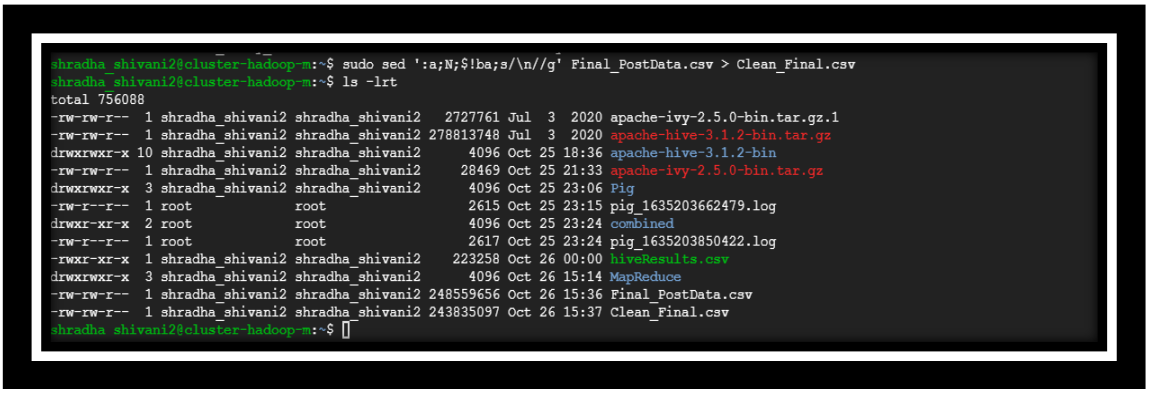
Shradha Shivani (21261104)

In this document per-user TF-IDF of the top 10 terms for each of the top 10 users is being caluculated.

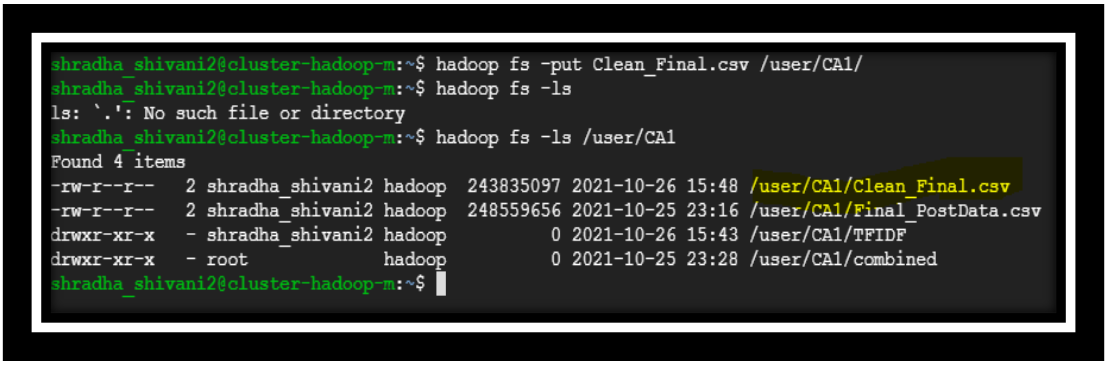
Steps Prior to calculate TF-IDF:

1. Dataset Final_PostsData.csv is cleaned using basic sed command to achieve a clean csv file. Using the sed command new lines with spaces have been replaced.

   Command: sed ':a;N;$!ba;s/\n//g' Final_PostData.csv > Clean_Final.csv

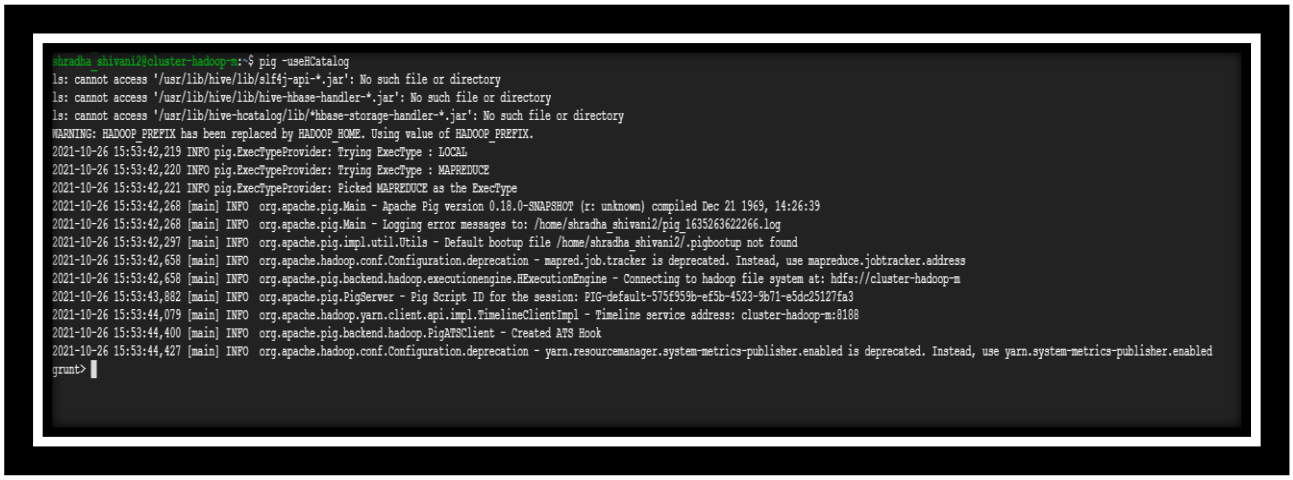


2. Basic cleaned file 'Clean_Final.csv' is moved to '/user/CA1' in HDFS
   Command: hadoop fs -put Clean_Final.csv /user/CA1



3. Logged in to pig terminal using HCatalog. The Command used is pig -useHCatalog.

Shradha Shivani (21261104)



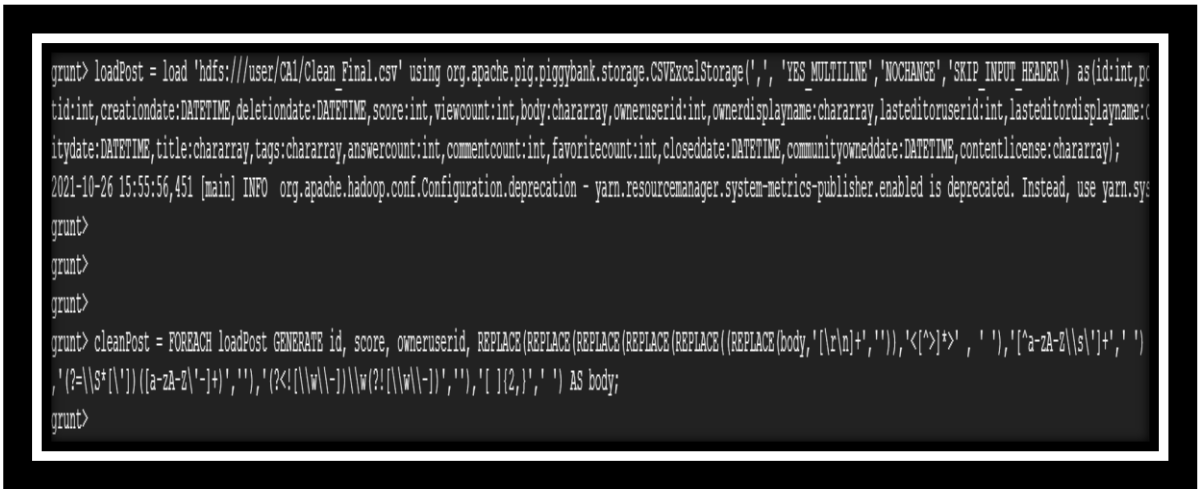
4. Using Pig, the data present in Clean_Final.csv is further cleaned. New lines, tab or carriage return characters, single characters, special symbols etc are replaced with space in 'Body' column. Additionally, the dataset is filtered for NOT NULL OWNERUSERID tuples.

Command:

```
grunt> loadPost = load 'hdfs:///user/CA1/Clean_Final.csv' using org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'YES_MULTILINE','NOCHANGE','SKIP_INPUT_HEADER') as(id:int,posttypeid:int,acceptedanswerid:int, parentid:int,creationdate:DATETIME,deletiondate:DATETIME,score:int,viewcount:int,body:chararray,owneruserid:int,ownerdisplayname:chararray,lasteditoruserid:int,lasteditordisplayname:chararray,lasteditdate:DATETIME,lastactivitydate:DATETIME,title:chararray,tags:chararray,answercount:int,commentcount:int,favoritecount:int,closeddate:DATETIME,communityowneddate:DATETIME,contentlicense:chararray);
2021-10-26 15:55:56,451 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
grunt>

grunt> cleanPost = FOREACH loadPost GENERATE id, score, owneruserid, REPLACE(REPLACE(REPLACE(REPLACE(REPLACE((REPLACE(body,'[\r\n]+',' ')),'<[^>]*>' , ' '),'[^a-zA-Z\\s\']+',' '),'(?=\\S*[\'])([a-zA-Z\'-]+)',''),'(?<![\\w\\-])\\w(?![\\w\\-])',''),'[ ]{2,}',' ') AS body;
```

Shradha Shivani (21261104)



grunt> filter_data = FILTER cleanPost BY (owneruserid is not null);

Computation of TF-IDF:

1. In this section, the users are grouped together, their total score of their various posts is calculated and arranged in descending order . The result is just limited to top 10.So, in other terms top 10 posts of top 10 users are selected as per their total score.

   Command:

   grunt> select_distinct_users_post = GROUP filter_data BY owneruserid;
   grunt> select_users_by_max_score = FOREACH select_distinct_users_post GENERATE group AS userid, SUM(filter_data.score) AS maxscore;
   grunt> select_users_by_max_score_desc_order = ORDER select_users_by_max_score BY maxscore DESC;
   grunt> select_data_limit_10 = LIMIT select_users_by_max_score_desc_order 10;
   grunt> select_top_10_user_id = FOREACH select_data_limit_10 GENERATE userid;
   grunt> select_posts_by_10_users = JOIN filter_data BY owneruserid, select_top_10_user_id BY userid;
   grunt> select_posts_by_10_users = FOREACH select_posts_by_10_users GENERATE filter_data::owneruserid, LOWER(TRIM(REPLACE(filter_data::body,'[ ]{2,}',' '))) AS filter_data::body

```
grunt> select_distinct_users_post = GROUP filter_data BY owneruserid;
grunt> select_users_by_max_score = FOREACH select_distinct_users_post GENERATE group AS userid, SUM(filter_data.score) AS maxscore;
grunt> select_users_by_max_score_desc_order = ORDER select_users_by_max_score BY maxscore DESC;
grunt> select_data_limit_10 = LIMIT select_users_by_max_score_desc_order 10;
grunt> select_top_10_user_id = FOREACH select_data_limit_10 GENERATE userid;
grunt> select_posts_by_10_users = JOIN filter_data BY owneruserid, select_top_10_user_id BY userid;
grunt> select_posts_by_10_users = FOREACH select_posts_by_10_users GENERATE filter_data::owneruserid, LOWER(TRIM(REPLACE(filter_data::body,'[ ]{2,}',' '))) AS filter_data::body;

grunt>
```

2. Next, the resultant set is stored in a folder named ‘TFIDF’ under ‘/user/CA1/’

   Command:
   Grunt>STORE select_posts_by_10_users INTO '/user/CA1/TFIDF' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','NOCHANGE','SKIP_OUTPUT_HEADER');

```
ystem-metrics-publisher.enabled
grunt> STORE select_posts_by_10_users INTO '/user/CA1/TFIDF' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',','NO_MULTILINE','NOCHANGE','SKIP_OUTPUT_HEADER');
```

3. Successful logs are obtained for data storing in TFIDF location and the files part-r-00000 and _SUCCESS log can be seen under this location.

```
2021-10-26 16:49:40,711 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at cluster-hadoop-m/10.164.0.3:8032
2021-10-26 16:49:40,712 [main] INFO  org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at cluster-hadoop-m/10.164.0.3:10200
2021-10-26 16:49:40,714 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job histo
ry server
2021-10-26 16:49:40,736 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at cluster-hadoop-m/10.164.0.3:8032
2021-10-26 16:49:40,736 [main] INFO  org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at cluster-hadoop-m/10.164.0.3:10200
2021-10-26 16:49:40,738 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job histo
ry server
2021-10-26 16:49:40,754 [main] WARN  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Encountered Warning FIELD_DISCARDED_TYPE_CONVERSION_FAILED
 184 time(s).
2021-10-26 16:49:40,754 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
grunt>
grunt>
grunt>
grunt>
grunt> fs -ls /user/CA1/TFIDF
Found 2 items
-rw-r--r--   2 shradha_shivani2 hadoop          0 2021-10-26 16:49 /user/CA1/TFIDF/_SUCCESS
-rw-r--r--   2 shradha_shivani2 hadoop     125193 2021-10-26 16:49 /user/CA1/TFIDF/part-r-00000
grunt>
```

Implementing TFIDF:

1. 1st the file stored in ‘/user/CA1/TFIDF/part-r-00000’ is merged with the local folder.
   Command: hadoop fs -getmerge /user/CA1/TFIDF/part-r-00000/ TFIDF.csv

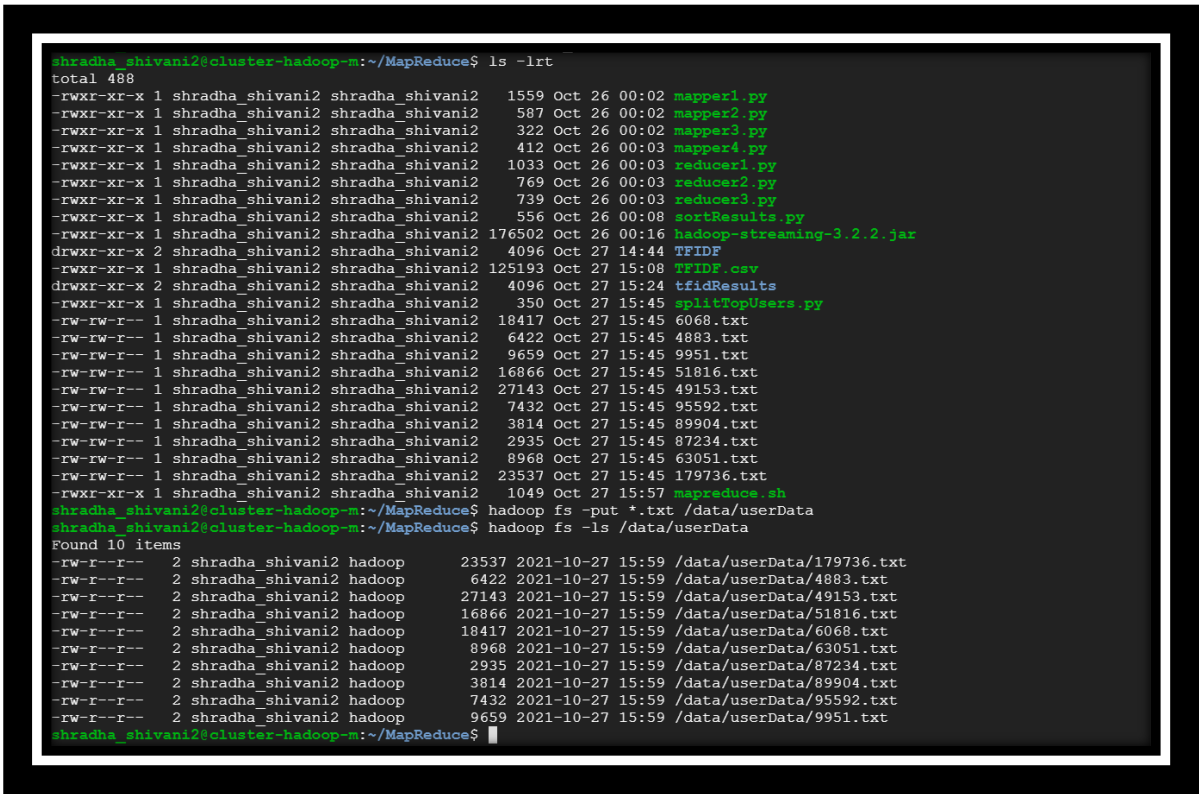
Shradha Shivani (21261104)

TFIDF.csv file gets created under /home/shradha_shivani2/MapReduce

2. The top 10 users listed in TFIDF.csv file are split into individual text files using splitTopUser.py python program. The resultant shows 10 different text files , one for each user.

```
-rwxr-xr-x 1 shradha_shivani2 shradha_shivani2 125193 Oct 27 15:08 TFIDF.csv
drwxr-xr-x 2 shradha_shivani2 shradha_shivani2   4096 Oct 27 15:24 tfidResults
shradha_shivani2@cluster-hadoop-m:~/MapReduce$ vi splitTopUsers.py
shradha_shivani2@cluster-hadoop-m:~/MapReduce$ vi
vi          view        viewres     vigpg       vigr        vim         vim.basic   vim.tiny    vimdiff     vimtutor    vipw        virtualenv  visudo
shradha_shivani2@cluster-hadoop-m:~/MapReduce$ vi splitTopUsers.py
shradha_shivani2@cluster-hadoop-m:~/MapReduce$ python3 splitTopUsers.py
shradha_shivani2@cluster-hadoop-m:~/MapReduce$ ls -lrt
total 492
-rwxr-xr-x 1 shradha_shivani2 shradha_shivani2   1559 Oct 26 00:02 mapper1.py
-rwxr-xr-x 1 shradha_shivani2 shradha_shivani2    587 Oct 26 00:02 mapper2.py
-rwxr-xr-x 1 shradha_shivani2 shradha_shivani2    322 Oct 26 00:02 mapper3.py
-rwxr-xr-x 1 shradha_shivani2 shradha_shivani2    412 Oct 26 00:03 mapper4.py
-rwxr-xr-x 1 shradha_shivani2 shradha_shivani2   1033 Oct 26 00:03 reducer1.py
-rwxr-xr-x 1 shradha_shivani2 shradha_shivani2    769 Oct 26 00:03 reducer2.py
-rwxr-xr-x 1 shradha_shivani2 shradha_shivani2    739 Oct 26 00:03 reducer3.py
-rwxr-xr-x 1 shradha_shivani2 shradha_shivani2    556 Oct 26 00:08 sortResults.py
-rwxr-xr-x 1 shradha_shivani2 shradha_shivani2 176502 Oct 26 00:16 hadoop-streaming-3.2.2.jar
-rwxr-xr-x 1 shradha_shivani2 shradha_shivani2    344 Oct 26 17:16 TFIDF_Result.txt
-rwxr-xr-x 1 shradha_shivani2 shradha_shivani2   1090 Oct 27 14:02 mapreduce.sh
drwxr-xr-x 2 shradha_shivani2 shradha_shivani2   4096 Oct 27 14:44 TFIDF
-rwxr-xr-x 1 shradha_shivani2 shradha_shivani2 125193 Oct 27 15:08 TFIDF.csv
drwxr-xr-x 2 shradha_shivani2 shradha_shivani2   4096 Oct 27 15:24 tfidResults
-rwxr-xr-x 1 shradha_shivani2 shradha_shivani2    350 Oct 27 15:45 splitTopUsers.py
-rw-rw-r-- 1 shradha_shivani2 shradha_shivani2  18417 Oct 27 15:45 6068.txt
-rw-rw-r-- 1 shradha_shivani2 shradha_shivani2   6422 Oct 27 15:45 4883.txt
-rw-rw-r-- 1 shradha_shivani2 shradha_shivani2   9659 Oct 27 15:45 9951.txt
-rw-rw-r-- 1 shradha_shivani2 shradha_shivani2  16866 Oct 27 15:45 51816.txt
-rw-rw-r-- 1 shradha_shivani2 shradha_shivani2  27143 Oct 27 15:45 49153.txt
-rw-rw-r-- 1 shradha_shivani2 shradha_shivani2   7432 Oct 27 15:45 95592.txt
-rw-rw-r-- 1 shradha_shivani2 shradha_shivani2   3814 Oct 27 15:45 89904.txt
-rw-rw-r-- 1 shradha_shivani2 shradha_shivani2   2935 Oct 27 15:45 87234.txt
-rw-rw-r-- 1 shradha_shivani2 shradha_shivani2   8968 Oct 27 15:45 63051.txt
-rw-rw-r-- 1 shradha_shivani2 shradha_shivani2  23537 Oct 27 15:45 179736.txt
shradha_shivani2@cluster-hadoop-m:~/MapReduce$
```

3. All the 10 text files are placed in HDFS under /data/userData



4. In this section, TFIDF is implemented in Hadoop using Python scripts. Altogether there are 4 mapper and 3 reducer python program files. The implementation takes place in four phases. The first phase uses three mappers and three reducers. The last phase uses the fourth mapper to generate a single file with 10 users word list and its TF-IDF value. The output of one phase is fed as an input for the next phase. Each text file is given as an input to mapreduce.sh script one by one. For each of the 10 text files, the algorithm is run and the resultant text file for each 10 text files gets merged under /home/shradha_shivani2/MapReduce/tfidfResults/

The mapreducer.sh script consists of below commands:

hadoop jar hadoop-streaming-3.2.2.jar -files /home/shradha_shivani2/MapReduce/mapper1.py,/home/shradha_shivani2/MapReduce/reducer1.py -mapper 'python3 mapper1.py' -reducer 'python3 reducer1.py' -input hdfs:///data/userData/$1 -output hdfs:///data/output1

hadoop jar hadoop-streaming-3.2.2.jar -files /home/shradha_shivani2/MapReduce/mapper2.py,/home/shradha_shivani2/MapReduce/reducer2.py -mapper 'python3 mapper2.py' -reducer 'python3 reducer2.py' -input hdfs:///data/output1/ -output hdfs:///data/output2

hadoop jar hadoop-streaming-3.2.2.jar -files /home/shradha_shivani2/MapReduce/mapper3.py,/home/shradha_shivani2/MapReduce/reducer3.py -mapper 'python3 mapper3.py' -reducer 'python3 reducer3.py' -input hdfs:///data/output2/ -output hdfs:///data/output3

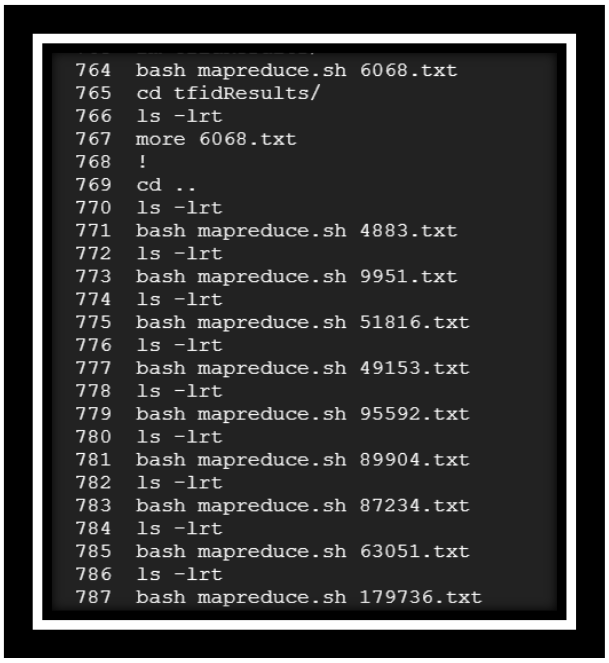
hadoop jar hadoop-streaming-3.2.2.jar -files /home/shradha_shivani2/MapReduce/mapper4.py -mapper 'python3 mapper4.py' -input hdfs:///data/output3/ -output hdfs:///data/output4

hadoop fs -getmerge hdfs:///data/output4/ /home/shradha_shivani2/MapReduce/tfidResults/$1

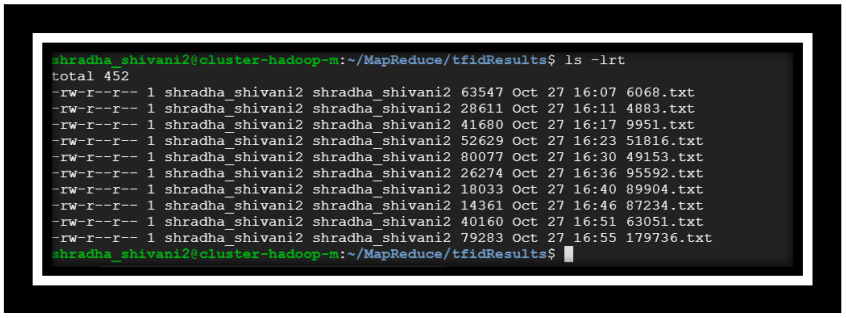
hadoop fs -rm -r hdfs://data/output*

Shradha Shivani (21261104)

5. The history command shows that individual mapper reducer algorithms



6. Running mapreduce.sh script for each text file gives the resultant text files in tfidResults directory under /home/shradha_shivani2/Mapreduce



7. Next, sorting program sortResults.py is run to get the expected result. The resultant is captured in TFIDF_Results_GCP.txt and stored in /home/shradha_shivani2/MapReduce