Movie Recommendation with MLlib

Natnael Haile
ID: 20007
July 18, 2024

Table of Contents

Introduction

Design

Implementation

Test

Enhancement Ideas

Conclusion

References

Introduction

- **Title:** Movie Recommendation System with MLlib
- **Objective:** Develop a collaborative filtering model for personalized movie recommendations using MLlib on GCP.
- **Technologies Used:**
  - PySpark, GCS, Google Dataproc, MLlib
- **Purpose:** Enhance user experience with personalized recommendations.
- **Challenges:**
  - Handling large datasets
  - Scalability of the recommendation engine
  - Efficient processing and model training

Design: System Architecture

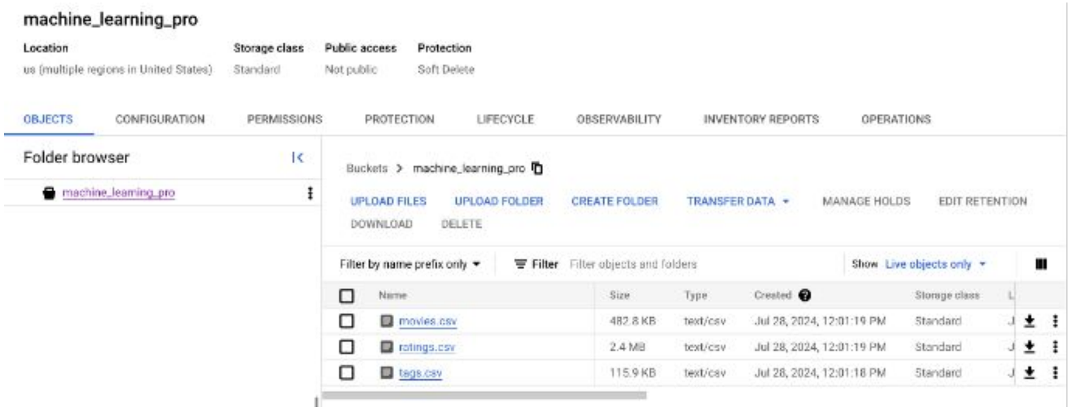
- **Components:**
  - **Data Storage:** Google Cloud Storage (GCS) for movies and ratings data.
  - **Processing:** Google Dataproc for scalable data processing.
  - **Modeling:** MLlib for collaborative filtering model.
- **Workflow:**
  - Data ingestion from GCS
  - Data processing and cleaning
  - Model training and evaluation
  - Deployment and predictions

Design: Data Flow

- **Step 1:** Data Upload
  - Movies and ratings datasets uploaded to GCS.
- **Step 2:** Data Processing
  - Data read into Spark DataFrames.
  - Data transformation and preparation for modeling.
- **Step 3:** Model Training
  - Use ALS (Alternating Least Squares) algorithm.
  - Train model on ratings data.
- **Step 4:** Prediction and Recommendation
  - Generate recommendations based on trained model.
  - Output results for further use or analysis.

Implementation: Data Upload to GCS

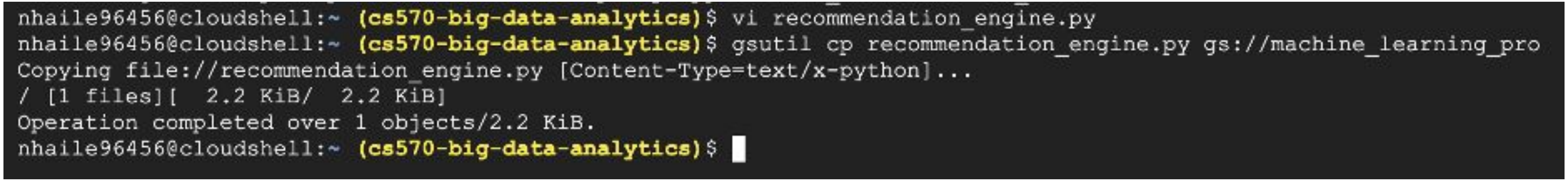
- **Steps:**
  - Create GCS bucket.
  - Upload datasets (movies.csv, ratings.csv).
  - Upload PySpark script (recommendation_engine.py).
- **Commands:**
  - gsutil cp movies.csv gs://machine_learning_pro
  - gsutil cp ratings.csv gs://machine_learning_pro
  - gsutil cp recommendation_engine.py gs://machine_learning_pro



machine_learning_pro

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

□	Name	Size	Type	Created	Storage class	L
□	movies.csv	482.8 KB	text/csv	Jul 28, 2024, 12:01:19 PM	Standard	J
□	ratings.csv	2.4 MB	text/csv	Jul 28, 2024, 12:01:19 PM	Standard	J
□	tags.csv	115.9 KB	text/csv	Jul 28, 2024, 12:01:18 PM	Standard	J

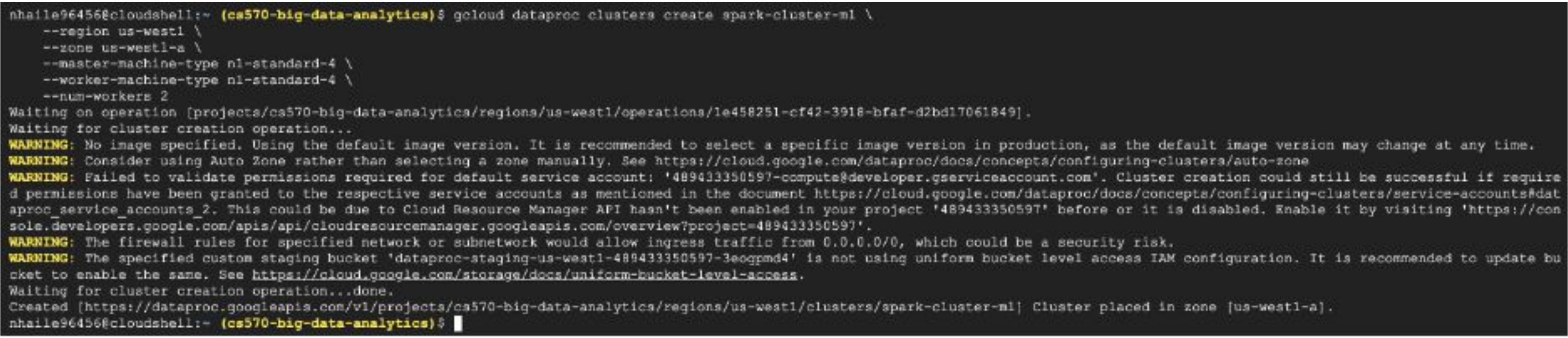


```
nhaile96456@cloudshell:~ (cs570-big-data-analytics)$ vi recommendation_engine.py
nhaile96456@cloudshell:~ (cs570-big-data-analytics)$ gsutil cp recommendation_engine.py gs://machine_learning_pro
Copying file://recommendation_engine.py [Content-Type=text/x-python]...
/ [1 files][  2.2 KiB/  2.2 KiB]
Operation completed over 1 objects/2.2 KiB.
nhaile96456@cloudshell:~ (cs570-big-data-analytics)$
```

Implementation: Dataproc Cluster Configuration

- **Cluster Setup:**
  - Specify region and zone.
  - Define machine types for master and worker nodes.
  - Set the number of workers.
- **Commands:**

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

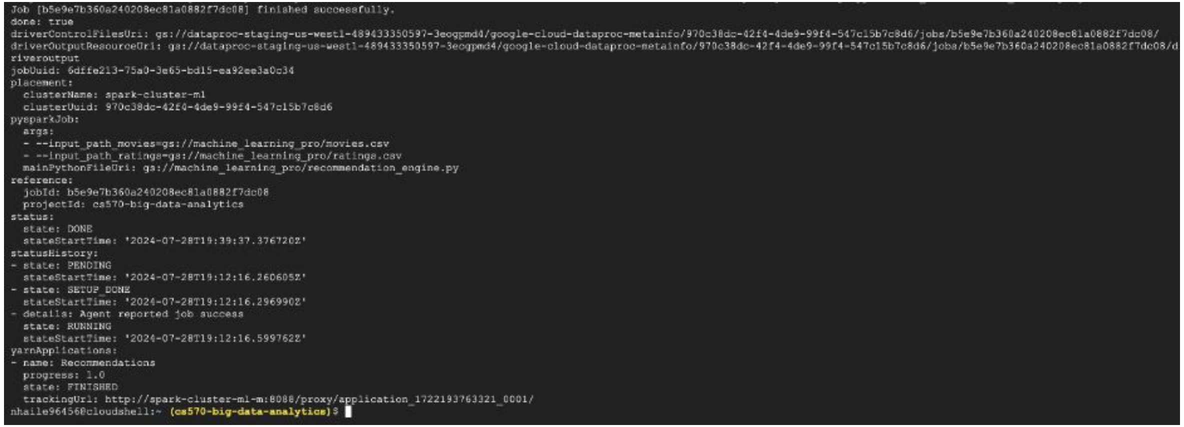


```
nhaile96456@cloudshell:~ (cs570-big-data-analytics)$ gcloud dataproc clusters create spark-cluster-ml \
    --region us-west1 \
    --zone us-west1-a \
    --master-machine-type n1-standard-4 \
    --worker-machine-type n1-standard-4 \
    --num-workers 2
Waiting on operation [projects/cs570-big-data-analytics/regions/us-west1/operations/1e458251-cf42-3918-bfaf-d2bd17061849].
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: '489433350597-compute@developer.gserviceaccount.com'. Cluster creation could still be successful if require
d permissions have been granted to the respective service accounts as mentioned in the document https://cloud.google.com/dataproc/docs/concepts/configuring-clusters/service-accounts#dat
aproc_service_accounts_2. This could be due to Cloud Resource Manager API hasn't been enabled in your project '489433350597' before or it is disabled. Enable it by visiting 'https://con
sole.developers.google.com/apis/api/cloudresourcemanager.googleapis.com/overview?project=489433350597'.
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-west1-489433350597-3eogpmd4' is not using uniform bucket level access IAM configuration. It is recommended to update bu
cket 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-analytics/regions/us-west1/clusters/spark-cluster-ml] Cluster placed in zone [us-west1-a].
nhaile96456@cloudshell:~ (cs570-big-data-analytics)$
```

Implementation: Job Submission and Execution

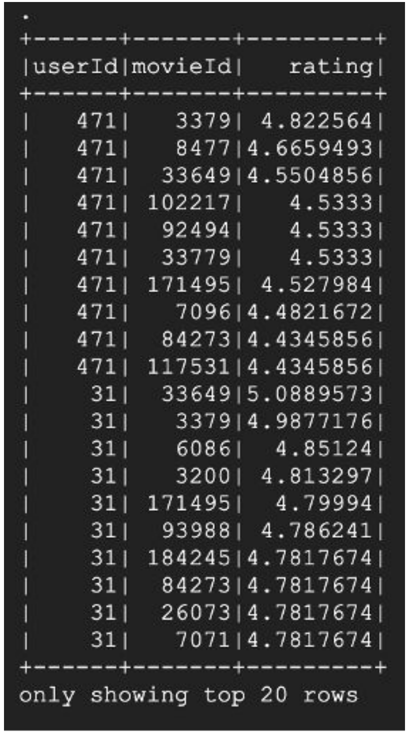
- **Submit PySpark Job:**
  - Specify PySpark script path in GCS.
  - Provide input data paths.
- **Commands:**

```
gcloud dataproc jobs submit pyspark gs://machine_learning_pro/recommendation_engine.py  --cluster=spark-cluster-ml  --region=us-west1 --
--input_path_movies=gs://machine_learning_pro/movies.csv  --input_path_ratings=gs://machine_learning_pro/ratings.csv
```

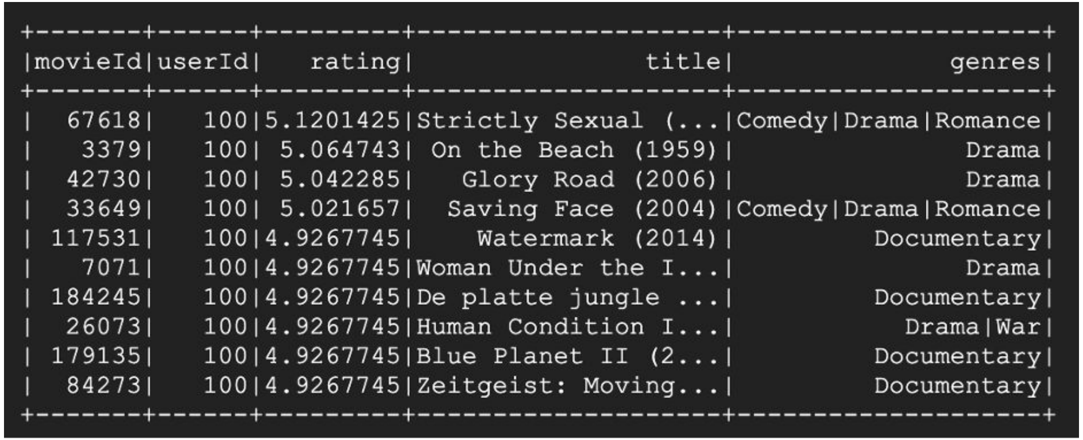


```
Job [b5e9e7b360a240208ec81a0882f7dc08] finished successfully.
done: true
driverControlFilesUri: gs://dataproc-staging-us-west1-489433350597-3eogpmd4/google-cloud-dataproc-metainfo/970c38dc-42f4-4de9-99f4-547c15b7c8d6/jobs/b5e9e7b360a240208ec81a0882f7dc08/
driverOutputResourceUri: gs://dataproc-staging-us-west1-489433350597-3eogpmd4/google-cloud-dataproc-metainfo/970c38dc-42f4-4de9-99f4-547c15b7c8d6/jobs/b5e9e7b360a240208ec81a0882f7dc08/d
riveroutput
jobUuid: 6dffe213-75a0-3e65-bd15-ea92ee3a0c34
placement:
  clusterName: spark-cluster-ml
  clusterUuid: 970c38dc-42f4-4de9-99f4-547c15b7c8d6
pysparkJob:
  args:
  - --input_path_movies=gs://machine_learning_pro/movies.csv
  - --input_path_ratings=gs://machine_learning_pro/ratings.csv
  mainPythonFileUri: gs://machine_learning_pro/recommendation_engine.py
reference:
  jobId: b5e9e7b360a240208ec81a0882f7dc08
  projectId: cs570-big-data-analytics
status:
  state: DONE
  stateStartTime: '2024-07-28T19:39:37.376720Z'
statusHistory:
- state: PENDING
  stateStartTime: '2024-07-28T19:12:16.260605Z'
- state: SETUP_DONE
  stateStartTime: '2024-07-28T19:12:16.296990Z'
- details: Agent reported job success
  state: RUNNING
  stateStartTime: '2024-07-28T19:12:16.599762Z'
yarnApplications:
- name: Recommendations
  progress: 1.0
  state: FINISHED
  trackingUrl: http://spark-cluster-ml-m:8088/proxy/application_1722193763321_0001/
nhaile96456@cloudshell:~ (cs570-big-data-analytics)$
```

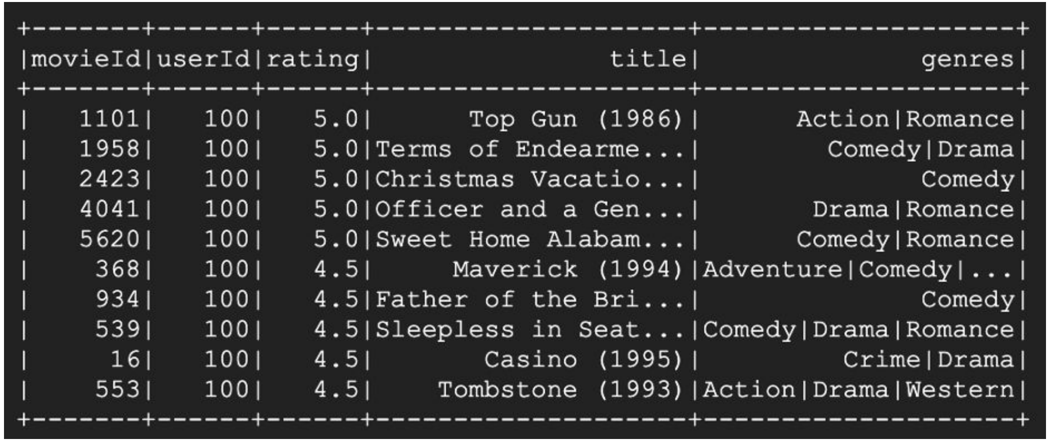
Test



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



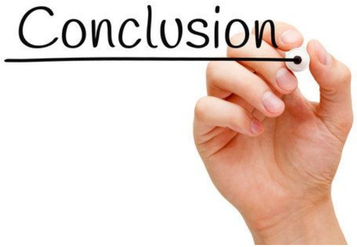
```
+-------+------+------+--------------------+--------------------+
|movieId|userId|rating|               title|              genres|
+-------+------+------+--------------------+--------------------+
|   1101|   100|   5.0|      Top Gun (1986)|      Action|Romance|
|   1958|   100|   5.0|Terms of Endearme...|        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|
+-------+------+------+--------------------+--------------------+
```

Enhancement Ideas

- Improve model with parameter tuning and feature engineering.
- Integrate additional data sources and real-time data processing.
- Implement auto-scaling for clusters and explore distributed storage.
- Develop a personalized interface and feedback mechanism.

Conclusion

- Successfully developed and deployed a movie recommendation system using MLlib on GCP.
- Efficiently handled large datasets and trained a collaborative filtering model.
- Gained insights into scalable infrastructure and model optimization.
- Future work: explore advanced techniques and continuously improve based on user feedback.



References

Movie Recommendation with Spark MLlib

Collaborative Filtering - RDD-based API

Collaborative Filtering for Movie Recommendations

Movie Recommendation with Collaborative Filtering in …

Collaborative Filtering - Spark 2.2.0 Documentation

GitHub Link

- https://github.com/cur10usityDrives/Big-Data/new/main/PySpark/Movie-Recommendation-with-MLlib-implementation-3

THANK YOU