ml.regression.LinearRegressionModel;
23  import org.apache.spark.ml.regression.LinearRegressionTrainingSummary;
24  import org.apache.spark.ml.linalg.Vectors;
25  import org.apache.spark.sql.Dataset;
26  import org.apache.spark.sql.Row;
27  import org.apache.spark.sql.SparkSession;
28  // $example off$
29
30  public class JavaLinearRegressionWithElasticNetExample {
31      public static void main(String[] args) {
32          SparkSession spark = SparkSession
33              .builder()
34              .appName("JavaLinearRegressionWithElasticNetExample")
35              .getOrCreate();
36
37          // $example on$
38          // Load training data.
39          Dataset<Row> training = spark.read().format("libsvm")
```

```
28  // $example off$
29
30  public class JavaLinearRegressionWithElasticNetExample {
31      public static void main(String[] args) {
32          SparkSession spark = SparkSession
33              .builder()
34              .appName("JavaLinearRegressionWithElasticNetExample")
35              .getOrCreate();
36
37          // $example on$
38          // Load training data.
39          Dataset<Row> training = spark.read().format("libsvm")
40              .load("hdfs://master:50000/user/sample_linear_regression_data.txt");
41
42          LinearRegression lr = new LinearRegression()
43              .setMaxIter(10)
44              .setRegParam(0.3)
45              .setElasticNetParam(0.8);
46
47          // Fit the model.
48          LinearRegressionModel lrModel = lr.fit(training);
49
50          // Print the coefficients and intercept for linear regression.
51          System.out.println("Coefficients: "
52              + lrModel.coefficients() + " Intercept: " + lrModel.intercept());
53
54          // Summarize the model over the training set and print out some metrics.
55          LinearRegressionTrainingSummary trainingSummary = lrModel.summary();
56          System.out.println("numIterations: " + trainingSummary.totalIterations());
57          System.out.println("objectiveHistory: " + Vectors.dense(trainingSummary.objectiveHistory()));
58          trainingSummary.residuals().show();
59          System.out.println("RMSE: " + trainingSummary.rootMeanSquaredError());
60          System.out.println("r2: " + trainingSummary.r2());
61          // $example off$
62
63          spark.stop();
64      }
65  }
```

iiit@master: ~/ml_algorithm/JLR

iiit@slave-1: -

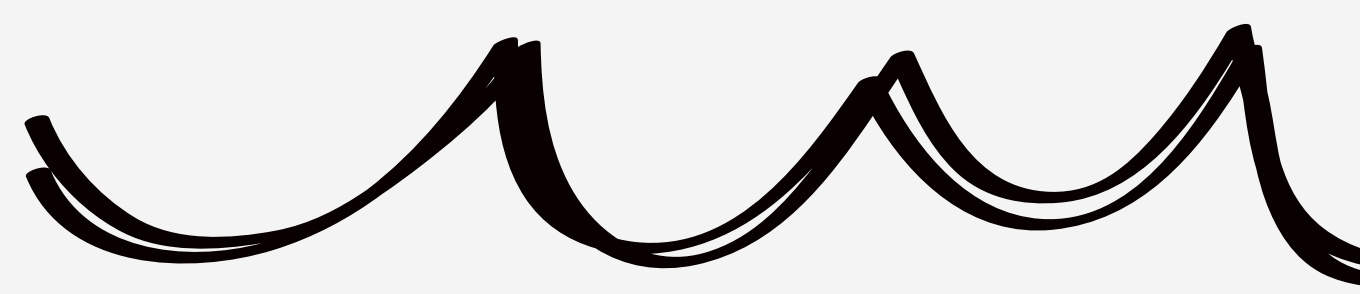
iiit@slave-2: -

iiit@master: ~/ml_algorithm

iiit@master:~/ml_algorithm/JLR\$

Output:

```
+-----+
|           residuals|
+-----+
| -9.889232683103197|
|  0.5533794340053553|
| -5.204019455758822|
| -20.566686715507508|
|  -9.4497405180564|
| -6.909112502719487|
| -10.00431602969873|
|  2.0623978070504845|
|  3.1117508432954772|
| -15.89360822941938|
| -5.036284254673026|
|  6.4832158769943335|
| 12.429497299109002|
| -20.32003219007654|
|  -2.0049838218725|
| -17.867901734183793|
|  7.646455887420495|
| -2.2653482182417406|
| -0.10308920436195645|
|  -1.380034070385301|
+-----+
only showing top 20 rows
```

Thank You

