Title	Assignment 01 - Data Analysis
Name	Siddarth Shantinath Patil
Student ID Number	20210993
E-mail ID	siddarth.patil2@mail.dcu.ie
Programme of Study	MSc in Computing with Data Analytics Major
Module Code	CA675
Date of Submission	22-Nov-2020
GitHub Link	siddarth-patil/Coud-Tech-Assginment-01-Data-Analysis: DCU Masters | CA675 | Cloud Technologies (github.com)

INTRODUCTION:

The objective of this assignment it to perform data analysis on data which can be considered big data (200,000 records) and try to find the answers to the tasks given. Using the data obtained from StackExchange, I loaded, transformed and stored the data using PIG, further to find solutions to tasks I made use of HIVE and finally the last task was to find the TF-IDF of the data obtained in one of the tasks from HIVE and to perform this I made use of HIVEMALL.

The dataset consists of twenty-three columns in total which was further reduced to eight columns after carefully comparing it with our requirements. This was done using PIG queries. On the stored data, HIVE queries were used to answer three questions: finding top 10 posts by scores, finding top 10 users by post scores and finding number of distinct users who used the word “Hadoop” in their posts. Finally, HIVEMALL was used to find TF-IDF on the data which contained all the posts made by top 10 users who were found in the second task of HIVE queries. All the codes with output screenshots and explanations can be found at the GitHub Repository

DATA ACQUISITION:

The objective is to collect the top 200,000 records by score from StackExchange. StackExchange has a limit of 50,000 records per query. And therefore, to collect the data required for the assignment I made use of 5 queries. The code of the quires are as follows:

For the top 44918 posts:
```
select top 50000 * from posts where posts.ViewCount > 121000 ORDER BY posts.ViewCount
```

For the next 47817 posts:
```
select top 50000 * from posts where posts.ViewCount <= 121000
AND posts.ViewCount >70000 ORDER BY posts.ViewCount
```

For the next 47066 posts:
```
select top 50000 * from posts where posts.ViewCount <= 70000
AND posts.ViewCount > 50000 ORDER BY posts.ViewCount
```

For the next 41595 posts:
```
select top 50000 * from posts where posts.ViewCount <= 50000
AND posts.ViewCount > 40000 ORDER BY posts.ViewCount
```

For the next 18604 posts:
```
select top 18604 * from posts where posts.ViewCount <= 40000
AND posts.ViewCount > 36000 ORDER BY posts.ViewCount
```

I arrived to the first number 121000 by hit and trail. The logic was to fetch a little less than 50,000 records to make sure that no data was missed. And I continued to fetch the data in similar way till I reached 200,000 records.

DATA TRANSFORMATION USING PIG:

Initially, the data acquired was loaded to local machine in GCP using the GUI and later was copied to HSFS. As there are 5 different csv files, the data was loaded in five different relations and then were joined using a UNION function. Later this joined data was transformed using the PIG queries. To transform, first the records where the OwnerUserID or the Id was null were removed and later out of 23 columns only 8 were kept and rest were omitted. Finally, this transformed data was stored in HDFS. The bellow screenshots provide the code and the output. The below codes and their explanation with references and screenshots can be found in the GitHub repository

```
Input(s):
Successfully read 47817 records from: "hdfs://cluster-cb25-m/input_dir/2.csv"
Successfully read 44918 records from: "hdfs://cluster-cb25-m/input_dir/1.csv"
Successfully read 41595 records from: "hdfs://cluster-cb25-m/input_dir/4.csv"
Successfully read 47066 records from: "hdfs://cluster-cb25-m/input_dir/3.csv"
Successfully read 22742 records from: "hdfs://cluster-cb25-m/input_dir/5.csv"

Output(s):
Successfully stored 194862 records (210094375 bytes) in: "hdfs://cluster-cb25-m/pig_table"

Counters:
Total records written : 194862
Total bytes written : 210094375
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
```

```
siddarth_patil2@cluster-cb25-m:~$ hadoop fs -ls /pig_table/
Found 6 items
-rw-r--r--   2 siddarth_patil2 hadoop          0 2020-11-22 06:18 /pig_table/_SUCCESS
-rw-r--r--   2 siddarth_patil2 hadoop   51610448 2020-11-22 06:18 /pig_table/part-m-00000
-rw-r--r--   2 siddarth_patil2 hadoop   49354287 2020-11-22 06:18 /pig_table/part-m-00001
-rw-r--r--   2 siddarth_patil2 hadoop   47979912 2020-11-22 06:18 /pig_table/part-m-00002
-rw-r--r--   2 siddarth_patil2 hadoop   39531963 2020-11-22 06:17 /pig_table/part-m-00003
-rw-r--r--   2 siddarth_patil2 hadoop   21617765 2020-11-22 06:17 /pig_table/part-m-00004
```

HIVE LAODING AND QUERY:

The data which was transformed using PIG in the above step was used and loaded here. Firstly, a Database was created and further a new table inside it. Finally, the data was loaded in the created table. The code with output for these steps can be seen below:

```
hive> set hive.cli.print.header=true;
hive> CREATE DATABASE part_three;
OK
Time taken: 0.196 seconds
hive> create table part3_posts
    > (Title string, Body string, Tags string, Score int, Id int,  ViewCount int, OwnerUserId int, OwnerDisplayName
string)
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
OK
Time taken: 0.874 seconds
hive> LOAD DATA INPATH 'hdfs://cluster-cb25-m/copied/' INTO TABLE posts;
Loading data to table default.posts
OK
Time taken: 1.042 seconds
```

Further on this table three quires were run for each of the three tasks:

1. Finding top 10 posts by scores.

```
hive> SELECT Id, Score, Title FROM part3_posts ORDER BY Score DESC LIMIT 10;
Query ID = siddarth_patil2_20201122064008_51147a9f-cd97-4643-a4a0-d64c44d17c9a
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1606025527048_0006)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      5          5        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 22.79 s
--------------------------------------------------------------------------------
OK
id      score   title
11227809        24969   (Why is processing a sorted array faster than processing an unsorted array?
927358  21777   (How do I undo the most recent local commits in Git?
2003505 17395   (How do I delete a Git branch locally and remotely?
292357  12200   (What is the difference between 'git pull' and 'git fetch'?
231767  10627   (What does the "yield" keyword do?
477816  10467   (What is the correct JSON content type?
348170  9309    (How do I undo 'git add' before commit?
1642028 9174    (What is the "-->" operator in C++?
6591213 8919    (How do I rename a local Git branch?
5767325 8762    (How can I remove a specific item from an array?
Time taken: 123.123 seconds, Fetched: 10 row(s)
```

2. Finding top 10 users by post scores.

```
hive> SELECT OwnerUserId, SUM(Score) AS TotalScore FROM part3_posts GROUP BY OwnerUserId ORDER BY TotalScore DESC LI
MIT 10;
Query ID = siddarth_patil2_20201122064552_64ca2b3d-6ab8-46c3-a976-c5b82164cd13
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1606025527048_0006)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      5          5        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 26.49 s
--------------------------------------------------------------------------------
OK
owneruserid     totalscore
87234   36212
4883    26934
6068    24575
9951    23631
89904   22426
51816   21257
49153   18200
95592   17793
63051   16907
39677   15048
Time taken: 28.19 seconds, Fetched: 10 row(s)
```

3. Finding number of distinct users who used the word “Hadoop” in their posts

```
hive> SELECT COUNT(DISTINCT OwnerUserId) AS Count_User
    > FROM part3_posts
    > WHERE (UPPER(Body) LIKE '%HADOOP%' OR UPPER(Title) LIKE '%HADOOP%' OR UPPER(Tags) LIKE '%HADOOP%');
Query ID = siddarth_patil2_20201122064717_9211ef88-c5bd-47d8-acdb-c803cc250437
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1606025527048_0006)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      5          5        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 30.95 s
--------------------------------------------------------------------------------
OK
count_user
346
Time taken: 32.219 seconds, Fetched: 1 row(s)
```

CALCULATING TF-IDF USING HIVEMALL:

The data for TF-IDF consists of all the posts made by the top 10 users as calculated in the above steps. For this, the **OwnerUserId** of those 10 users were made use of with JOIN function on the table **part3_posts** to get and store only the posts made by top 10 users. As the created table consists of all the eight fields it was cleaned and the final table consisted of only 2 columns: one was the OenerUserID

and the other was the concatenation of **Body, Title** and **Tags** columns. Finally, dependencies of HIVEMALL were installed in order to use its functionalities. The screenshot of the code and output can be seen below.

```
hive> create temporary macro max2(x INT, y INT) if(x>y,x,y);
OK
Time taken: 0.283 seconds
hive> create temporary macro tfidf(tf FLOAT, df_t INT, n_docs INT) tf * (log(10, CAST(n_docs as FLOAT)/max2(1,df_t))
 + 1.0);
OK
Time taken: 0.145 seconds
```

```
hive> create or replace view exploded as select ownerUserId, word from final_table LATERAL VIEW explode(tokenize(Pos
ts, True)) t as word where not is_stopword(word);
OK
Time taken: 0.627 seconds
hive> create or replace view term_frequency as select ownerUserid, word, freq from (select ownerUserId, tf(word) as
word2freq from exploded group by ownerUserId) t LATERAL VIEW explode(word2freq) t2 as word, freq;
OK
Time taken: 0.394 seconds
```

```
hive> create or replace view document_frequency as select word, count(distinct ownerUserId) docs from exploded group
 by word;
OK
Time taken: 0.284 seconds
```

```
hive> select count(ownerUserId) from final_table;
Query ID = siddarth_patil2_20201122124413_f1fe4aef-1c3b-4214-838c-5fc1fcda920f
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1606048296103_0004)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 7.22 s
--------------------------------------------------------------------------------
OK
0
Time taken: 14.725 seconds, Fetched: 1 row(s)
```

```
hive> set hivevar:n_docs=10;
hive> create or replace view tfidf as select tf.ownerUserId,  tf.word, tfidf(tf.freq, df.docs, ${n_docs}) as tfidf f
rom  term_frequency tf  JOIN document_frequency df ON (tf.word = df.word) order by tfidf desc;
OK
Time taken: 0.497 seconds
```

```
89904   dump    0.0010600365931168199
89904   got 0.0010600365931168199
89904   lines   0.0010600365931168199
89904   solution    0.0010600365931168199
89904   replace 0.0010600365931168199
89904   #   0.0010600365931168199
89904   copy    0.0010600365931168199
89904   end 0.0010600365931168199
89904   collection  0.0010600365931168199
49153   oop 0.0010600365931168199
49153   given   0.0010600365931168199
49153   value=  0.0010600365931168199
49153   12  0.0010600365931168199
49153   void    0.0010600365931168199
49153   b   0.0010600365931168199
49153   put 0.0010600365931168199
```

REFERENCES:

1. TF-IDF Term Weighting · Hivemall User Manual (apache.org)
2. Stack Overflow - Where Developers Learn, Share, & Build Careers
3. Getting Started · Hivemall User Manual (apache.org)
4. Pig Latin Basics (apache.org)