



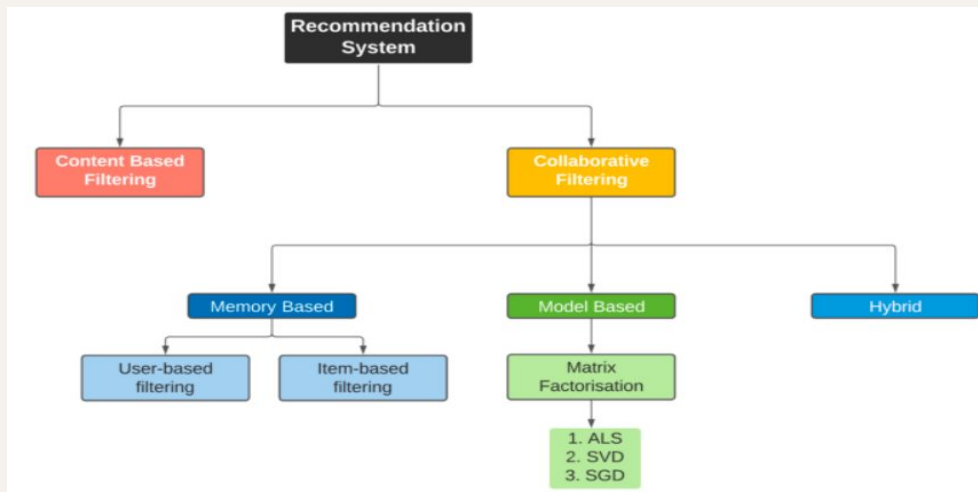
Movie Recommendation with MLlib (Implementation 1)

Ashritha Lopelli



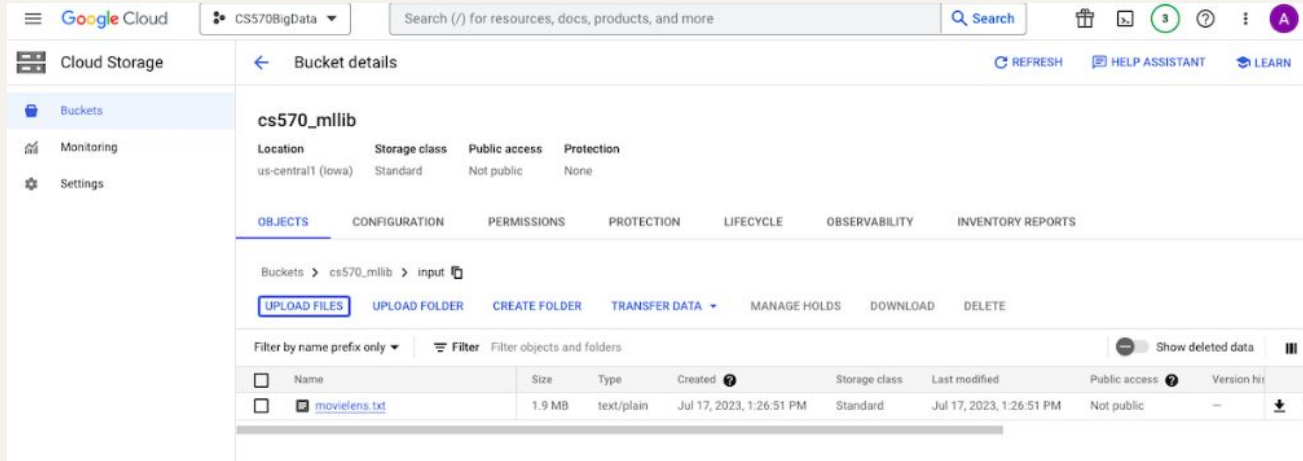
Introduction

- The MovieLens dataset is widely used as a benchmark for evaluating recommendation systems.
- The program utilizes PySpark, a Python library designed for distributed data processing, to leverage collaborative filtering techniques.
- This Python program showcases the implementation of a recommendation engine that utilizes the ALS (Alternating Least Squares) model on the MovieLens dataset.
- By applying collaborative filtering, the program generates personalized movie recommendations for users based on their preferences and historical ratings.



Implementation

1. Create a bucket on the GCP and load the movielens.txt file to the bucket.



2. On the cloud shell Download this movielens.txt file

```
gsutil cp gs://cs570_mllib/input/movielens.txt .
```

Implementation

3. Convert the file to the desired format. This format has only the userid, movieid, rating from the movielens.txt file.


```
cat movielens.txt | while read userid movieid rating timestamp; do echo  
    "${userid},${movieid},${rating}"; done > converted_data.txt
```

4. To check if the file is converted to the desired format run the command:


```
cat converted_data.txt
```


5. Now create the dataproc cluster.

- In the Navigation menu, click on "Dataproc".
- Click on the "Create cluster" button to create a new cluster.
- Give the necessary details such as Cluster name, Region, Zone and Cluster type.
- Create cluster
- The cluster is now created and running.


 Dataproc


Jobs on Clusters


 Clusters


 Jobs


Clusters

 CREATE CLUSTER



 REFRESH

 START

 STOP

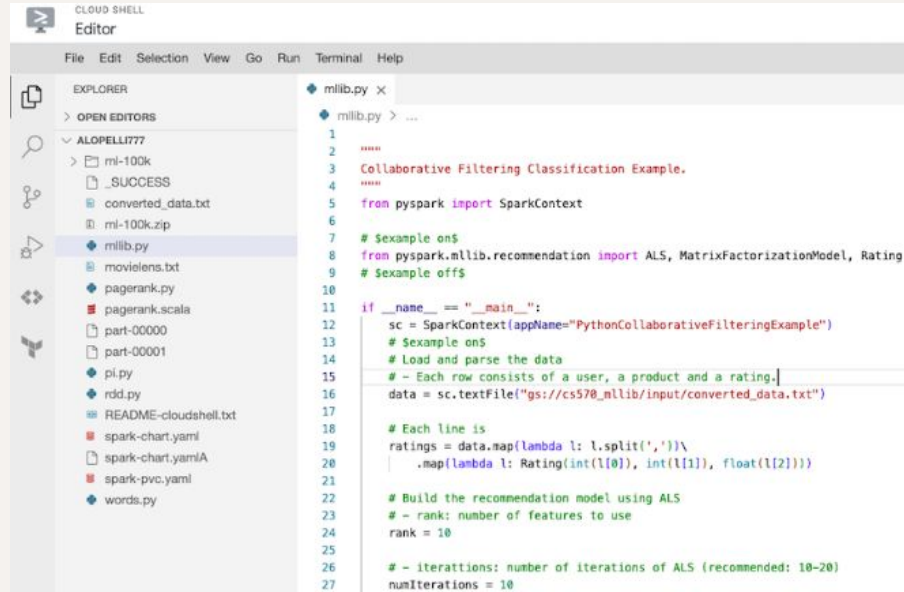
 Filter

Search clusters, press Enter

| <input type="checkbox"/> | Name  | Status | Region | Zone | Total worker nodes |
|--------------------------|--|---|-------------|---------------|--------------------|
| <input type="checkbox"/> | cluster-ef2d |  Running | us-central1 | us-central1-a | 0 |

Implementation

6. Now click on the activate cloud shell and open the editor window, in where we have to insert the code (the code file is attached in the folder and also change the path of the .txt file).



```
Cloud Shell
Editor

File Edit Selection View Go Run Terminal Help

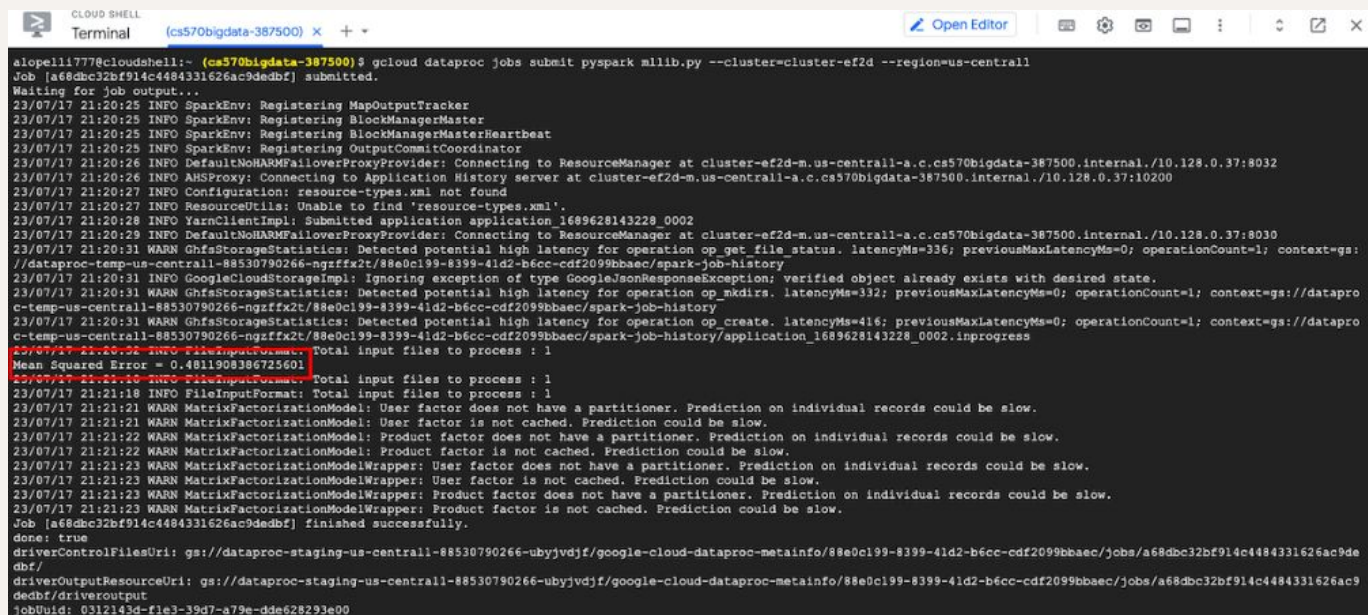
EXPLORER
> OPEN EDITORS
ALOPELL777
  ml-100k
    _SUCCESS
    converted_data.txt
    ml-100k.zip
    ml-lib.py
    movielens.txt
    pagerank.py
    pagerank.scala
    part-00000
    part-00001
    pi.py
    rdd.py
    README-cloudshell.txt
    spark-chart.yaml
    spark-chart.yamlA
    spark-pvc.yaml
    words.py

ml-lib.py x
ml-lib.py > ...
1
2
3 Collaborative Filtering Classification Example.
4
5 from pyspark import SparkContext
6
7 # $example on$
8 from pyspark.mllib.recommendation import ALS, MatrixFactorizationModel, Rating
9 # $example off$
10
11 if __name__ == "__main__":
12     sc = SparkContext(appName="PythonCollaborativeFilteringExample")
13     # $example on$
14     # Load and parse the data
15     # - Each row consists of a user, a product and a rating.
16     data = sc.textFile("gs://cs570_mllib/input/converted_data.txt")
17
18     # Each line is
19     ratings = data.map(lambda l: l.split(','))\
20         .map(lambda l: Rating(int(l[0]), int(l[1]), float(l[2])))
21
22     # Build the recommendation model using ALS
23     # - rank: number of features to use
24     rank = 10
25
26     # - iterations: number of iterations of ALS (recommended: 10-20)
27     numIterations = 10
```

Implementation

7. To execute for the output Run the command on the cloud shell

```
gcloud dataproc jobs submit pyspark milib.py --cluster-cluster-ef2d --region us-central1
```



```
alopelli777@cloudshell:~ (cs570bigdata-387500) x + *  
Terminal (cs570bigdata-387500) x + *  
alopelli777@cloudshell:~ (cs570bigdata-387500) $ gcloud dataproc jobs submit pyspark milib.py --cluster=cluster-ef2d --region=us-central1  
Job [a68dbc32bf914c4484331626ac9dedbf] submitted.  
Waiting for job output...  
23/07/17 21:20:25 INFO SparkEnv: Registering MapOutputTracker  
23/07/17 21:20:25 INFO SparkEnv: Registering BlockManagerMaster  
23/07/17 21:20:25 INFO SparkEnv: Registering BlockManagerMasterHeartbeat  
23/07/17 21:20:25 INFO SparkEnv: Registering OutputCommitCoordinator  
23/07/17 21:20:26 INFO DefaultNoHARMFalloverProxyProvider: Connecting to ResourceManager at cluster-ef2d-m.us-central1-a.c.cs570bigdata-387500.internal./10.128.0.37:8032  
23/07/17 21:20:26 INFO AHSPROxy: Connecting to Application History server at cluster-ef2d-m.us-central1-a.c.cs570bigdata-387500.internal./10.128.0.37:10200  
23/07/17 21:20:27 INFO Configuration: resource-types.xml not found  
23/07/17 21:20:27 INFO ResourceUtils: Unable to find 'resource-types.xml'.  
23/07/17 21:20:28 INFO YarnClientImpl: Submitted application application_1689628143228_0002  
23/07/17 21:20:29 INFO DefaultNoHARMFalloverProxyProvider: Connecting to ResourceManager at cluster-ef2d-m.us-central1-a.c.cs570bigdata-387500.internal./10.128.0.37:8032  
23/07/17 21:20:31 WARN GHfsStorageStatistics: Detected potential high latency for operation on get_file_status. latencyMs=336; previousMaxLatencyMs=0; operationCount=1; context=gs://dataproc-temp-us-central1-88530790266-nginxffx2t/88e0c199-8399-41d2-b6cc-cdf2099bbaec/spark-job-history  
23/07/17 21:20:31 INFO GoogleCloudStorageImpl: Ignoring exception of type GoogleJsonResponseException: verified object already exists with desired state.  
23/07/17 21:20:31 WARN GHfsStorageStatistics: Detected potential high latency for operation on mkdirs. latencyMs=332; previousMaxLatencyMs=0; operationCount=1; context=gs://dataproc-temp-us-central1-88530790266-nginxffx2t/88e0c199-8399-41d2-b6cc-cdf2099bbaec/spark-job-history  
23/07/17 21:20:31 WARN GHfsStorageStatistics: Detected potential high latency for operation on create. latencyMs=416; previousMaxLatencyMs=0; operationCount=1; context=gs://dataproc-temp-us-central1-88530790266-nginxffx2t/88e0c199-8399-41d2-b6cc-cdf2099bbaec/spark-job-history/application_1689628143228_0002.inprogress  
23/07/17 21:20:32 INFO FileInputFormat: Total input files to process : 1  
Mean Squared Error = 0.481908386725601  
23/07/17 21:21:10 INFO FileInputFormat: Total input files to process : 1  
23/07/17 21:21:18 INFO FileInputFormat: Total input files to process : 1  
23/07/17 21:21:21 WARN MatrixFactorizationModel: User factor does not have a partitioner. Prediction on individual records could be slow.  
23/07/17 21:21:21 WARN MatrixFactorizationModel: User factor is not cached. Prediction could be slow.  
23/07/17 21:21:22 WARN MatrixFactorizationModel: Product factor does not have a partitioner. Prediction on individual records could be slow.  
23/07/17 21:21:22 WARN MatrixFactorizationModel: Product factor is not cached. Prediction could be slow.  
23/07/17 21:21:23 WARN MatrixFactorizationModelWrapper: User factor does not have a partitioner. Prediction on individual records could be slow.  
23/07/17 21:21:23 WARN MatrixFactorizationModelWrapper: User factor is not cached. Prediction could be slow.  
23/07/17 21:21:23 WARN MatrixFactorizationModelWrapper: Product factor does not have a partitioner. Prediction on individual records could be slow.  
23/07/17 21:21:23 WARN MatrixFactorizationModelWrapper: Product factor is not cached. Prediction could be slow.  
Job [a68dbc32bf914c4484331626ac9dedbf] finished successfully.  
done: true  
driverControlFilesUri: gs://dataproc-staging-us-central1-88530790266-ubyjvdfj/google-cloud-dataproc-metainfo/88e0c199-8399-41d2-b6cc-cdf2099bbaec/jobs/a68dbc32bf914c4484331626ac9dedbf/driveroutput  
driverOutputResourceUri: gs://dataproc-staging-us-central1-88530790266-ubyjvdfj/google-cloud-dataproc-metainfo/88e0c199-8399-41d2-b6cc-cdf2099bbaec/jobs/a68dbc32bf914c4484331626ac9dedbf/driveroutput  
jobUuid: 0312143d-f1e3-39d7-a79e-dde28293e00
```



Thank You!!!