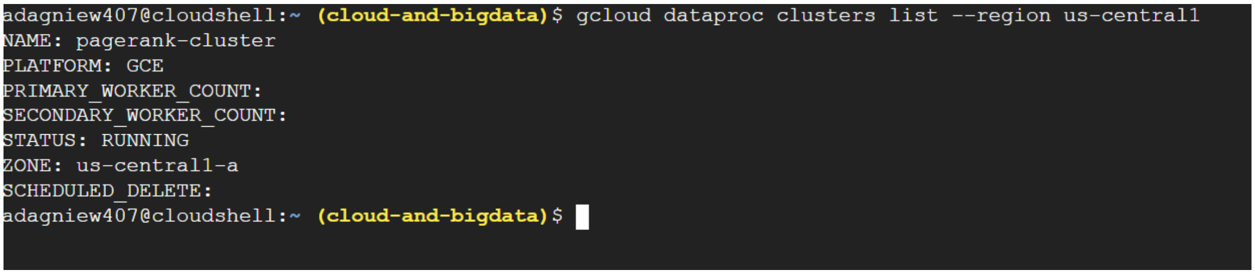
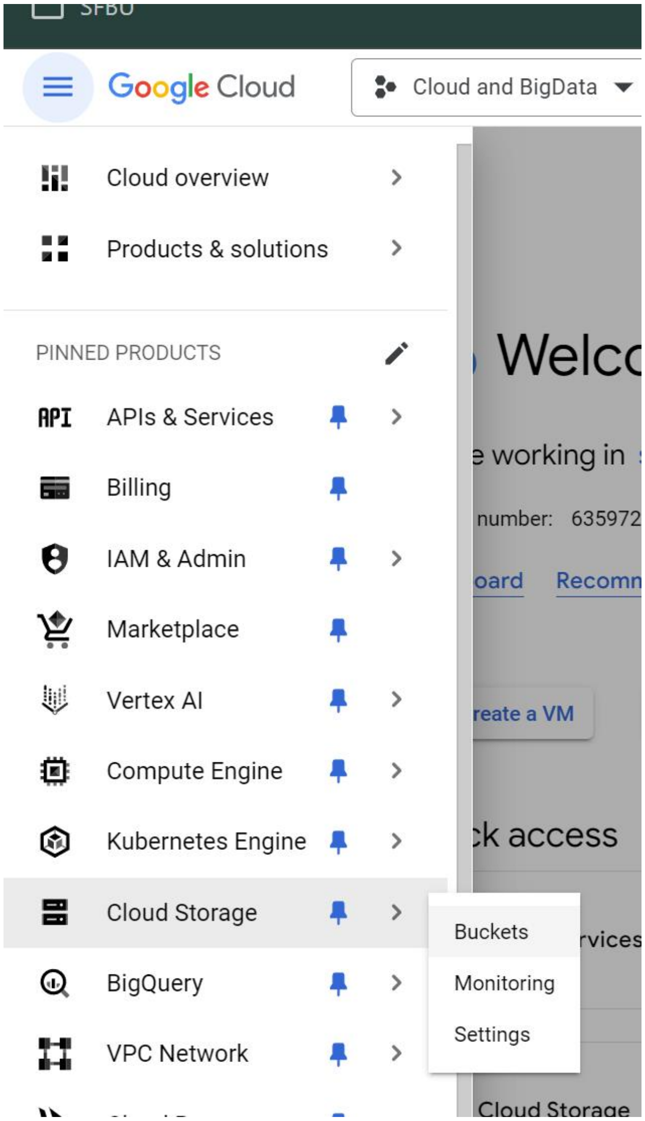
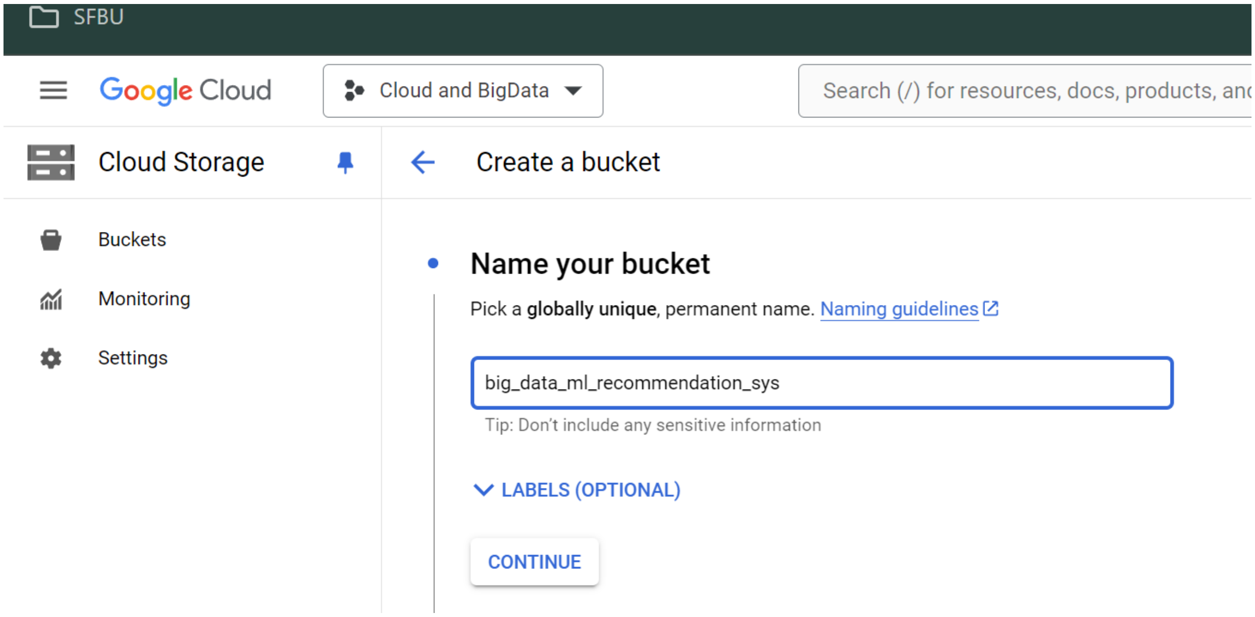
Step 1: gcloud dataproc clusters list --region us-central1



```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ gcloud dataproc clusters list --region us-central1
NAME: pagerank-cluster
PLATFORM: GCE
PRIMARY_WORKER_COUNT:
SECONDARY_WORKER_COUNT:
STATUS: RUNNING
ZONE: us-central1-a
SCHEDULED_DELETE:
adagniew407@cloudshell:~ (cloud-and-bigdata)$
```

Step 2: Create a bucket:



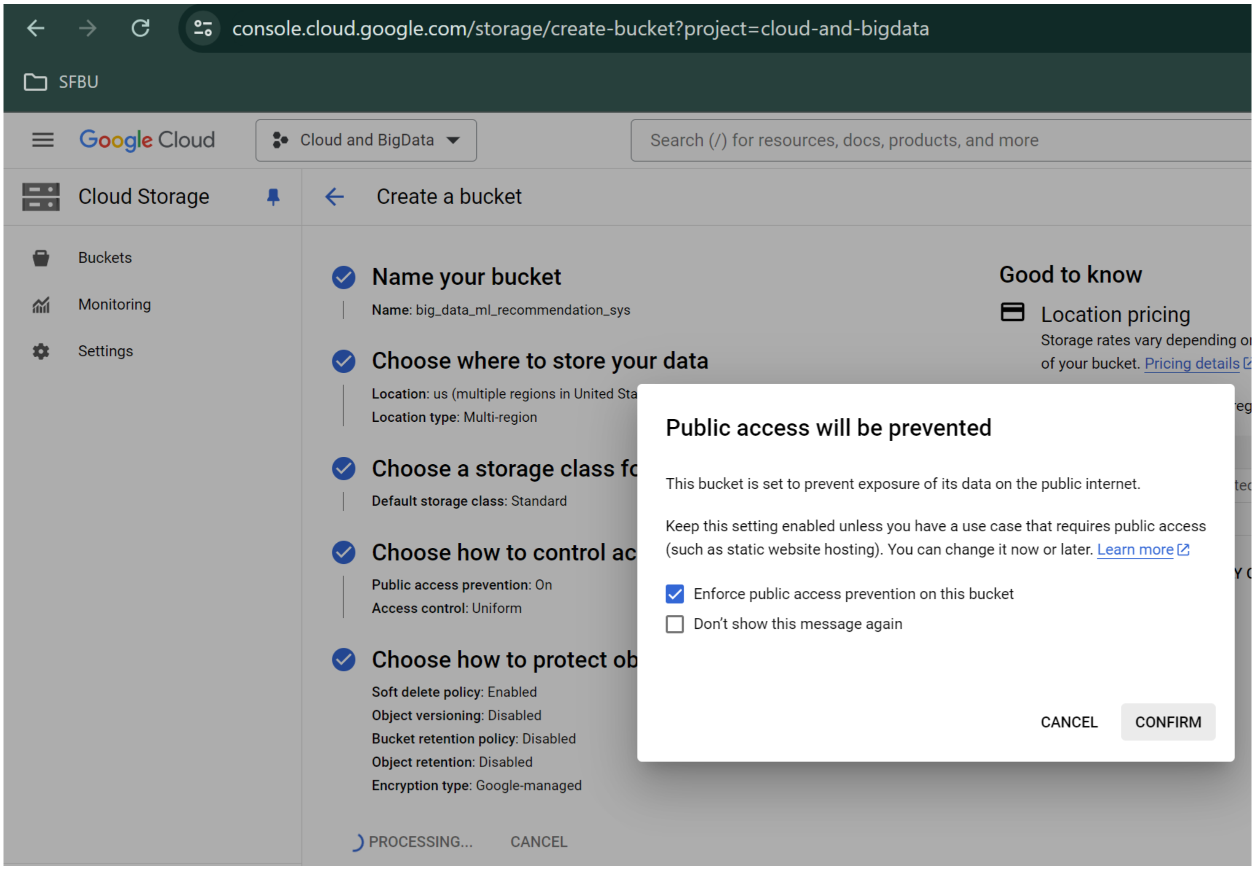


Name your bucket

Pick a **globally unique**, permanent name. Naming guidelines

big_data_ml_recommendation_sys

Tip: Don't include any sensitive information


console.cloud.google.com/storage/create-bucket?project=cloud-and-bigdata
SFBU
Google Cloud
Cloud and BigData
Search (/) for resources, docs, products, and more
Cloud Storage
Buckets
Monitoring
Settings
Create a bucket

Name your bucket

Name: big_data_ml_recommendation_sys

Choose where to store your data

Location: us (multiple regions in United Sta

Location type: Multi-region

Choose a storage class fo

Default storage class: Standard

Choose how to control ac

Public access prevention: On

Access control: Uniform

Choose how to protect ob

Soft delete policy: Enabled

Object versioning: Disabled

Bucket retention policy: Disabled

Object retention: Disabled

Encryption type: Google-managed

Good to know

Location pricing

Storage rates vary depending o of your bucket. Pricing details

Public access will be prevented

This bucket is set to prevent exposure of its data on the public internet.

Keep this setting enabled unless you have a use case that requires public access (such as static website hosting). You can change it now or later. Learn more

☑ Enforce public access prevention on this bucket

□ Don't show this message again

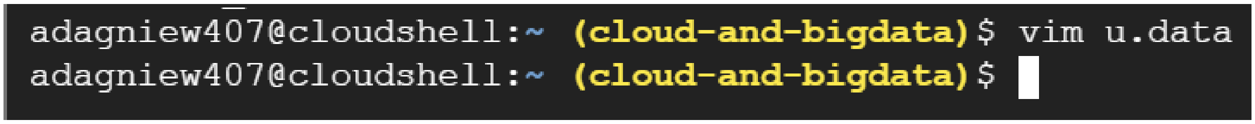
Step 1: Prepare and Transform Data

Description:

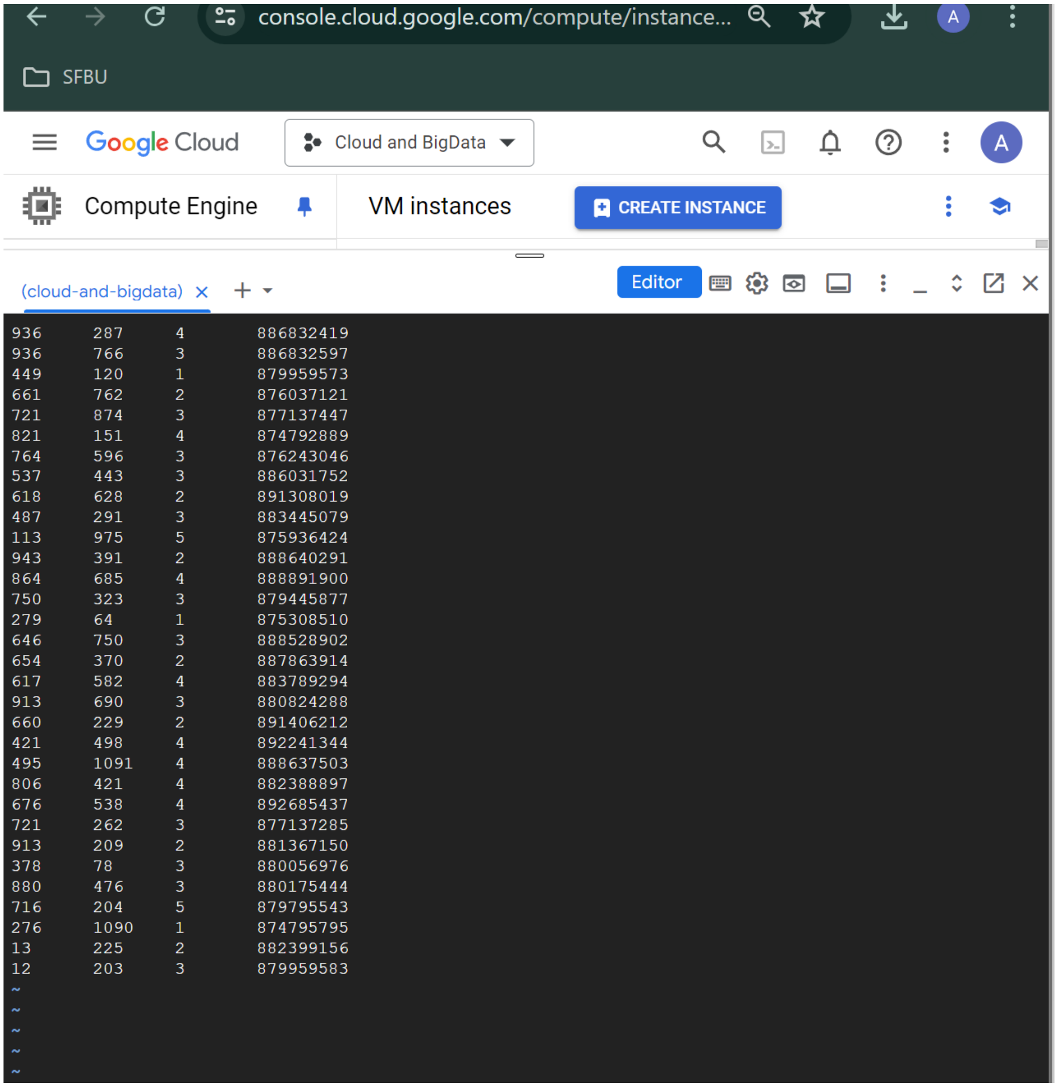
Transform the `u.data` file to the required format (UserID, MovieID, rating) using a shell script and upload it to your Cloud Storage bucket.

Code:

1. **Create the `u.data` File:** Create a file named `u.data` and populate it with your data.



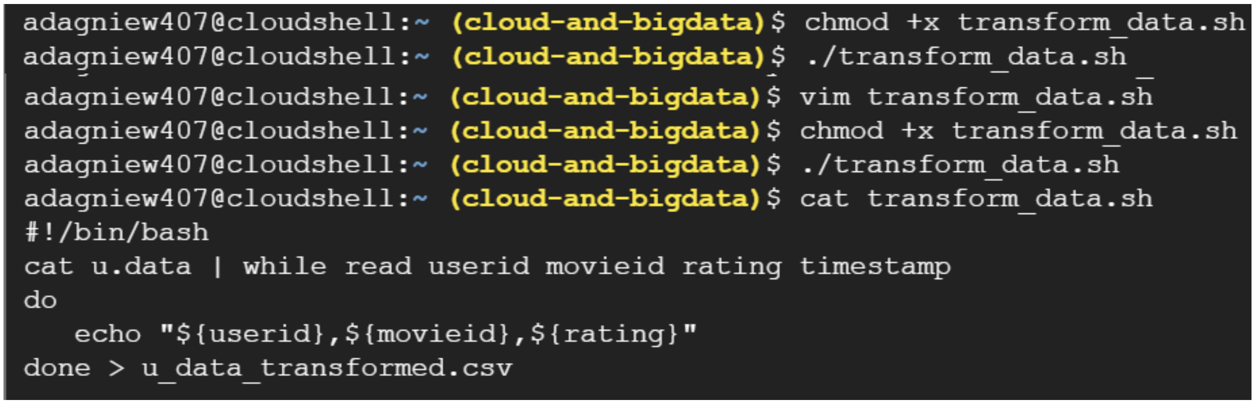
```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ vim u.data
adagniew407@cloudshell:~ (cloud-and-bigdata)$ 
```



```
936     287     4       886832419
936     766     3       886832597
449     120     1       879959573
661     762     2       876037121
721     874     3       877137447
821     151     4       874792889
764     596     3       876243046
537     443     3       886031752
618     628     2       891308019
487     291     3       883445079
113     975     5       875936424
943     391     2       888640291
864     685     4       888891900
750     323     3       879445877
279     64      1       875308510
646     750     3       888528902
654     370     2       887863914
617     582     4       883789294
913     690     3       880824288
660     229     2       891406212
421     498     4       892241344
495     1091    4       888637503
806     421     4       882388897
676     538     4       892685437
721     262     3       877137285
913     209     2       881367150
378     78      3       880056976
880     476     3       880175444
716     204     5       879795543
276     1090    1       874795795
13      225     2       882399156
12      203     3       879959583
~
~
~
~
~
```

2. **Transform Data Using Shell Script:**

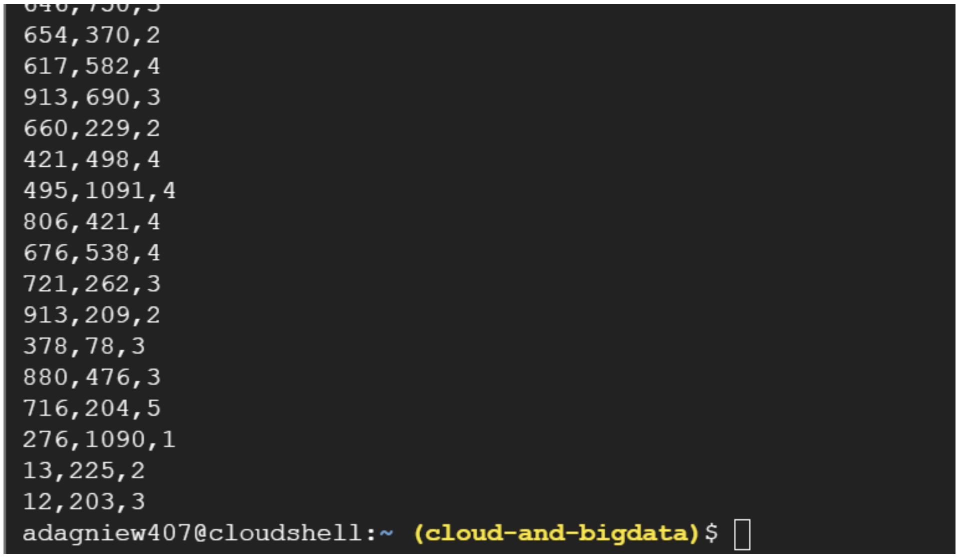
```
# Create transform_data.sh
echo '#!/bin/bash
cat u.data | tr -s ' ' | cut -d' ' -f1-3 | tr ' ' ',' >
u_data_transformed.csv' > transform_data.sh

# Make the script executable
chmod +x transform_data.sh

# Run the script
./transform_data.sh
```



```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ chmod +x transform_data.sh
adagniew407@cloudshell:~ (cloud-and-bigdata)$ ./transform_data.sh
adagniew407@cloudshell:~ (cloud-and-bigdata)$ vim transform_data.sh
adagniew407@cloudshell:~ (cloud-and-bigdata)$ chmod +x transform_data.sh
adagniew407@cloudshell:~ (cloud-and-bigdata)$ ./transform_data.sh
adagniew407@cloudshell:~ (cloud-and-bigdata)$ cat transform_data.sh
#!/bin/bash
cat u.data | while read userid movieid rating timestamp
do
   echo "${userid},${movieid},${rating}"
done > u_data_transformed.csv
```

Explanation:

The shell script reads the `u.data` file, trims extra spaces, extracts the first three fields (UserID, MovieID, rating), and replaces spaces with commas. The transformed data is saved in `u_data_transformed.csv`.



```
654,370,2
617,582,4
913,690,3
660,229,2
421,498,4
495,1091,4
806,421,4
676,538,4
721,262,3
913,209,2
378,78,3
880,476,3
716,204,5
276,1090,1
13,225,2
12,203,3
adagniew407@cloudshell:~ (cloud-and-bigdata)$ 
```

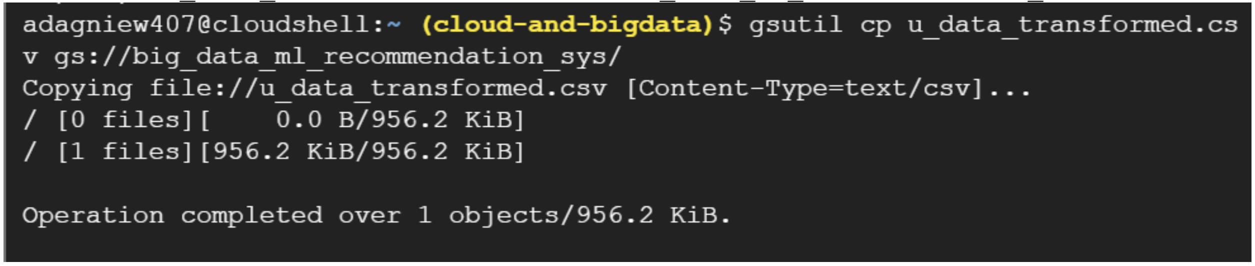
Step 2: Upload Data to Cloud Storage Bucket

Description:

Upload the transformed data file `u_data_transformed.csv` to your Cloud Storage bucket.

Code:

```
# Upload the transformed data to Cloud Storage
gsutil cp u_data_transformed.csv gs://big_data_ml_recommendation_sys/
```



```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ gsutil cp u_data_transformed.cs
v gs://big_data_ml_recommendation_sys/
Copying file://u_data_transformed.csv [Content-Type=text/csv]...
/ [0 files][    0.0 B/956.2 KiB]
/ [1 files][956.2 KiB/956.2 KiB]

Operation completed over 1 objects/956.2 KiB.
```

Explanation:

The `gsutil cp` command copies the `u_data_transformed.csv` file from your local machine to your specified Cloud Storage bucket.

Step 3: Create and Upload the PySpark Script

Description:

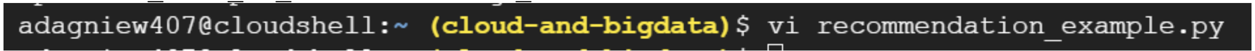
Create a PySpark script to perform collaborative filtering using MLlib and upload it to your Cloud Storage bucket.

Code:

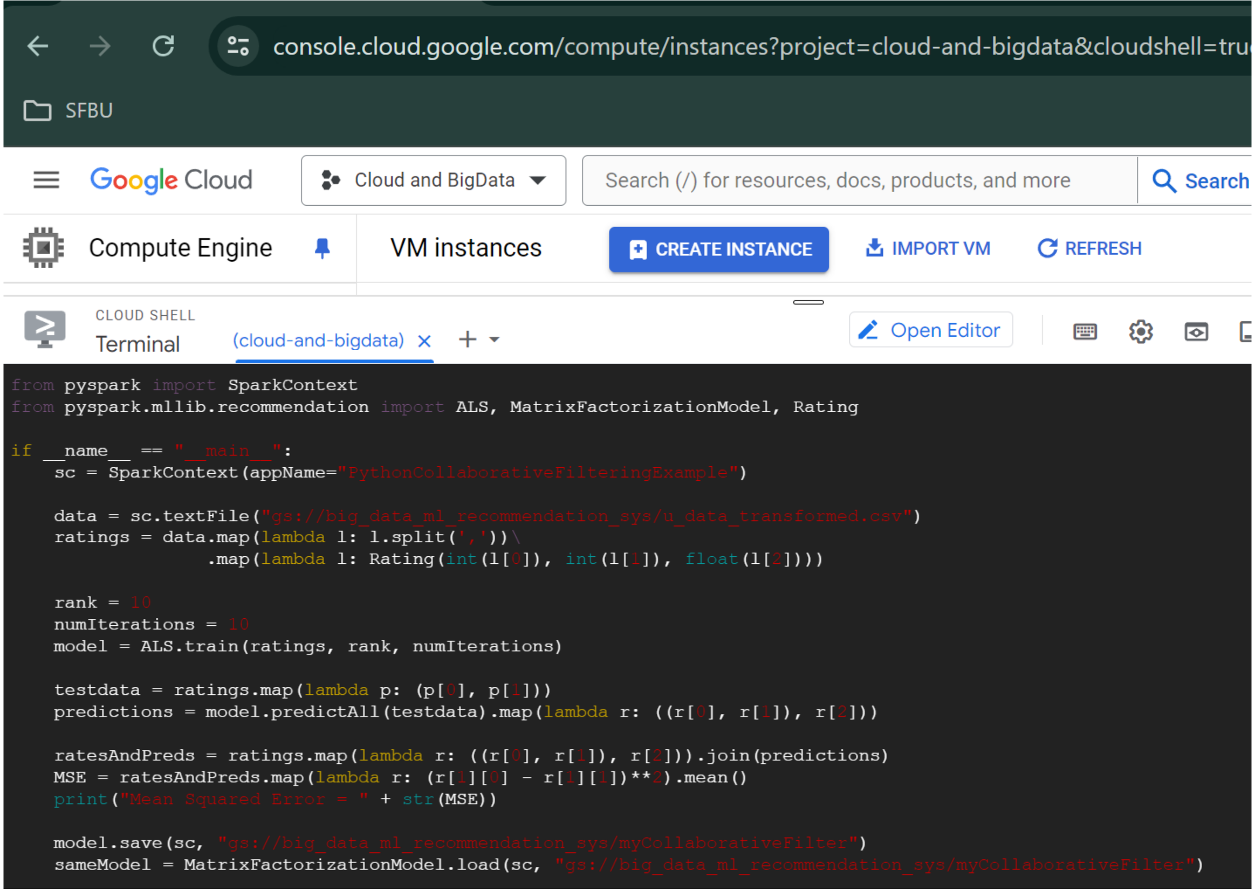
1. **Create the PySpark Script:** Create a file named `recommendation_example.py` with the following content:

```
from pyspark import SparkContext
from pyspark.mllib.recommendation import ALS, MatrixFactorizationModel,
Rating

if __name__ == "__main__":
    sc = SparkContext(appName="PythonCollaborativeFilteringExample")

    data =
sc.textFile("gs://big_data_ml_recommendation_sys/u_data_transformed.csv
")
    ratings = data.map(lambda l: l.split(','))\
                  .map(lambda l: Rating(int(l[0]), int(l[1]),
float(l[2])))
```

```
    rank = 10
    numIterations = 10
    model = ALS.train(ratings, rank, numIterations)

    testdata = ratings.map(lambda p: (p[0], p[1]))
    predictions = model.predictAll(testdata).map(lambda r: ((r[0],
r[1]), r[2]))

    ratesAndPreds = ratings.map(lambda r: ((r[0], r[1]),
r[2])).join(predictions)
    MSE = ratesAndPreds.map(lambda r: (r[1][0] - r[1][1])**2).mean()
    print("Mean Squared Error = " + str(MSE))

    model.save(sc,
"gs://big_data_ml_recommendation_sys/myCollaborativeFilter")
    sameModel = MatrixFactorizationModel.load(sc,
"gs://big_data_ml_recommendation_sys/myCollaborativeFilter")
```



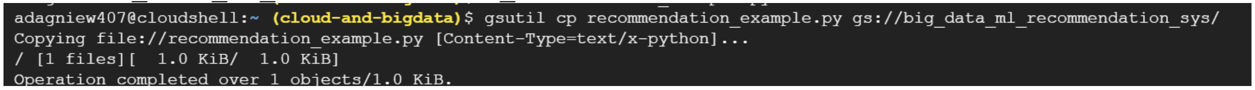
```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ vi recommendation_example.py
```



```
from pyspark import SparkContext
from pyspark.mllib.recommendation import ALS, MatrixFactorizationModel, Rating

if __name__ == "__main__":
    sc = SparkContext(appName="PythonCollaborativeFilteringExample")

    data = sc.textFile("gs://big_data_ml_recommendation_sys/u_data_transformed.csv")
    ratings = data.map(lambda l: l.split(','))\
                  .map(lambda l: Rating(int(l[0]), int(l[1]), float(l[2])))

    rank = 10
    numIterations = 10
    model = ALS.train(ratings, rank, numIterations)

    testdata = ratings.map(lambda p: (p[0], p[1]))
    predictions = model.predictAll(testdata).map(lambda r: ((r[0], r[1]), r[2]))

    ratesAndPreds = ratings.map(lambda r: ((r[0], r[1]), r[2])).join(predictions)
    MSE = ratesAndPreds.map(lambda r: (r[1][0] - r[1][1])**2).mean()
    print("Mean Squared Error = " + str(MSE))

    model.save(sc, "gs://big_data_ml_recommendation_sys/myCollaborativeFilter")
    sameModel = MatrixFactorizationModel.load(sc, "gs://big_data_ml_recommendation_sys/myCollaborativeFilter")
```

2. **Upload the PySpark Script:**

   ```
   gsutil cp recommendation_example.py
   gs://big_data_ml_recommendation_sys/
   ```



```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ gsutil cp recommendation_example.py gs://big_data_ml_recommendation_sys/
Copying file://recommendation_example.py [Content-Type=text/x-python]...
/ [1 files][  1.0 KiB/  1.0 KiB]
Operation completed over 1 objects/1.0 KiB.
```

Explanation:

The PySpark script loads the transformed data from Cloud Storage, trains a collaborative filtering model using ALS, evaluates the model by calculating the mean squared error, and saves the model back to Cloud Storage. The script is then uploaded to the Cloud Storage bucket.

Step 4: Submit the PySpark Job to Dataproc

Description:

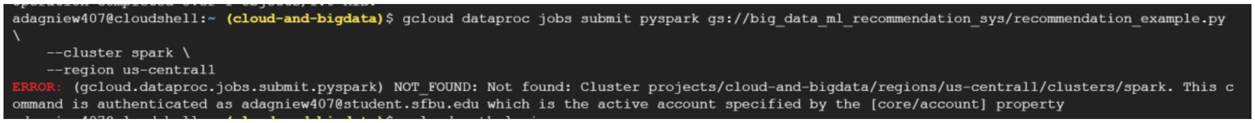
Submit the PySpark job to your Dataproc cluster to execute the collaborative filtering task.

Code:

```
gcloud dataproc jobs submit pyspark
gs://big_data_ml_recommendation_sys/recommendation_example.py \
    --cluster spark \
    --region us-central1
```

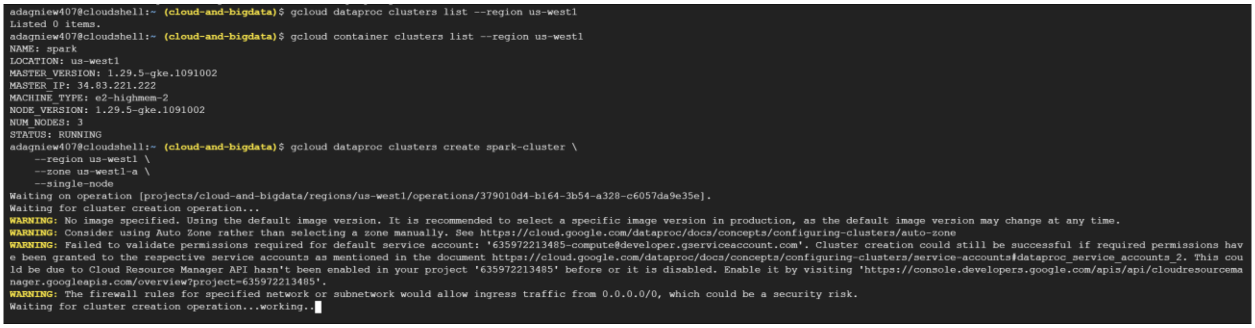
Explanation:

The `gcloud dataproc jobs submit pyspark` command submits the PySpark script stored in Cloud Storage to the Dataproc cluster named `spark` located in the `us-central1` region for execution.



```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ gcloud dataproc jobs submit pyspark gs://big_data_ml_recommendation_sys/recommendation_example.py
\
    --cluster spark \
    --region us-central1
ERROR: (gcloud.dataproc.jobs.submit.pyspark) NOT_FOUND: Not found: Cluster projects/cloud-and-bigdata/regions/us-central1/clusters/spark. This c
ommand is authenticated as adagniew407@student.sfbu.edu which is the active account specified by the [core/account] property
```

I faced an Issue here: So I need to authenticate using the below command.



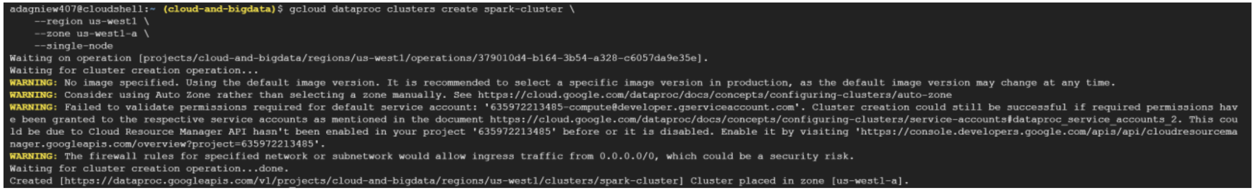
```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ gcloud dataproc clusters list --region us-west1
Listed 0 items.
adagniew407@cloudshell:~ (cloud-and-bigdata)$ gcloud container clusters list --region us-west1
NAME: spark
LOCATION: us-west1
MASTER_VERSION: 1.29.5-gke.1091002
MASTER_IP: 34.83.221.222
MACHINE_TYPE: e2-highmem-2
NODE_VERSION: 1.29.5-gke.1091002
NUM_NODES: 3
STATUS: RUNNING
adagniew407@cloudshell:~ (cloud-and-bigdata)$ gcloud dataproc clusters create spark-cluster \
    --region us-west1 \
    --zone us-west1-a \
    --single-node
Waiting on operation [projects/cloud-and-bigdata/regions/us-west1/operations/379010d4-b164-3b54-a328-c6057da9e35e].
Waiting for cluster creation operation...
WARNING: No image specified. Using the default image version. It is recommended to select a specific image version in production, as the default image version may change at any time.
WARNING: Consider using Auto Zone rather than selecting a zone manually. See https://cloud.google.com/dataproc/docs/concepts/configuring-clusters/auto-zone
WARNING: Failed to validate permissions required for default service account: '635972213485-compute@developer.gserviceaccount.com'. Cluster creation could still be successful if required permissions hav
e been granted to the respective service accounts as mentioned in the document https://cloud.google.com/dataproc/docs/concepts/configuring-clusters/service-accounts#dataproc_service_accounts_2. This cou
ld be due to Cloud Resource Manager API hasn't been enabled in your project '635972213485' before or it is disabled. Enable it by visiting 'https://console.developers.google.com/apis/api/cloudresourcema
nager.googleapis.com/overview?project=635972213485'.
WARNING: The firewall rules for specified network or subnetwork would allow ingress traffic from 0.0.0.0/0, which could be a security risk.
Waiting for cluster creation operation...working..
```

it seems like I have a Google Kubernetes Engine (GKE) cluster named `spark` in the `us-west1` region, but no Google Cloud Dataproc clusters.

To run a PySpark job on a Dataproc cluster, you need to first create a Dataproc cluster. Here's how to do that:

1. **Create a Dataproc Cluster:**

   ```
   gcloud dataproc clusters create spark-cluster \
       --region us-west1 \
       --zone us-west1-a \
       --single-node
   ```



```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ gcloud dataproc clusters create spark-cluster \
    --region us-west1 \
    --zone us-west1-a \
    --single-node
Waiting on operation [projects/cloud-and-bigdata/regions/us-west1/operations/379010d4-b164-3b54-a328-c6057da9e35e].
Waiting for cluster creation operation...
WARNING: No image specified. Using the default image version. It is recommended to select a specific image version in production, as the default image version may change at any time.
WARNING: Consider using Auto Zone rather than selecting a zone manually. See https://cloud.google.com/dataproc/docs/concepts/configuring-clusters/auto-zone
WARNING: Failed to validate permissions required for default service account: '635972213485-compute@developer.gserviceaccount.com'. Cluster creation could still be successful if required permissions hav
e been granted to the respective service accounts as mentioned in the document https://cloud.google.com/dataproc/docs/concepts/configuring-clusters/service-accounts#dataproc_service_accounts_2. This cou
ld be due to Cloud Resource Manager API hasn't been enabled in your project '635972213485' before or it is disabled. Enable it by visiting 'https://console.developers.google.com/apis/api/cloudresourcema
nager.googleapis.com/overview?project=635972213485'.
WARNING: The firewall rules for specified network or subnetwork would allow ingress traffic from 0.0.0.0/0, which could be a security risk.
Waiting for cluster creation operation...done.
Created [https://dataproc.googleapis.com/v1/projects/cloud-and-bigdata/regions/us-west1/clusters/spark-cluster] Cluster placed in zone [us-west1-a].
```

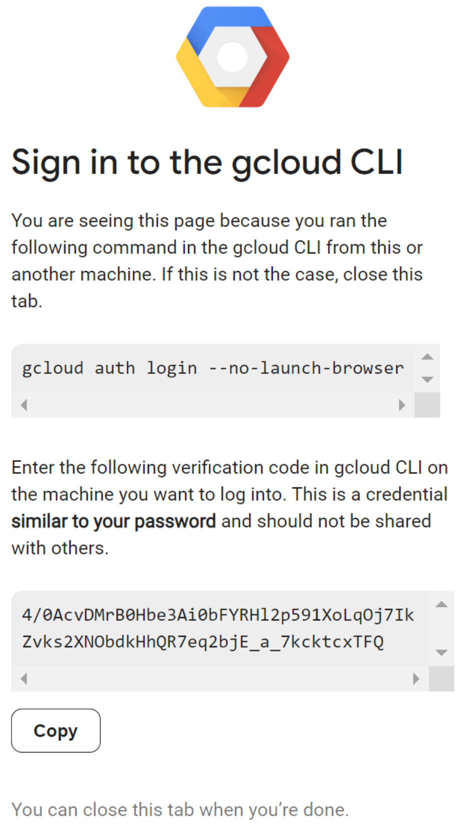
2. **Submit the PySpark Job:**

   After the cluster is created, I can submit your PySpark job:

   ```
   gcloud dataproc jobs submit pyspark
   gs://big_data_ml_recommendation_sys/recommendation_example.py \
       --cluster spark-cluster \
       --region us-west1
   ```

Make sure to replace `spark-cluster` with the actual name of your Dataproc cluster if you choose a different name. The cluster creation step might take a few minutes. Once it's running, you can then submit your job.

4o



Sign in to the gcloud CLI

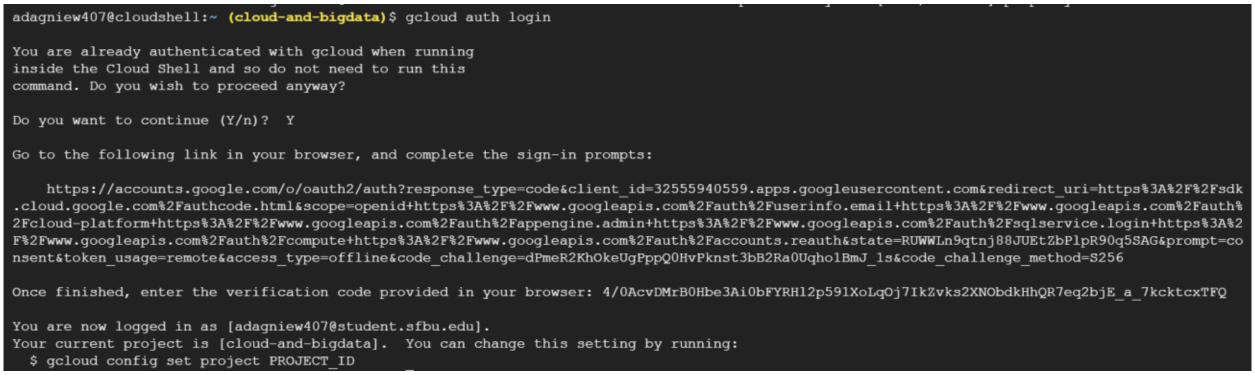
You are seeing this page because you ran the following command in the gcloud CLI from this or another machine. If this is not the case, close this tab.

```
gcloud auth login --no-launch-browser
```

Enter the following verification code in gcloud CLI on the machine you want to log into. This is a credential **similar to your password** and should not be shared with others.

```
4/0AcvDMrB0Hbe3Ai0bFYRHl2p591XoLqOj7Ik
Zvks2XNObdkHhQR7eq2bjE_a_7kcktcxTFQ
```

You can close this tab when you're done.



```
adagniew407@cloudshell:~ (cloud-and-bigdata)$ gcloud auth login

You are already authenticated with gcloud when running
inside the Cloud Shell and so do not need to run this
command. Do you wish to proceed anyway?

Do you want to continue (Y/n)?  Y

Go to the following link in your browser, and complete the sign-in prompts:

    https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleusercontent.com&redirect_uri=https%3A%2F%2Fsdk
.cloud.google.com%2Fauthcode.html&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%
2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fsqlservice.login+https%3A%2
F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&state=RUWWLn9qtnj88JUEtZbPlpR90q5SAG&prompt=co
nsent&token_usage=remote&access_type=offline&code_challenge=dPmeR2KhOkeUgPppQ0HvPknst3bB2Ra0Uqho1BmJ_1s&code_challenge_method=S256

Once finished, enter the verification code provided in your browser: 4/0AcvDMrB0Hbe3Ai0bFYRHl2p591XoLqOj7IkZvks2XNObdkHhQR7eq2bjE_a_7kcktcxTFQ

You are now logged in as [adagniew407@student.sfbu.edu].
Your current project is [cloud-and-bigdata].  You can change this setting by running:
  $ gcloud config set project PROJECT_ID
```

Step 5: Check Job Status and Output

Description:

Monitor the status of your submitted job and check its output to verify the results.

Code:

1. **View Job Status:**

   ```
   sh
   Copy code
   ```

```
gcloud dataproc jobs list --region us-central1
```

2. **Describe Job to View Details:**

   ```
   sh
   Copy code
   gcloud dataproc jobs describe job-id --region us-central1
   ```

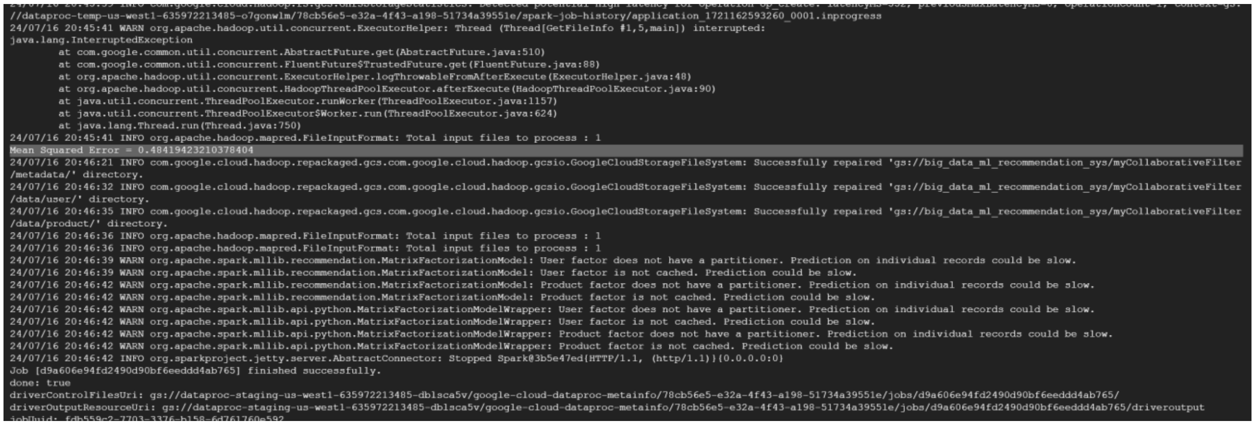
   Replace `job-id` with the actual job ID from the previous command's output.

Explanation:

The `gcloud dataproc jobs list` command lists all jobs and their statuses in the specified region. The `gcloud dataproc jobs describe` command provides detailed information about a specific job, including its output and any errors.

By following these steps, you will be able to successfully complete your assignment using your Dataproc cluster and Cloud Storage bucket on GCP.

Mean Squared Error = 0.48419423210378404



```
24/07/16 20:45:39 INFO com.google.cloud.hadoop.fs.gcs.GhfsStorageStatistics: Detected potential high latency for operation op_create. latencyMs=332; previousMaxLatencyMs=0; operationCount=1; context=gs:
//dataproc-temp-us-west1-635972213485-o7gonwlm/78cb56e5-e32a-4f43-a198-51734a39551e/spark-job-history/application_1721162593260_0001.inprogress
24/07/16 20:45:41 WARN org.apache.hadoop.util.concurrent.ExecutorHelper: Thread (Thread[GetFileInfo #1,5,main]) interrupted:
java.lang.InterruptedException
        at com.google.common.util.concurrent.AbstractFuture.get(AbstractFuture.java:510)
        at com.google.common.util.concurrent.FluentFuture$TrustedFuture.get(FluentFuture.java:88)
        at org.apache.hadoop.util.concurrent.ExecutorHelper.logThrowableFromAfterExecute(ExecutorHelper.java:48)
        at org.apache.hadoop.util.concurrent.HadoopThreadPoolExecutor.afterExecute(HadoopThreadPoolExecutor.java:90)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1157)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:750)
24/07/16 20:45:41 INFO org.apache.hadoop.mapred.FileInputFormat: Total input files to process : 1
Mean Squared Error = 0.48419423210378404
24/07/16 20:46:21 INFO com.google.cloud.hadoop.repackaged.gcs.com.google.cloud.hadoop.gcsio.GoogleCloudStorageFileSystem: Successfully repaired 'gs://big_data_ml_recommendation_sys/myCollaborativeFilter
/metadata/' directory.
24/07/16 20:46:32 INFO com.google.cloud.hadoop.repackaged.gcs.com.google.cloud.hadoop.gcsio.GoogleCloudStorageFileSystem: Successfully repaired 'gs://big_data_ml_recommendation_sys/myCollaborativeFilter
/data/user/' directory.
24/07/16 20:46:35 INFO com.google.cloud.hadoop.repackaged.gcs.com.google.cloud.hadoop.gcsio.GoogleCloudStorageFileSystem: Successfully repaired 'gs://big_data_ml_recommendation_sys/myCollaborativeFilter
/data/product/' directory.
24/07/16 20:46:36 INFO org.apache.hadoop.mapred.FileInputFormat: Total input files to process : 1
24/07/16 20:46:36 INFO org.apache.hadoop.mapred.FileInputFormat: Total input files to process : 1
24/07/16 20:46:39 WARN org.apache.spark.mllib.recommendation.MatrixFactorizationModel: User factor does not have a partitioner. Prediction on individual records could be slow.
24/07/16 20:46:39 WARN org.apache.spark.mllib.recommendation.MatrixFactorizationModel: User factor is not cached. Prediction could be slow.
24/07/16 20:46:42 WARN org.apache.spark.mllib.recommendation.MatrixFactorizationModel: Product factor does not have a partitioner. Prediction on individual records could be slow.
24/07/16 20:46:42 WARN org.apache.spark.mllib.recommendation.MatrixFactorizationModel: Product factor is not cached. Prediction could be slow.
24/07/16 20:46:42 WARN org.apache.spark.mllib.api.python.MatrixFactorizationModelWrapper: User factor does not have a partitioner. Prediction on individual records could be slow.
24/07/16 20:46:42 WARN org.apache.spark.mllib.api.python.MatrixFactorizationModelWrapper: User factor is not cached. Prediction could be slow.
24/07/16 20:46:42 WARN org.apache.spark.mllib.api.python.MatrixFactorizationModelWrapper: Product factor does not have a partitioner. Prediction on individual records could be slow.
24/07/16 20:46:42 WARN org.apache.spark.mllib.api.python.MatrixFactorizationModelWrapper: Product factor is not cached. Prediction could be slow.
24/07/16 20:46:42 INFO org.sparkproject.jetty.server.AbstractConnector: Stopped Spark@3b5e47ed{HTTP/1.1, (http/1.1)}{0.0.0.0:0}
Job [d9a606e94fd2490d90bf6eeddd4ab765] finished successfully.
done: true
driverControlFilesUri: gs://dataproc-staging-us-west1-635972213485-dblsca5v/google-cloud-dataproc-metainfo/78cb56e5-e32a-4f43-a198-51734a39551e/jobs/d9a606e94fd2490d90bf6eeddd4ab765/
driverOutputResourceUri: gs://dataproc-staging-us-west1-635972213485-dblsca5v/google-cloud-dataproc-metainfo/78cb56e5-e32a-4f43-a198-51734a39551e/jobs/d9a606e94fd2490d90bf6eeddd4ab765/driveroutput
jobUuid: fdb559c2-7703-3376-b158-6d761760e592
```