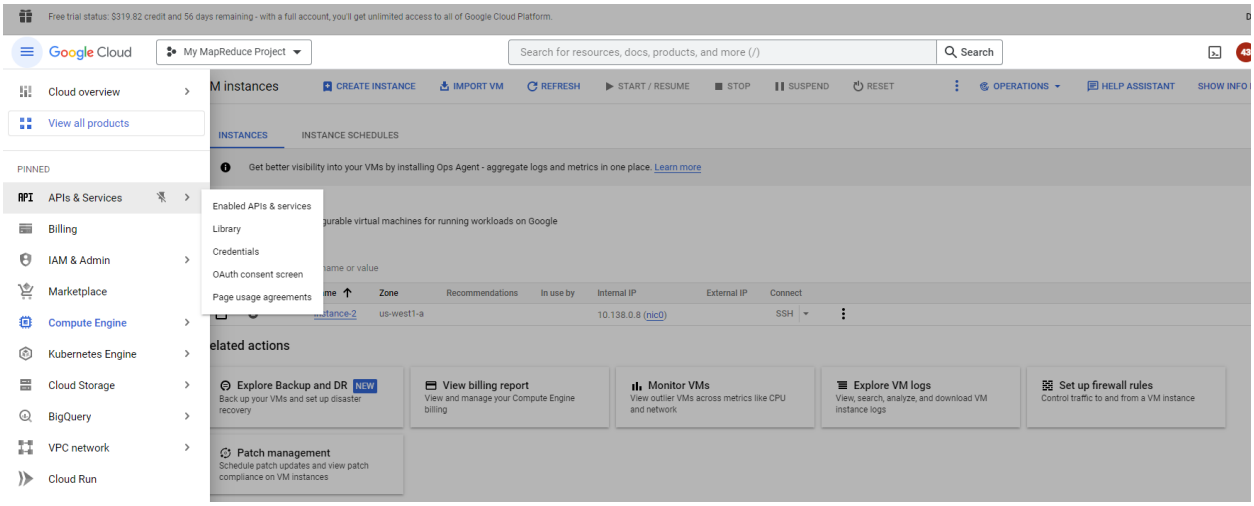
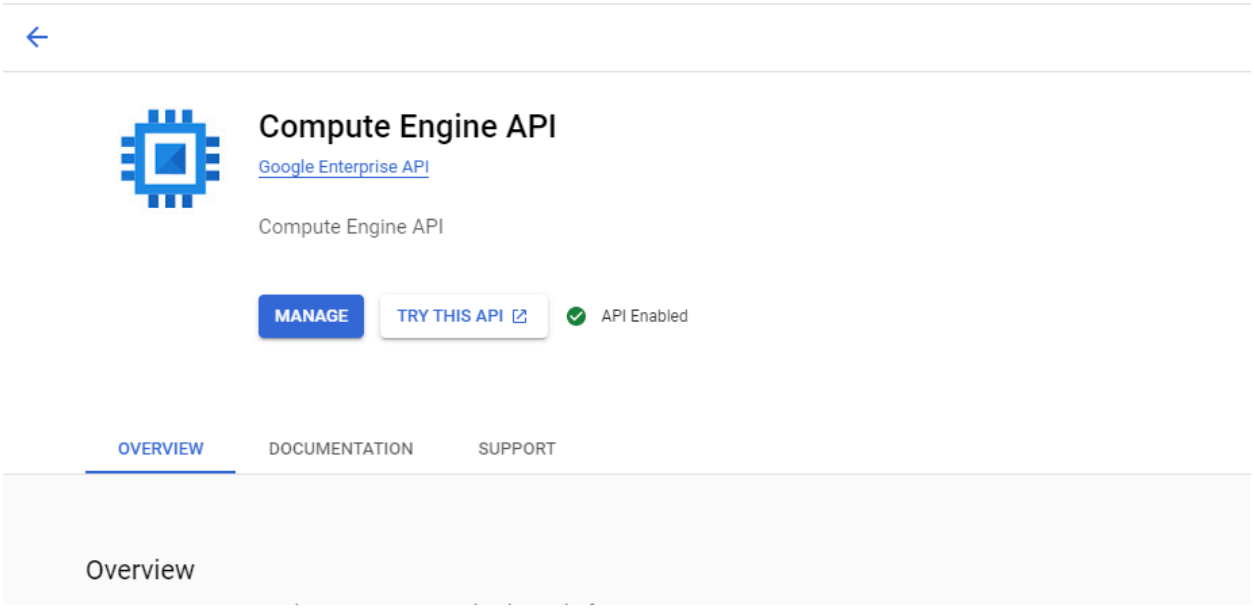
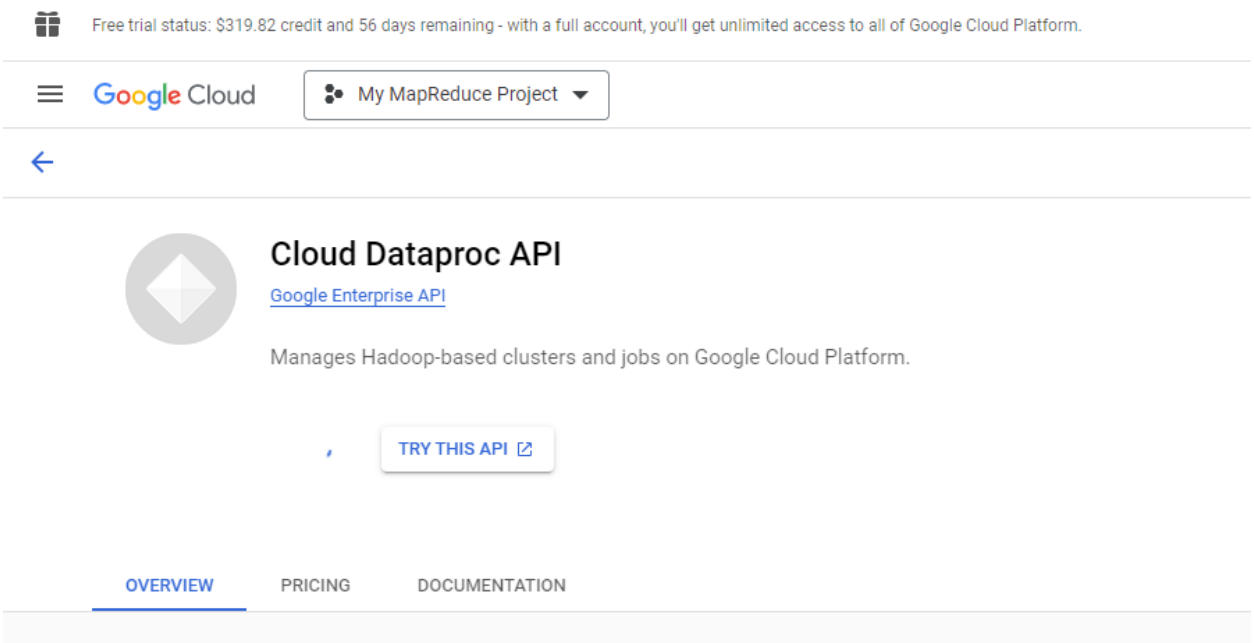
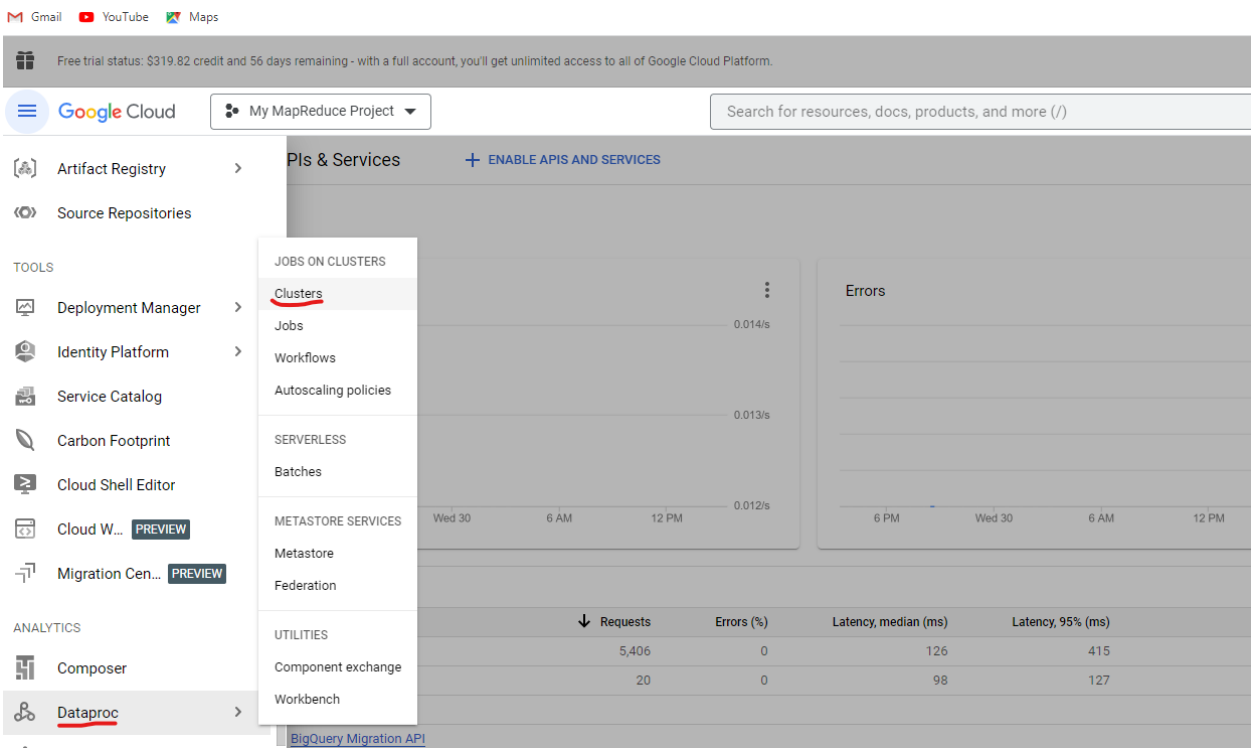
Part 1: Enable the Google Cloud Engine API and Dataproc API

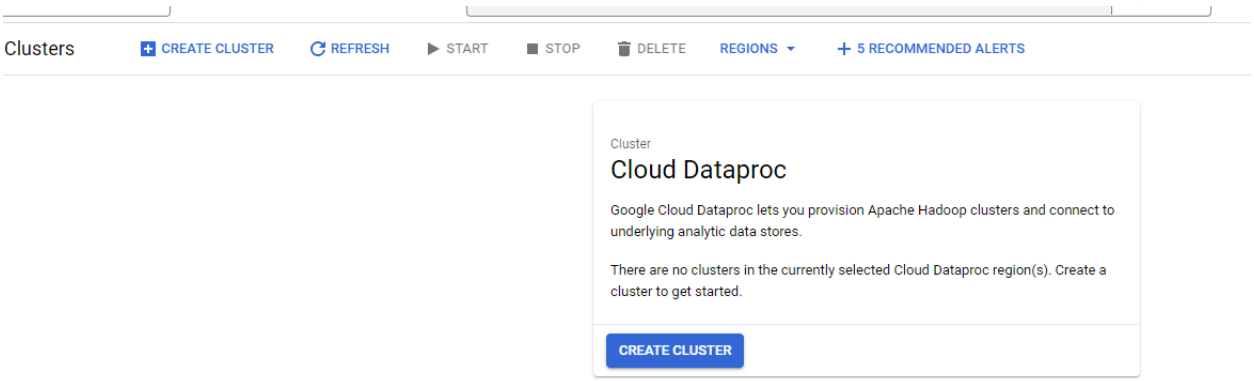


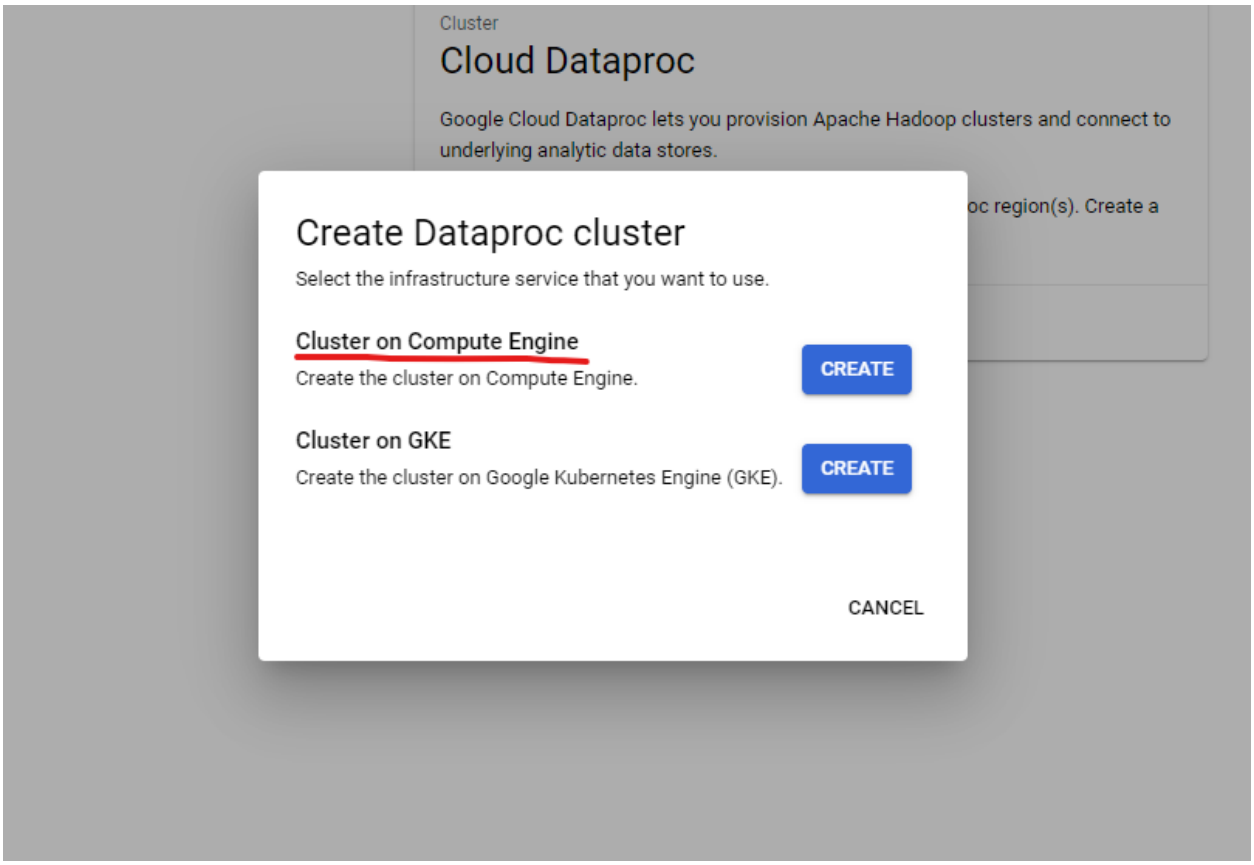


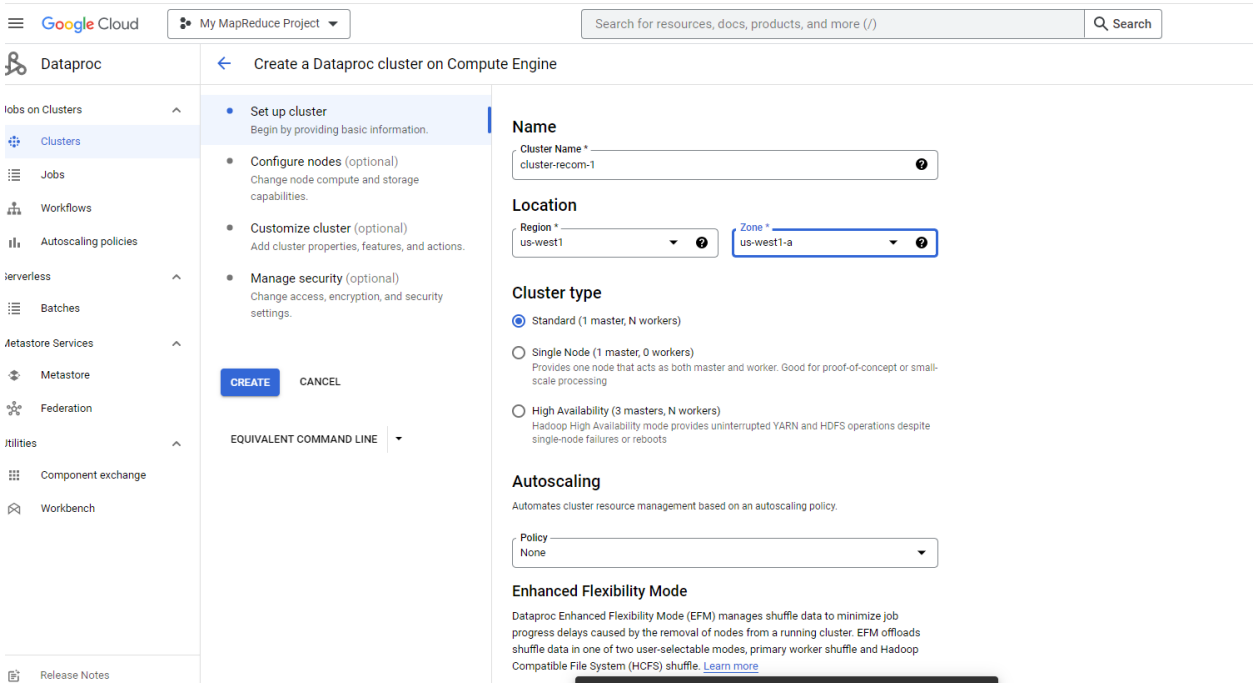


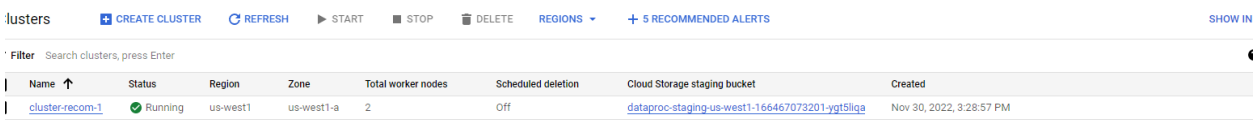
Part 2: Create a Dataproc Cluster



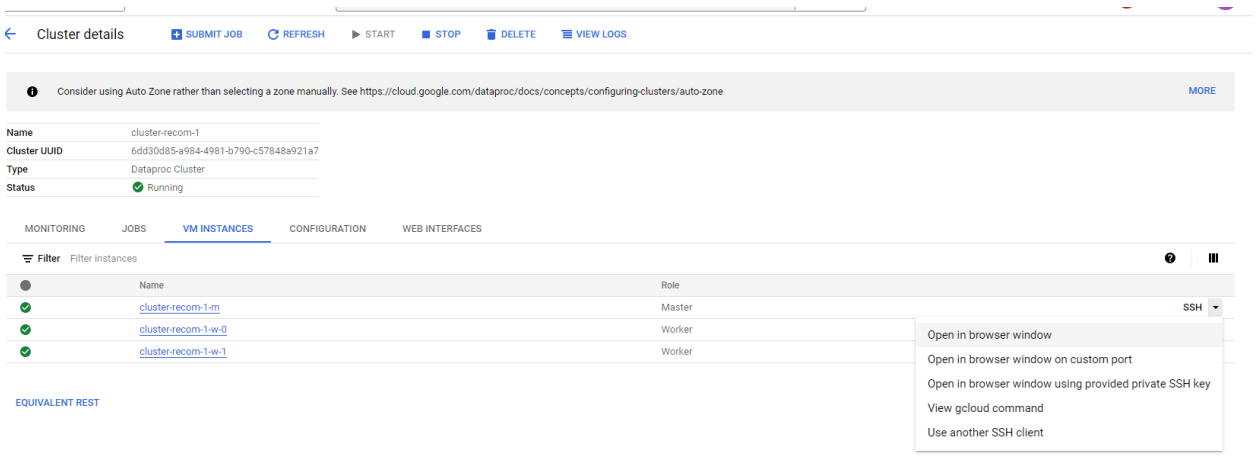

Clusters
CREATE CLUSTER
REFRESH
START
STOP
DELETE
REGIONS
5 RECOMMENDED ALERTS
Cluster
Cloud Dataproc
Google Cloud Dataproc lets you provision Apache Hadoop clusters and connect to underlying analytic data stores.
There are no clusters in the currently selected Cloud Dataproc region(s). Create a cluster to get started.
CREATE CLUSTER


Cluster
Cloud Dataproc
Google Cloud Dataproc lets you provision Apache Hadoop clusters and connect to underlying analytic data stores.
oc region(s). Create a
Create Dataproc cluster
Select the infrastructure service that you want to use.
Cluster on Compute Engine
Create the cluster on Compute Engine.
CREATE
Cluster on GKE
Create the cluster on Google Kubernetes Engine (GKE).
CREATE
CANCEL





Part 3: Connecting to the Master Node using Secure Shell (ssh)



Part 4: Download and store data in HDFS

```
https://ssh.cloud.google.com/v2/ssh/projects/my-mapreduce-project-367604/zones/us-west1-a/instances/cluster-recom-1-m?authuser=...
ssh.cloud.google.com/v2/ssh/projects/my-mapreduce-project-367604/zones/us-west1-a/instances/cluster-recom-1-m?authuser=0&hl=...
SSH-in-browser                                   UPLOAD FILE   DOWNLOAD FILE

Linux cluster-recom-1-m 5.10.0-0.deb10.16-amd64 #1 SMP Debian 5.10.127-2~bpo10+1 (2022-07-28) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
rpan@cluster-recom-1-m:~$ hdfs dfs -mkdir hdfs:///mydata
rpan@cluster-recom-1-m:~$ curl http://optionsdata.baruch.cuny.edu/data1/delivery/data/trades_sample_header.csv
| hdfs dfs -put - hdfs:///mydata/trades_sample_header.csv
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 10.9M  100 10.9M    0     0   922k      0  0:00:12  0:00:12 --:--:-- 1190k
rpan@cluster-recom-1-m:~$
```

To verify that the file is indeed located in the `mydata` folder, run the following command:

```
rpan@cluster-recom-1-m:~$ hdfs dfs -ls hdfs:///mydata
Found 1 items
-rw-r--r--   2 rpan hadoop   11473124 2022-11-30 23:40 hdfs:///mydata/trades_sample_header.csv
```

Upload and store movielens.py from local

https://ssh.cloud.google.com/v2/ssh/projects/my-mapreduce-project-367604/zones/us-west1-a/instances/cluster-recom-1-m?authuser=...

ssh.cloud.google.com/v2/ssh/projects/my-mapreduce-project-367604/zones/us-west1-a/instances/cluster-recom-1-m?authuser=0&hl=...

SSH-in-browser UPLOAD FILE DOWNLOAD FILE

```
Linux cluster-recom-1-m 5.10.0-0.deb10.16-amd64 #1 SMP Debian 5.10.127-2~bpo10+1 (2022-07-28) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec  1 00:13:29 2022 from 35.235.241.65
rpan@cluster-recom-1-m:~$ hdfs dfs -mkdir hdfs://mydata
-mkdir: java.net.UnknownHostException: mydata
Usage: hadoop fs [generic options]
        [-appendToFile <localsrc> ... <dst>]
        [-cat [-ignoreCrc] <src> ...]
        [-checksum <src> ...]
        [-chgrp [-R] GROUP PATH...]
        [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
        [-chown [-R] [OWNER][:[GROUP]] PATH...]
        [-copyFromLocal [-f] [-p] [-l] [-d] [-t <thread count>] [-q <thread pool queue size>] <localsrc> ... <d
st>]
        [-copyToLocal [-f] [-p] [-crc] [-ignoreCrc] [-t <thread count>] [-q <thread pool queue size>] <src> ...
 <localdst>]
        [-count [-q] [-h] [-v] [-t [<storage type>]] [-u] [-x] [-e] <path> ...]
        [-cp [-f] [-p | -p[topax]] [-d] [-t <thread count>] [-q <thread pool queue size>] <src> ... <dst>]
        [-createSnapshot <snapshotDir> [<snapshotName>]]
        [-deleteSnapshot <snapshotDir> <snapshotName>]
        [-df [-h] [<path> ...]]
        [-du [-s] [-h] [-v] [-x] <path> ...]
```

```
rpan@cluster-recom-1-m:~$ hdfs dfs -mkdir hdfs:///mydata
mkdir: `hdfs:///mydata': File exists
rpan@cluster-recom-1-m:~$ rm -rf mydata
rpan@cluster-recom-1-m:~$ ls
recommendation_engine_movielens.py
rpan@cluster-recom-1-m:~$ hdfs dfs -mkdir hdfs:///mydata
mkdir: `hdfs:///mydata': File exists
rpan@cluster-recom-1-m:~$ hdfs dfs -put recommendation_engine_movielens.py hdfs:///mydata
rpan@cluster-recom-1-m:~$ hdfs dfs -ls hdfs:///mydata
Found 2 items
-rw-r--r--   2 rpan hadoop       4853 2022-12-01 01:14 hdfs:///mydata/recommendation_engine_movielens.py
-rw-r--r--   2 rpan hadoop   11473124 2022-11-30 23:40 hdfs:///mydata/trades_sample_header.csv
rpan@cluster-recom-1-m:~$ pyspark
Python 3.8.13 | packaged by conda-forge | (default, Mar 25 2022, 06:04:10)
[GCC 10.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/12/01 01:14:59 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
22/12/01 01:14:59 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
22/12/01 01:14:59 INFO org.apache.spark.SparkEnv: Registering BlockManagerMasterHeartbeat
22/12/01 01:14:59 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.1.3
      /_/

Using Python version 3.8.13 (default, Mar 25 2022 06:04:10)
Spark context Web UI available at http://cluster-recom-1-m.us-west1-a.c.my-mapreduce-project-367604.internal:40
853
Spark context available as 'sc' (master = yarn, app id = application_1669853514400_0001).
SparkSession available as 'spark'.
>>>
```

rpan@cluster-recom-1-m:~$ hdfs dfs -put movies.csv hdfs:///mydata

rpan@cluster-recom-1-m:~$ hdfs dfs -put ratings.csv hdfs:///mydata

rpan@cluster-recom-1-m:~$ spark-submit recommendation_engine_movielens.py

```
rpan@cluster-recom-1-m:~$ vi recommendation_engine_movielens.py
rpan@cluster-recom-1-m:~$ vi recommendation_engine_movielens_test.py
rpan@cluster-recom-1-m:~$ spark-submit recommendation_engine_movielens_test.py
22/12/01 01:58:22 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
22/12/01 01:58:22 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
22/12/01 01:58:22 INFO org.apache.spark.SparkEnv: Registering BlockManagerMasterHeartbeat
22/12/01 01:58:22 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
22/12/01 01:58:22 INFO org.sparkproject.jetty.util.log: Logging initialized @3429ms to org.sparkproject.jetty.u
til.log.Slf4jLog
22/12/01 01:58:22 INFO org.sparkproject.jetty.server.Server: jetty-9.4.40.v20210413; built: 2021-04-13T20:42:42
.668Z; git: b881a572662e1943a14ae12e7e1207989f218b74; jvm 1.8.0_352-b08
22/12/01 01:58:22 INFO org.sparkproject.jetty.server.Server: Started @3548ms
22/12/01 01:58:22 INFO org.sparkproject.jetty.server.AbstractConnector: Started ServerConnector@60cdfcd8{HTTP/1
.1, (http/1.1)}{0.0.0.0:41397}
22/12/01 01:58:23 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at cluster-recom-1-
m/10.138.0.9:8032
22/12/01 01:58:23 INFO org.apache.hadoop.yarn.client.AHSProxy: Connecting to Application History server at clus
ter-recom-1-m/10.138.0.9:10200
22/12/01 01:58:24 INFO org.apache.hadoop.conf.Configuration: resource-types.xml not found
22/12/01 01:58:24 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find 'resource-types.xml'.
22/12/01 01:58:25 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted application application
_1669853514400_0006
22/12/01 01:58:26 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at cluster-recom-1-
m/10.138.0.9:8030
22/12/01 01:58:28 INFO com.google.cloud.hadoop.repackaged.gcs.com.google.cloud.hadoop.gcsio.GoogleCloudStorageI
mpl: Ignoring exception of type GoogleJsonResponseException; verified object already exists with desired state.
+------+-------+------+---------+
|userId|movieId|rating|timestamp|
+------+-------+------+---------+
|     1|      1|   4.0|964982703|
|     1|      3|   4.0|964981247|
|     1|      6|   4.0|964982224|
|     1|     47|   5.0|964983815|
|     1|     50|   5.0|964982931|
```

https://ssh.cloud.google.com/v2/ssh/projects/my-mapreduce-project-367604/zones/us-west1-a/instances/cluster-recom-1-m?authuser=...

ssh.cloud.google.com/v2/ssh/projects/my-mapreduce-project-367604/zones/us-west1-a/instances/cluster-recom-1-m?authuser=0&hl=...

SSH-in-browser UPLOAD FILE DOWNLOAD FILE

```
m/10.138.0.9:8030
22/12/01 01:58:28 INFO com.google.cloud.hadoop.repackaged.gcs.com.google.cloud.hadoop.gcsio.GoogleCloudStorageI
mpl: Ignoring exception of type GoogleJsonResponseException; verified object already exists with desired state.
+------+-------+------+---------+
|userId|movieId|rating|timestamp|
+------+-------+------+---------+
|     1|      1|   4.0|964982703|
|     1|      3|   4.0|964981247|
|     1|      6|   4.0|964982224|
|     1|     47|   5.0|964983815|
|     1|     50|   5.0|964982931|
|     1|     70|   3.0|964982400|
|     1|    101|   5.0|964980868|
|     1|    110|   4.0|964982176|
|     1|    151|   5.0|964984041|
|     1|    157|   5.0|964984100|
|     1|    163|   5.0|964983650|
|     1|    216|   5.0|964981208|
|     1|    223|   3.0|964980985|
|     1|    231|   5.0|964981179|
|     1|    235|   4.0|964980908|
|     1|    260|   5.0|964981680|
|     1|    296|   3.0|964982967|
|     1|    316|   3.0|964982310|
|     1|    333|   5.0|964981179|
|     1|    349|   4.0|964982563|
+------+-------+------+---------+
only showing top 20 rows

root
 |-- userId: string (nullable = true)
 |-- movieId: string (nullable = true)
 |-- rating: string (nullable = true)
 |-- timestamp: string (nullable = true)

+------+-------+------+
|userId|movieId|rating|
+------+-------+------+
|     1|      1|   4.0|
```

```
root
 |-- userId: string (nullable = true)
 |-- movieId: string (nullable = true)
 |-- rating: string (nullable = true)
 |-- timestamp: string (nullable = true)

+------+-------+------+
|userId|movieId|rating|
+------+-------+------+
|     1|      1|   4.0|
|     1|      3|   4.0|
|     1|      6|   4.0|
|     1|     47|   5.0|
|     1|     50|   5.0|
|     1|     70|   3.0|
|     1|    101|   5.0|
|     1|    110|   4.0|
|     1|    151|   5.0|
|     1|    157|   5.0|
|     1|    163|   5.0|
|     1|    216|   5.0|
|     1|    223|   3.0|
|     1|    231|   5.0|
|     1|    235|   4.0|
|     1|    260|   5.0|
|     1|    296|   3.0|
|     1|    316|   3.0|
|     1|    333|   5.0|
|     1|    349|   4.0|
+------+-------+------+
only showing top 20 rows

The ratings dataframe is  98.30% empty.
+------+-----+
|userId|count|
+------+-----+
```

```
The ratings dataframe is  98.30% empty.
+------+-----+
|userId|count|
+------+-----+
|   414| 2698|
|   599| 2478|
|   474| 2108|
|   448| 1864|
|   274| 1346|
|   610| 1302|
|    68| 1260|
|   380| 1218|
|   606| 1115|
|   288| 1055|
|   249| 1046|
|   387| 1027|
|   182|  977|
|   307|  975|
|   603|  943|
|   298|  939|
|   177|  904|
|   318|  879|
|   232|  862|
|   480|  836|
+------+-----+
only showing top 20 rows

+-------+-----+
|movieId|count|
+-------+-----+
|    356|  329|
```

```
only showing top 20 rows

+-------+-----+
|movieId|count|
+-------+-----+
|    356|  329|
|    318|  317|
|    296|  307|
|    593|  279|
|   2571|  278|
|    260|  251|
|    480|  238|
|    110|  237|
|    589|  224|
|    527|  220|
|   2959|  218|
|      1|  215|
|   1196|  211|
|   2858|  204|
|     50|  204|
|     47|  203|
|    780|  202|
|    150|  201|
|   1198|  200|
|   4993|  198|
+-------+-----+
only showing top 20 rows

Num models to be tested:  16
CrossValidator_96912e7e175b
**Best Model**
  Rank: 50
  MaxIter: 10
```

```
only showing top 20 rows

Num models to be tested:  16
CrossValidator_96912e7e175b
**Best Model**
  Rank: 50
  MaxIter: 10
  RegParam: 0.15
0.8685666272031658
+------+-------+------+----------+
|userId|movieId|rating|prediction|
+------+-------+------+----------+
|   580|   1580|   4.0| 3.4476712|
|   580|  44022|   3.5| 3.2499712|
|   597|    471|   2.0| 4.2078404|
|   108|   1959|   5.0| 3.9294207|
|   368|   2122|   2.0| 1.8601142|
|   436|    471|   3.0| 3.6853335|
|   587|   1580|   4.0| 3.7985733|
|    27|   1580|   3.0| 3.4053385|
|   606|   1580|   2.5| 3.1694307|
|   606|  44022|   4.0| 2.8594952|
|    91|   2122|   4.0| 2.4488945|
|   597|   2387|   4.0|  3.818116|
|   368|    540|   2.0|   2.11591|
|   368|   1127|   4.0|  2.900031|
|    28|  48780|   1.5| 3.2513812|
|   497|    858|   4.0| 3.5255935|
|    76|    858|   5.0| 3.6248567|
|   332|  48780|   3.5| 3.8020349|
|   577|    858|   5.0| 3.9408183|
|   606|    897|   3.5|  3.474449|
+------+-------+------+----------+
```

```
only showing top 20 rows

+------+--------------------+
|userId|     recommendations|
+------+--------------------+
|    80|[{3379, 5.54679},...|
|   240|[{67618, 5.295908...|
|   160|[{6591, 4.592959}...|
|    70|[{3379, 5.514155}...|
|   480|[{3379, 4.615519}...|
|   390|[{3379, 4.8776674...|
|   550|[{3379, 5.2294383...|
|   490|[{3379, 4.474837}...|
|    60|[{3379, 4.678643}...|
|    90|[{3379, 5.1346874...|
+------+--------------------+

+------+-------+---------+
|userId|movieId|   rating|
+------+-------+---------+
|   350|  33649|4.3721337|
|   350|   3379| 4.272633|
|   350|  74226|4.2142377|
|   350|  84273|4.2142377|
|   350| 138966|4.2142377|
|   350|  26073|4.2142377|
|   350| 184245|4.2142377|
|   350| 179135|4.2142377|
|   350|   7071|4.2142377|
|   350| 117531|4.2142377|
+------+-------+---------+

+-------+------+---------+--------------------+--------------------+
|movieId|userId|   rating|               title|              genres|
```

```
+-------+------+---------+--------------------+--------------------+
|movieId|userId|   rating|               title|              genres|
+-------+------+---------+--------------------+--------------------+
|  67618|   100|5.1201425|Strictly Sexual (...|Comedy|Drama|Romance|
|   3379|   100| 5.064743| On the Beach (1959)|               Drama|
|  42730|   100| 5.042285|   Glory Road (2006)|               Drama|
|  33649|   100| 5.021657|  Saving Face (2004)|Comedy|Drama|Romance|
| 117531|   100|4.9267745|    Watermark (2014)|         Documentary|
|   7071|   100|4.9267745|Woman Under the I...|               Drama|
| 184245|   100|4.9267745|De platte jungle ...|         Documentary|
|  26073|   100|4.9267745|Human Condition I...|           Drama|War|
| 179135|   100|4.9267745|Blue Planet II (2...|         Documentary|
|  84273|   100|4.9267745|Zeitgeist: Moving...|         Documentary|
+-------+------+---------+--------------------+--------------------+

+-------+------+------+--------------------+--------------------+
|movieId|userId|rating|               title|              genres|
+-------+------+------+--------------------+--------------------+
|   1101|   100|   5.0|      Top Gun (1986)|      Action|Romance|
|   1958|   100|   5.0|Terms of Endearme...|        Comedy|Drama|
|   2423|   100|   5.0|Christmas Vacatio...|              Comedy|
|   4041|   100|   5.0|Officer and a Gen...|       Drama|Romance|
|   5620|   100|   5.0|Sweet Home Alabam...|      Comedy|Romance|
|    368|   100|   4.5|     Maverick (1994)|Adventure|Comedy|...|
|    934|   100|   4.5|Father of the Bri...|              Comedy|
|    539|   100|   4.5|Sleepless in Seat...|Comedy|Drama|Romance|
|     16|   100|   4.5|       Casino (1995)|         Crime|Drama|
|    553|   100|   4.5|    Tombstone (1993)|Action|Drama|Western|
+-------+------+------+--------------------+--------------------+

22/12/01 02:24:00 INFO org.sparkproject.jetty.server.AbstractConnector: Stopped Spark@60cdfcd8{HTTP/1.1, (http/
1.1)}{0.0.0.0:0}
```

```
rpan@cluster-recom-1-m:~$ ^C
rpan@cluster-recom-1-m:~$ ^C
rpan@cluster-recom-1-m:~$ ls
movies.csv  ratings.csv  recommendation_engine_movielens.py  recommendation_engine_movielens_test.py
rpan@cluster-recom-1-m:~$ vi ^C
rpan@cluster-recom-1-m:~$ vi recommendation_engine_movielens_test.py
rpan@cluster-recom-1-m:~$
```

https://ssh.cloud.google.com/v2/ssh/projects/my-mapreduce-project-367604/zones/us-west1-a/instances/cluster-recom-1-m?authuser=...

ssh.cloud.google.com/v2/ssh/projects/my-mapreduce-project-367604/zones/us-west1-a/instances/cluster-recom-1-m?authuser=0&hl=...

SSH-in-browser UPLOAD FILE DOWNLOAD FILE

```
# -*- coding: utf-8 -*-
"""CS522_Week9_HW2_Yixin_Cao_19536.ipynb
Automatically generated by Colaboratory.
Original file is located at
    https://colab.research.google.com/drive/1zN8WElepvxHXyWvIN5qB6rJd8RsRJnVP
### **Step 1: Go to the correct directory**
"""

# Commented out IPython magic to ensure Python compatibility.
# %cd drive/MyDrive/Colab Notebooks/CS570
#!pwd

"""### **Step 2: Convert the .txt data file to .csv file**"""

import pandas as pd
# covert txt to cvs
#read_file = pd.read_csv ("movies.txt")
#read_file.to_csv ("hdfs:///data/movies.csv", index=None)

#read_file = pd.read_csv ("ratings.txt")
#read_file.to_csv ("hdfs:///data/ratings.csv", index=None)

#read_file = pd.read_csv ("tags.txt")
#read_file.to_csv ("hdfs:///tags.csv", index=None)

"""### **Step 3: Install pyspark**"""

#!pip install pyspark

"""### **Step 4: Import libraries**"""

import pandas as pd
from pyspark.sql.functions import col, explode
from pyspark import SparkContext

"""### **Step 5: Initiate spark session**"""

from pyspark.sql import SparkSession
                                                                    1,23          Top
```

```python
from pyspark import SparkContext

"""### **Step 5: Initiate spark session**"""

from pyspark.sql import SparkSession
sc = SparkContext
# sc.setCheckpointDir('checkpoint')
spark = SparkSession.builder.appName('Recommendations').getOrCreate()

"""### **Step 6: Load data**"""

# Commented out IPython magic to ensure Python compatibility.
# %cd drive/MyDrive/Colab Notebooks/CS570
movies = spark.read.csv("hdfs:///mydata/movies.csv",header=True)
ratings = spark.read.csv("hdfs:///mydata/ratings.csv",header=True)

ratings.show()

ratings.printSchema()

ratings = ratings.\
    withColumn('userId', col('userId').cast('integer')).\
    withColumn('movieId', col('movieId').cast('integer')).\
    withColumn('rating', col('rating').cast('float')).\
    drop('timestamp')
ratings.show()

"""### **Step 7: Calculate sparsity**"""

# Count the total number of ratings in the dataset
numerator = ratings.select("rating").count()

# Count the number of distinct userIds and distinct movieIds
num_users = ratings.select("userId").distinct().count()
num_movies = ratings.select("movieId").distinct().count()

# Set the denominator equal to the number of users multiplied by the number of movies
denominator = num_users * num_movies
                                                                                    66,36          24%
```

https://ssh.cloud.google.com/v2/ssh/projects/my-mapreduce-project-367604/zones/us-west1-a/instances/cluster-recom-1-m?authuser=...

ssh.cloud.google.com/v2/ssh/projects/my-mapreduce-project-367604/zones/us-west1-a/instances/cluster-recom-1-m?authuser=0&hl=...

SSH-in-browser UPLOAD FILE DOWNLOAD FILE

```
num_movies = ratings.select("movieId").distinct().count()

# Set the denominator equal to the number of users multiplied by the number of movies
denominator = num_users * num_movies

# Divide the numerator by the denominator
sparsity = (1.0 - (numerator *1.0)/denominator)*100
print("The ratings dataframe is ", "%.2f" % sparsity + "% empty.")

"""### **Step 8: Interpret ratings**"""

# Group data by userId, count ratings
userId_ratings = ratings.groupBy("userId").count().orderBy('count', ascending=False)
userId_ratings.show()

# Group data by userId, count ratings
movieId_ratings = ratings.groupBy("movieId").count().orderBy('count', ascending=False)
movieId_ratings.show()

"""### **Step 9: Build Out An ALS Model**"""

# Import the required functions
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.recommendation import ALS
from pyspark.ml.tuning import ParamGridBuilder, CrossValidator

# Create test and train set
(train, test) = ratings.randomSplit([0.8, 0.2], seed = 1234)

# Create ALS model
als = ALS(userCol="userId", itemCol="movieId", ratingCol="rating", nonnegative =
          True, implicitPrefs = False, coldStartStrategy="drop")

# Confirm that a model called "als" was created
type(als)

"""### **Step 10: Tell Spark how to tune your ALS model**"""

                                                                    100,0-1       48%
```

```python
type(als)

"""### **Step 10: Tell Spark how to tune your ALS model**"""

# Import the requisite items
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.tuning import ParamGridBuilder, CrossValidator

# Add hyperparameters and their respective values to param_grid
param_grid = ParamGridBuilder() \
            .addGrid(als.rank, [10, 50, 100, 150]) \
            .addGrid(als.regParam, [.01, .05, .1, .15]) \
            .build()
            #             .addGrid(als.maxIter, [5, 50, 100, 200]) \


# Define evaluator as RMSE and print length of evaluator
evaluator = RegressionEvaluator(metricName="rmse", labelCol="rating", predictionCol="prediction")
print ("Num models to be tested: ", len(param_grid))

"""### **Step 11: Build your cross validation pipeline**"""

# Build cross validation using CrossValidator
cv = CrossValidator(estimator=als, estimatorParamMaps=param_grid, evaluator=evaluator, numFolds=5)

# Confirm cv was built
print(cv)

"""### **Step 12: Best Model and Best Model Parameters**"""

#Fit cross validator to the 'train' dataset
model = cv.fit(train)
#Extract best model from the cv model above
best_model = model.bestModel

# # Print best_model
# print(type(best_model))

                                                                                    134,36        73%
```

https://ssh.cloud.google.com/v2/ssh/projects/my-mapreduce-project-367604/zones/us-west1-a/instances/cluster-recom-1-m?authuser=...

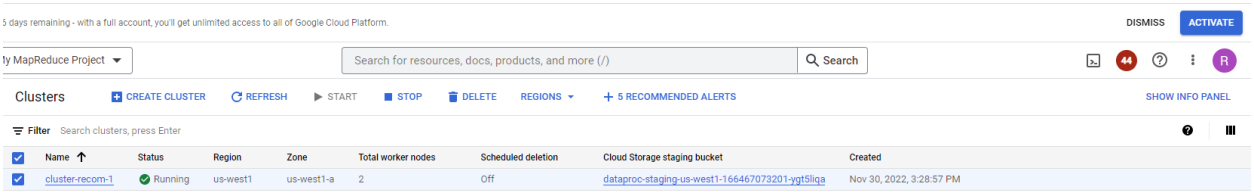
ssh.cloud.google.com/v2/ssh/projects/my-mapreduce-project-367604/zones/us-west1-a/instances/cluster-recom-1-m?authuser=0&hl=...

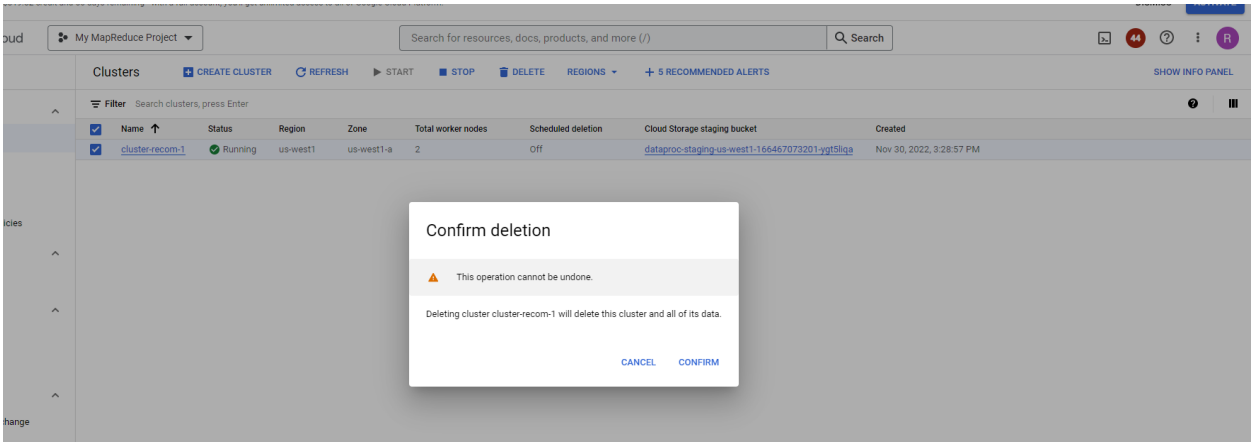
SSH-in-browser UPLOAD FILE DOWNLOAD FILE

```
print(cv)

"""### **Step 12: Best Model and Best Model Parameters**"""

#Fit cross validator to the 'train' dataset
model = cv.fit(train)
#Extract best model from the cv model above
best_model = model.bestModel

# # Print best_model
# print(type(best_model))

# Complete the code below to extract the ALS model parameters
print("**Best Model**")

# # Print "Rank"
print("  Rank:", best_model._java_obj.parent().getRank())

# Print "MaxIter"
print("  MaxIter:", best_model._java_obj.parent().getMaxIter())

# Print "RegParam"
print("  RegParam:", best_model._java_obj.parent().getRegParam())

# View the predictions
test_predictions = best_model.transform(test)
RMSE = evaluator.evaluate(test_predictions)
print(RMSE)

test_predictions.show()

"""### **Step 13: Make Recommendations**"""

# Generate n Recommendations for all users
nrecommendations = best_model.recommendForAllUsers(10)
nrecommendations.limit(10).show()

nrecommendations = nrecommendations\
                                                                                    160,0-1       92%
```

```
"""### **Step 13: Make Recommendations**"""

# Generate n Recommendations for all users
nrecommendations = best_model.recommendForAllUsers(10)
nrecommendations.limit(10).show()

nrecommendations = nrecommendations\
    .withColumn("rec_exp", explode("recommendations"))\
    .select('userId', col("rec_exp.movieId"), col("rec_exp.rating"))

nrecommendations.limit(10).show()

"""## **Do the recommendations make sense?**"""

nrecommendations.join(movies, on='movieId').filter('userId = 100').show()

ratings.join(movies, on='movieId').filter('userId = 100').sort('rating', ascending=False).limit(10).show()
                                                                                    175,36        Bot
```

Part 5: Delete the cluster from GCP


5 days remaining - with a full account, you'll get unlimited access to all of Google Cloud Platform.
DISMISS
ACTIVATE
My MapReduce Project
Search for resources, docs, products, and more (/)
Search
Clusters
CREATE CLUSTER
REFRESH
START
STOP
DELETE
REGIONS
5 RECOMMENDED ALERTS
SHOW INFO PANEL
Filter
Search clusters, press Enter
Name
Status
Region
Zone
Total worker nodes
Scheduled deletion
Cloud Storage staging bucket
Created
cluster-recom-1
Running
us-west1
us-west1-a
2
Off
dataproc-staging-us-west1-166467073201-ygt5liqa
Nov 30, 2022, 3:28:57 PM


My MapReduce Project
Search for resources, docs, products, and more (/)
Search
Clusters
CREATE CLUSTER
REFRESH
START
STOP
DELETE
REGIONS
5 RECOMMENDED ALERTS
SHOW INFO PANEL
Filter
Search clusters, press Enter
Name
Status
Region
Zone
Total worker nodes
Scheduled deletion
Cloud Storage staging bucket
Created
cluster-recom-1
Running
us-west1
us-west1-a
2
Off
dataproc-staging-us-west1-166467073201-ygt5liqa
Nov 30, 2022, 3:28:57 PM
Confirm deletion
This operation cannot be undone.
Deleting cluster cluster-recom-1 will delete this cluster and all of its data.
CANCEL
CONFIRM