Week 8: Homework 2: Project: Movie Recommendation with MLlib - Collaborative Filtering (implementation 3)

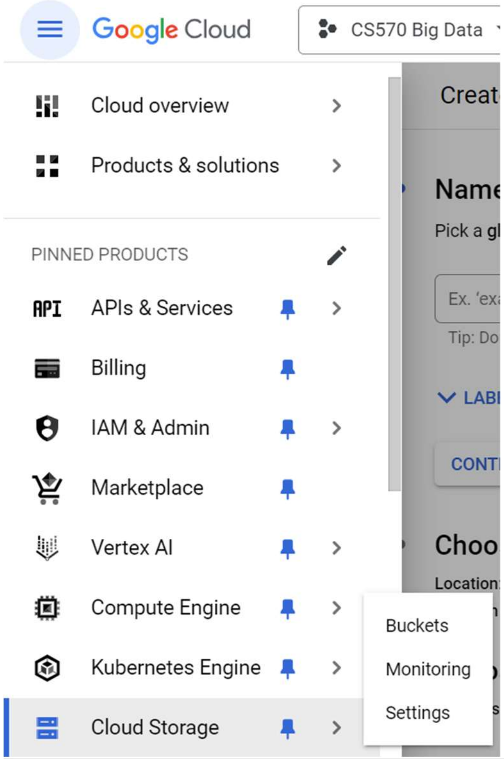
Step-by-Step Guide for Deployment on GCP with Correct File Paths

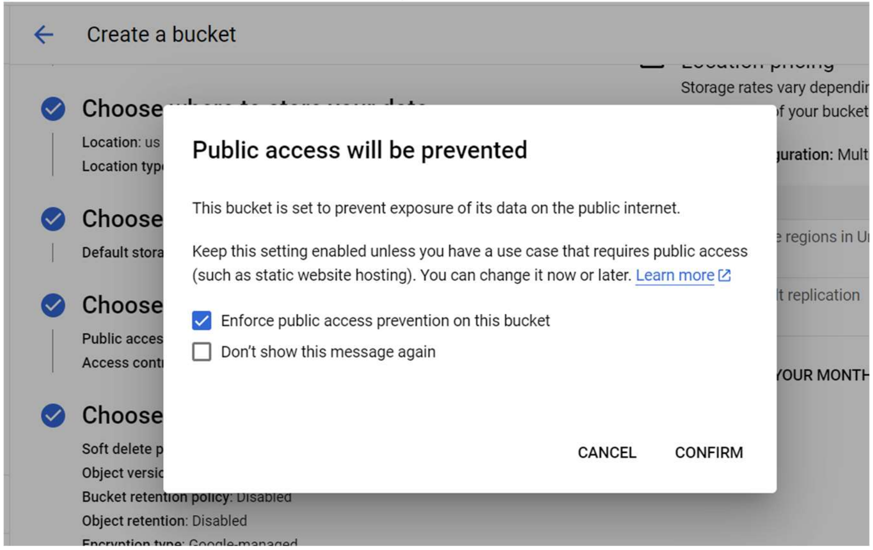
1. Upload Data and Scripts to GCS

Ensure that `movies.csv`, `ratings.csv`, and your PySpark script (e.g., `Recommendation_Engine_MovieLens.py`) are already uploaded to your GCS bucket.

2. Create a Google Cloud Storage (GCS) Bucket

Create a bucket in GCS to store your scripts and data:





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

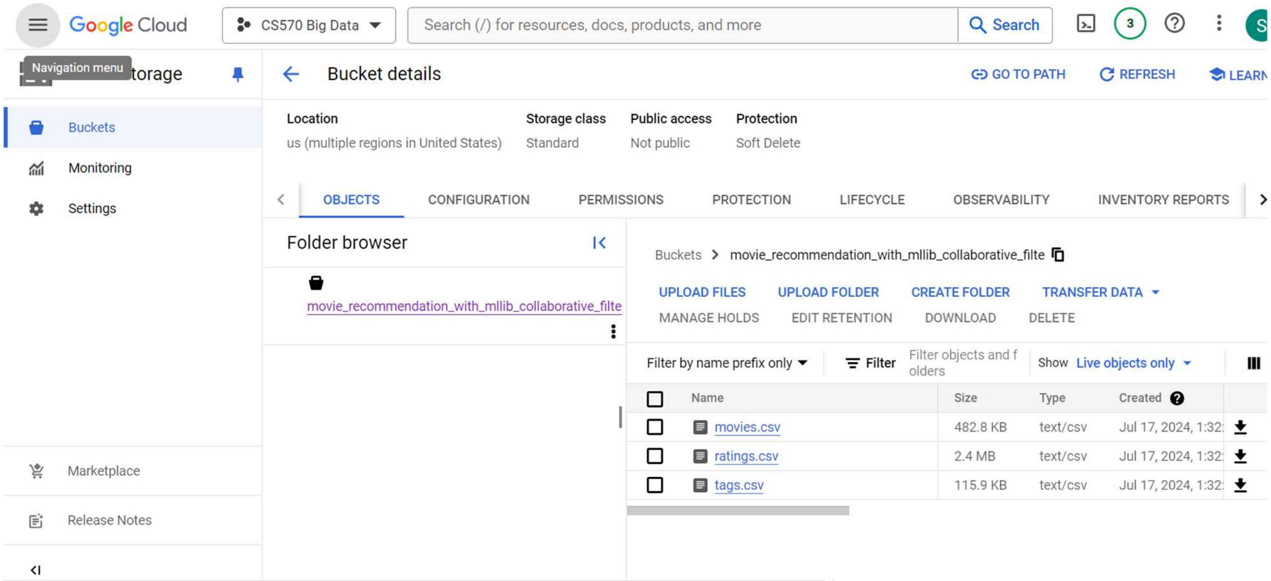
□ Don't show this message again

Bucket retention policy: Disabled

Object retention: Disabled

3. Upload Data and Scripts to GCS

Upload the `movies.csv`, `ratings.csv`, and your PySpark script to your GCS bucket:



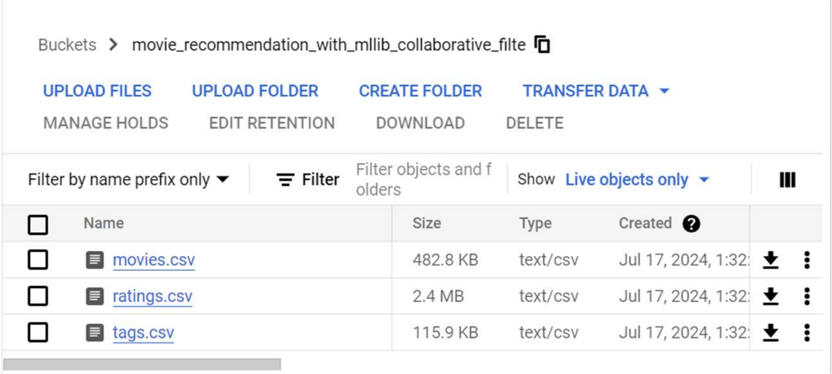
Bucket details

Location	Storage class	Public access	Protection
us (multiple regions in United States)	Standard	Not public	Soft Delete

Buckets > movie_recommendation_with_mllib_collaborative_filte

Show Live objects only

☐	Name	Size	Type	Created
☐	movies.csv	482.8 KB	text/csv	Jul 17, 2024, 1:32:
☐	ratings.csv	2.4 MB	text/csv	Jul 17, 2024, 1:32:
☐	tags.csv	115.9 KB	text/csv	Jul 17, 2024, 1:32:



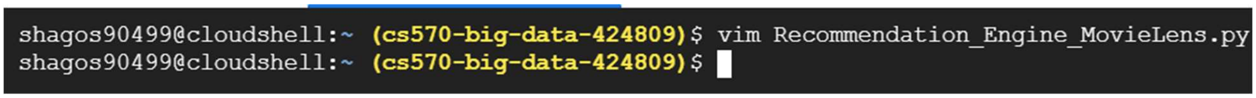
Buckets > movie_recommendation_with_mllib_collaborative_filte

Show Live objects only

☐	Name	Size	Type	Created
☐	movies.csv	482.8 KB	text/csv	Jul 17, 2024, 1:32:
☐	ratings.csv	2.4 MB	text/csv	Jul 17, 2024, 1:32:
☐	tags.csv	115.9 KB	text/csv	Jul 17, 2024, 1:32:

4. Modify the PySpark Script to Use GCS Paths

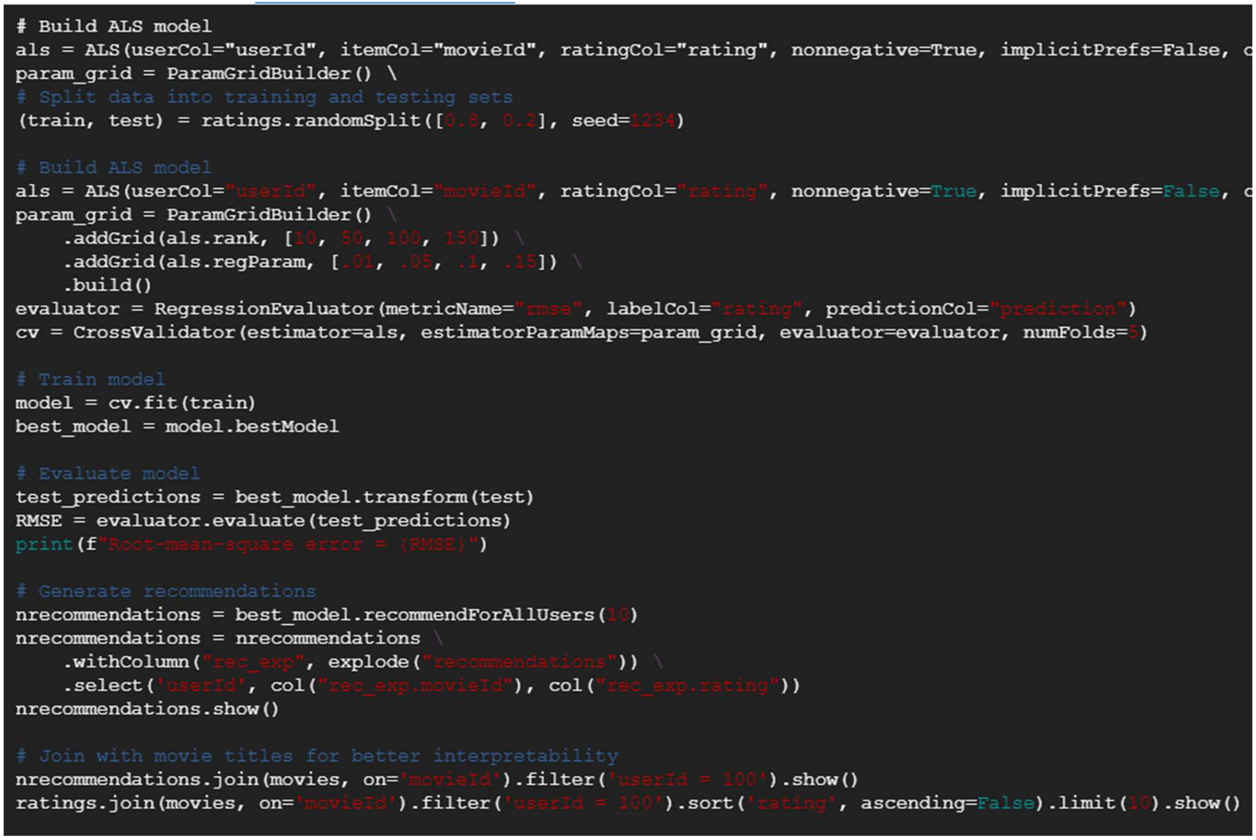
Update your PySpark script to read the files from GCS. Use command-line arguments to pass the paths of the CSV files, enhancing the script's flexibility. Upload the modified script to the bucket:



```
shagos90499@cloudshell:~ (cs570-big-data-424809)$ vim Recommendation_Engine_MovieLens.py
shagos90499@cloudshell:~ (cs570-big-data-424809)$ 
```

Here is the script:

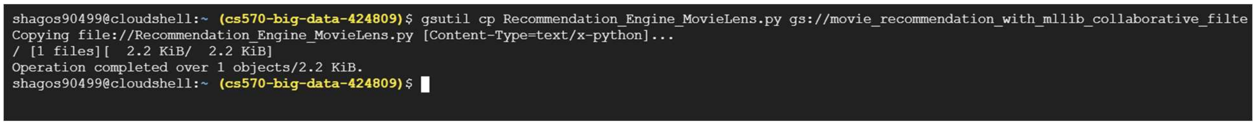
```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, explode
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.recommendation import ALS
from pyspark.ml.tuning import ParamGridBuilder, CrossValidator
import argparse

# Parse command-line arguments
parser = argparse.ArgumentParser()
parser.add_argument('--input_path_movies', required=True)
parser.add_argument('--input_path_ratings', required=True)
args = parser.parse_args()

# Initialize Spark session
spark = SparkSession.builder.appName('Recommendations').getOrCreate()

# Load data from GCS
movies = spark.read.csv(args.input_path_movies, header=True)
ratings = spark.read.csv(args.input_path_ratings, header=True)

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

# Split data into training and testing sets
(train, test) = ratings.randomSplit([0.8, 0.2], seed=1234)

# Build ALS model
als = ALS(userCol="userId", itemCol="movieId", ratingCol="rating",
nonnegative=True, implicitPrefs=False, coldStartStrategy="drop")
param_grid = ParamGridBuilder() \
    .addGrid(als.rank, [10, 50, 100, 150]) \
    .addGrid(als.regParam, [.01, .05, .1, .15]) \
    .build()
evaluator = RegressionEvaluator(metricName="rmse", labelCol="rating",
predictionCol="prediction")
cv = CrossValidator(estimator=als, estimatorParamMaps=param_grid,
evaluator=evaluator, numFolds=5)

# Train model
model = cv.fit(train)
best_model = model.bestModel

# Evaluate model
test_predictions = best_model.transform(test)
RMSE = evaluator.evaluate(test_predictions)
print(f"Root-mean-square error = {RMSE}")

# Generate recommendations
nrecommendations = best_model.recommendForAllUsers(10)
nrecommendations = nrecommendations \
    .withColumn("rec_exp", explode("recommendations")) \
    .select('userId', col("rec_exp.movieId"), col("rec_exp.rating"))
nrecommendations.show()

# Join with movie titles for better interpretability
nrecommendations.join(movies, on='movieId').filter('userId = 100').show()
ratings.join(movies, on='movieId').filter('userId = 100').sort('rating',
ascending=False).limit(10).show()

# Stop Spark session
spark.stop()
```



```
# Build ALS model
als = ALS(userCol="userId", itemCol="movieId", ratingCol="rating", nonnegative=True, implicitPrefs=False, c
param_grid = ParamGridBuilder() \
# Split data into training and testing sets
(train, test) = ratings.randomSplit([0.8, 0.2], seed=1234)

# Build ALS model
als = ALS(userCol="userId", itemCol="movieId", ratingCol="rating", nonnegative=True, implicitPrefs=False, c
param_grid = ParamGridBuilder() \
    .addGrid(als.rank, [10, 50, 100, 150]) \
    .addGrid(als.regParam, [.01, .05, .1, .15]) \
    .build()
evaluator = RegressionEvaluator(metricName="rmse", labelCol="rating", predictionCol="prediction")
cv = CrossValidator(estimator=als, estimatorParamMaps=param_grid, evaluator=evaluator, numFolds=5)

# Train model
model = cv.fit(train)
best_model = model.bestModel

# Evaluate model
test_predictions = best_model.transform(test)
RMSE = evaluator.evaluate(test_predictions)
print(f"Root-mean-square error = {RMSE}")

# Generate recommendations
nrecommendations = best_model.recommendForAllUsers(10)
nrecommendations = nrecommendations \
    .withColumn("rec_exp", explode("recommendations")) \
    .select('userId', col("rec_exp.movieId"), col("rec_exp.rating"))
nrecommendations.show()

# Join with movie titles for better interpretability
nrecommendations.join(movies, on='movieId').filter('userId = 100').show()
ratings.join(movies, on='movieId').filter('userId = 100').sort('rating', ascending=False).limit(10).show()
```

Upload the file:

```
gsutil cp Recommendation_Engine_MovieLens.py
gs://movie_recommendation_with_mllib_collaborative_filte
```

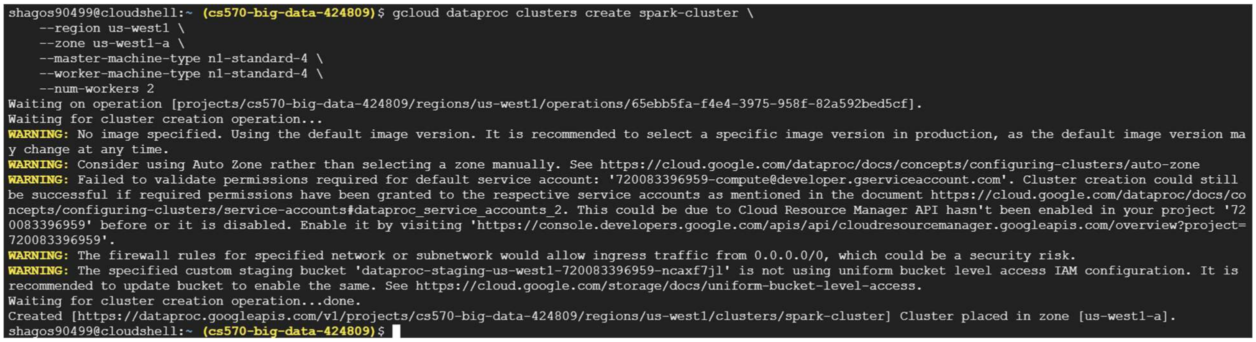


```
shagos90499@cloudshell:~ (cs570-big-data-424809)$ gsutil cp Recommendation_Engine_MovieLens.py gs://movie_recommendation_with_mllib_collaborative_filte
Copying file://Recommendation_Engine_MovieLens.py [Content-Type=text/x-python]...
/ [1 files][  2.2 KiB/  2.2 KiB]
Operation completed over 1 objects/2.2 KiB.
shagos90499@cloudshell:~ (cs570-big-data-424809)$ █
```

5. Create the Cluster with the Desired Configuration

Create a Dataproc cluster:

```
gcloud dataproc clusters create spark-cluster \
    --region us-west1 \
    --zone us-west1-a \
    --master-machine-type n1-standard-4 \
    --worker-machine-type n1-standard-4 \
    --num-workers 2
```



```
shagos90499@cloudshell:~ (cs570-big-data-424809)$ gcloud dataproc clusters create spark-cluster \
    --region us-west1 \
    --zone us-west1-a \
    --master-machine-type n1-standard-4 \
    --worker-machine-type n1-standard-4 \
    --num-workers 2
Waiting on operation [projects/cs570-big-data-424809/regions/us-west1/operations/65ebb5fa-f4e4-3975-958f-82a592bed5cf].
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 ma
y 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: '720083396959-compute@developer.gserviceaccount.com'. Cluster creation could still
be successful if required permissions have been granted to the respective service accounts as mentioned in the document https://cloud.google.com/dataproc/docs/co
ncepts/configuring-clusters/service-accounts#dataproc_service_accounts_2. This could be due to Cloud Resource Manager API hasn't been enabled in your project '72
0083396959' before or it is disabled. Enable it by visiting 'https://console.developers.google.com/apis/api/cloudresourcemanager.googleapis.com/overview?project=
720083396959'.
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.
WARNING: The specified custom staging bucket 'dataproc-staging-us-west1-720083396959-ncaxf7jl' is not using uniform bucket level access IAM configuration. It is
recommended to update bucket to enable the same. See https://cloud.google.com/storage/docs/uniform-bucket-level-access.
Waiting for cluster creation operation...done.
Created [https://dataproc.googleapis.com/v1/projects/cs570-big-data-424809/regions/us-west1/clusters/spark-cluster] Cluster placed in zone [us-west1-a].
shagos90499@cloudshell:~ (cs570-big-data-424809)$
```

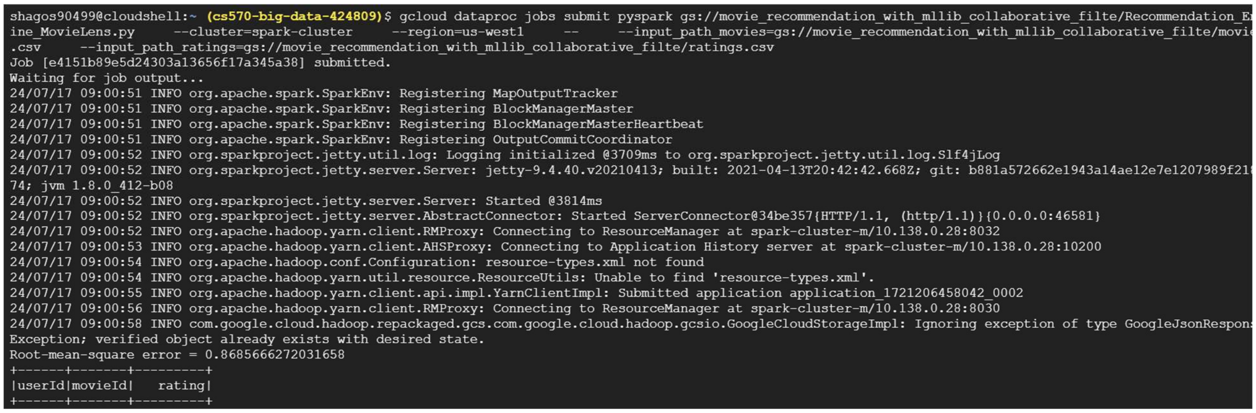
6. Submit the PySpark Job with GCS Paths

- Submit your PySpark job to the Dataproc cluster, specifying the GCS paths for the input files:

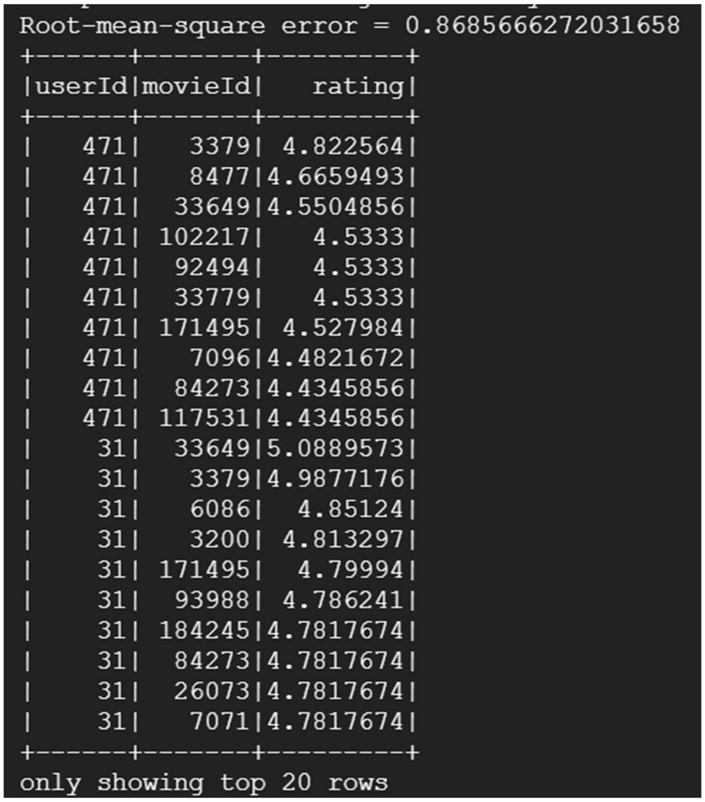
```
gcloud dataproc jobs submit pyspark
gs://movie_recommendation_with_mllib_collaborative_filte/Recommendation_Engin
e_MovieLens.py \
    --cluster=spark-cluster \
    --region=us-west1 \
    -- \
    --
input_path_movies=gs://movie_recommendation_with_mllib_collaborative_filte/mo
vies.csv \
    --
input_path_ratings=gs://movie_recommendation_with_mllib_collaborative_filte/r
atings.csv
```

Replace `your-bucket-name` with the actual name of your GCS bucket. By following these steps, your PySpark script will correctly read the files from GCS when running on GCP Dataproc.

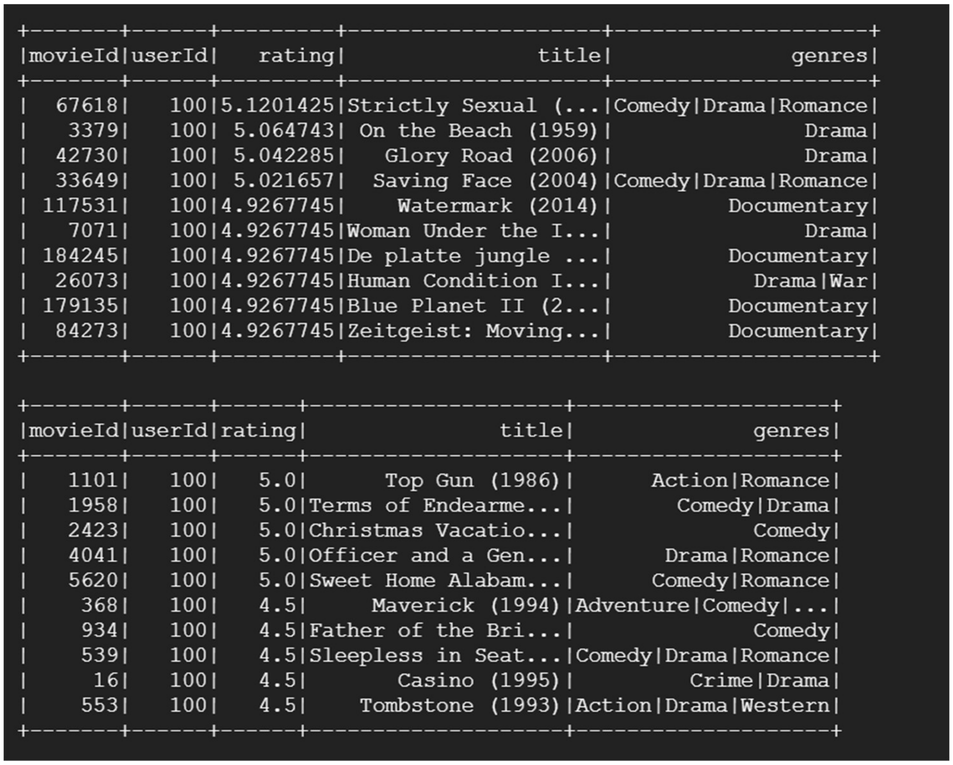
- Test result: root mean square error is calculated.



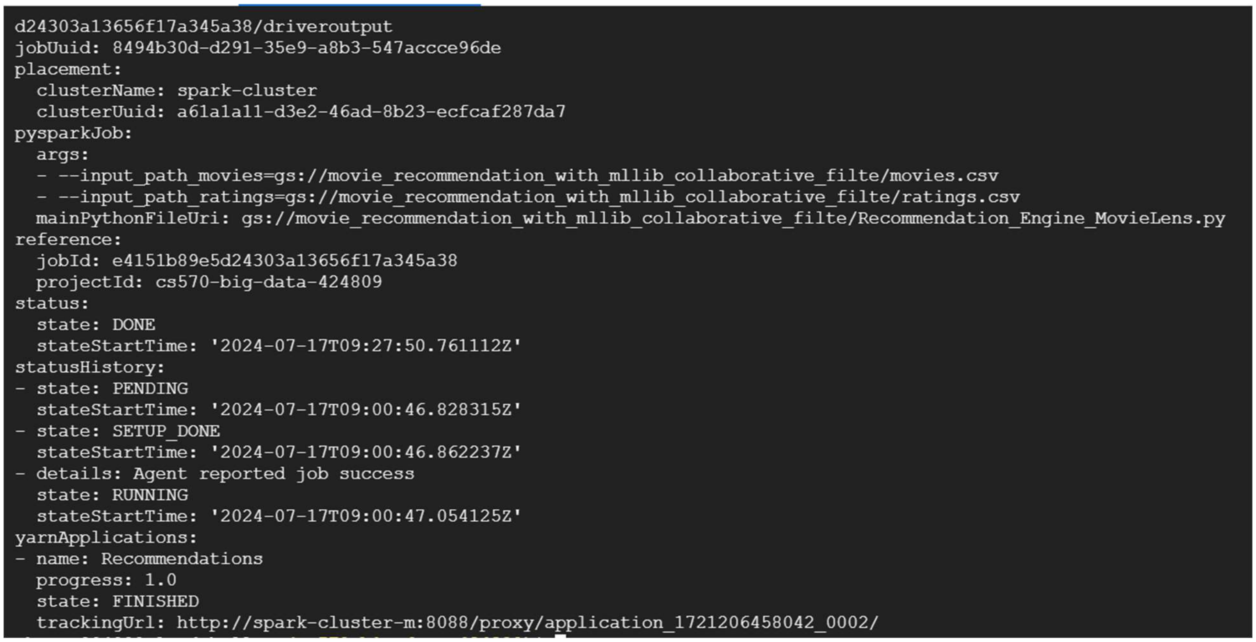
```
shagos90499@cloudshell:~ (cs570-big-data-424809)$ gcloud dataproc jobs submit pyspark gs://movie_recommendation_with_mllib_collaborative_filte/Recommendation_E
ine_MovieLens.py     --cluster=spark-cluster     --region=us-west1     --     --input_path_movies=gs://movie_recommendation_with_mllib_collaborative_filte/movi
.csv     --input_path_ratings=gs://movie_recommendation_with_mllib_collaborative_filte/ratings.csv
Job [e4151b89e5d24303a13656f17a345a38] submitted.
Waiting for job output...
24/07/17 09:00:51 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
24/07/17 09:00:51 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
24/07/17 09:00:51 INFO org.apache.spark.SparkEnv: Registering BlockManagerMasterHeartbeat
24/07/17 09:00:51 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
24/07/17 09:00:52 INFO org.sparkproject.jetty.util.log: Logging initialized @3709ms to org.sparkproject.jetty.util.log.Slf4jLog
24/07/17 09:00:52 INFO org.sparkproject.jetty.server.Server: jetty-9.4.40.v20210413; built: 2021-04-13T20:42:42.668Z; git: b881a572662e1943a14ae12e7e1207989f21
74; jvm 1.8.0_412-b08
24/07/17 09:00:52 INFO org.sparkproject.jetty.server.Server: Started @3814ms
24/07/17 09:00:52 INFO org.sparkproject.jetty.server.AbstractConnector: Started ServerConnector@34be357{HTTP/1.1, (http/1.1)}{0.0.0.0:46581}
24/07/17 09:00:52 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at spark-cluster-m/10.138.0.28:8032
24/07/17 09:00:53 INFO org.apache.hadoop.yarn.client.AHSProxy: Connecting to Application History server at spark-cluster-m/10.138.0.28:10200
24/07/17 09:00:54 INFO org.apache.hadoop.conf.Configuration: resource-types.xml not found
24/07/17 09:00:54 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find 'resource-types.xml'.
24/07/17 09:00:55 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted application application_1721206458042_0002
24/07/17 09:00:56 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at spark-cluster-m/10.138.0.28:8030
24/07/17 09:00:58 INFO com.google.cloud.hadoop.repackaged.gcs.com.google.cloud.hadoop.gcsio.GoogleCloudStorageImpl: Ignoring exception of type GoogleJsonRespon
Exception; verified object already exists with desired state.
Root-mean-square error = 0.8685666272031658
+------+-------+---------+
|userId|movieId|   rating|
+------+-------+---------+
```



```
Root-mean-square error = 0.8685666272031658
+------+-------+---------+
|userId|movieId|   rating|
+------+-------+---------+
|   471|   3379| 4.822564|
|   471|   8477|4.6659493|
|   471|  33649|4.5504856|
|   471| 102217|   4.5333|
|   471|  92494|   4.5333|
|   471|  33779|   4.5333|
|   471| 171495| 4.527984|
|   471|   7096|4.4821672|
|   471|  84273|4.4345856|
|   471| 117531|4.4345856|
|    31|  33649|5.0889573|
|    31|   3379|4.9877176|
|    31|   6086|  4.85124|
|    31|   3200| 4.813297|
|    31| 171495|  4.79994|
|    31|  93988| 4.786241|
|    31| 184245|4.7817674|
|    31|  84273|4.7817674|
|    31|  26073|4.7817674|
|    31|   7071|4.7817674|
+------+-------+---------+
only showing top 20 rows
```



```
+-------+------+---------+--------------------+--------------------+
|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|
+-------+------+------+--------------------+--------------------+
```



```
d24303a13656f17a345a38/driveroutput
jobUuid: 8494b30d-d291-35e9-a8b3-547accce96de
placement:
  clusterName: spark-cluster
  clusterUuid: a61a1a11-d3e2-46ad-8b23-ecfcaf287da7
pysparkJob:
  args:
  - --input_path_movies=gs://movie_recommendation_with_mllib_collaborative_filte/movies.csv
  - --input_path_ratings=gs://movie_recommendation_with_mllib_collaborative_filte/ratings.csv
  mainPythonFileUri: gs://movie_recommendation_with_mllib_collaborative_filte/Recommendation_Engine_MovieLens.py
reference:
  jobId: e4151b89e5d24303a13656f17a345a38
  projectId: cs570-big-data-424809
status:
  state: DONE
  stateStartTime: '2024-07-17T09:27:50.761112Z'
statusHistory:
- state: PENDING
  stateStartTime: '2024-07-17T09:00:46.828315Z'
- state: SETUP_DONE
  stateStartTime: '2024-07-17T09:00:46.862237Z'
- details: Agent reported job success
  state: RUNNING
  stateStartTime: '2024-07-17T09:00:47.054125Z'
yarnApplications:
- name: Recommendations
  progress: 1.0
  state: FINISHED
  trackingUrl: http://spark-cluster-m:8088/proxy/application_1721206458042_0002/
```