**CA675: Assignment 1**

This document details the steps taken by student Simranjeet Singh (19210505) to complete tasks of assignment 1 for CA675.

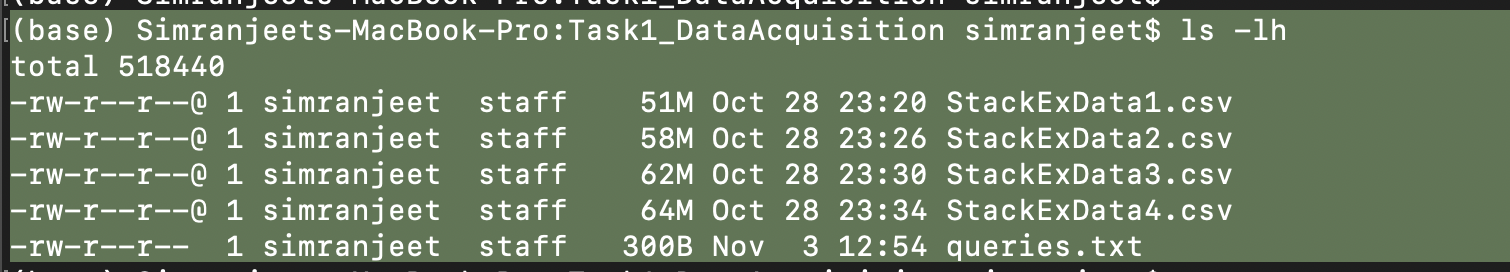
The entire code is available on my git repository and this document contains links of evidences of the result of each task. Most of the screenshots are available in respective task folder as this document was insufficient to hold all screenshots. A zipped file for the code is also submitted with this assignment on Loop.

Git:

**Task 1: Data Acquisition**

Following queries were triggered at stackexchange’s query composer to fetch top 200,000 records by viewcount.

1. select \* from posts where viewcount>96500;
2. select \* from posts where viewcount<96500 and viewcount>56750;
3. select \* from posts where viewcount<56799 and viewcount>40670;
4. select \* from posts where viewcount<40672 and viewcount>31700;



These 4 files are shared at my google drive and can be accessed at following links:

https://drive.google.com/file/d/1ofnZhU4itz3OUbhjmkBFe4jlruyy6ov\_/view?usp=sharing

https://drive.google.com/file/d/1eVwW4\_fVCvcO0wHKyXFFDbv51NpOoW4P/view?usp=sharing

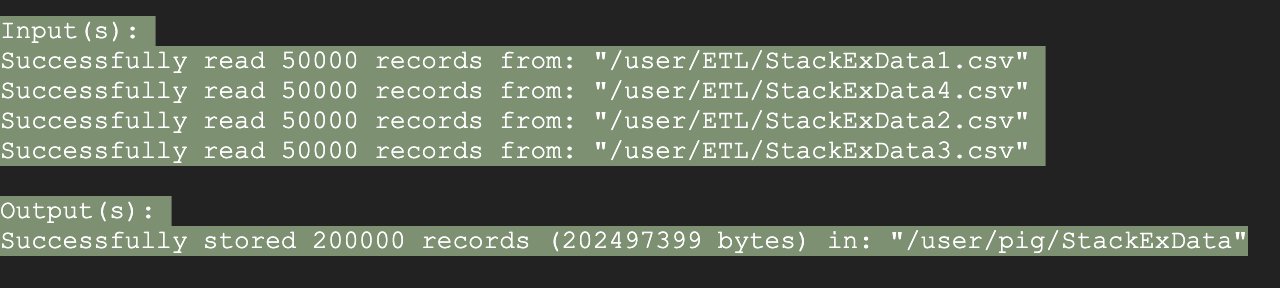
https://drive.google.com/file/d/1CGAwl-3UciP3bdBAMoeL\_EuN0g6vudZn/view?usp=sharing

https://drive.google.com/file/d/175lAHp2uVurJf6FGp3y48th7Hahcr8hJ/view?usp=sharing

**Task 2: ETL in Pig**

The above 4 files were placed at GCP server and then put to hdfs to be used in pig. In my code, a script ‘ETL.sh’ in Task2 folder will take care of downloading the files from google drive to GCP server and then putting them to hdfs path. This will then trigger a .pig script to perform rest of the ETL operations in Pig and storing the final data in hdfs for further analysis.

The output from Pig is taken in **parquet format** since our files contain newline characters in data itself and storing in parquet gives easy and comfortable data manipulation in hive.



Pig Screenshots:

<https://drive.google.com/file/d/11qaosf0lvKusFezQh9yRXQS5kf5zlTJJ/view?usp=sharing>

<https://drive.google.com/file/d/1ZO5GMstwzVeI4F3CEmNKXcJZ5ZeCByE7/view?usp=sharing>

**Task 3: Hive Data Load and Queries**

After receiving data from Pig from previous step, an external table is created in hive with parquet format. And below screenshot shows the creation & count of the table.

Table Creation: <https://drive.google.com/file/d/1gyiLk-ffNE9vkgcYIJxHzTYu5stQsbGb/view?usp=sharing>

Table Count: <https://drive.google.com/file/d/1LNGnGYcjdrmSTET28SbEDbATGxNIyBQC/view?usp=sharing>

Query 1: Top 10 posts by Score

<https://drive.google.com/file/d/14xBaQFP1mRXvGp0rTTmCglMIrYK4opWd/view?usp=sharing>

Query 2: Top 10 users by score

<https://drive.google.com/file/d/1E4HbSC031X-aX4NOGtVP_iJPGPa9OEg7/view?usp=sharing>

Query 3: Count of Users who used term ‘hadoop’ in their posts

<https://drive.google.com/file/d/1gl9XFNaNnI5JGtnpTtYVoZNfx_qULpZK/view?usp=sharing>

**Task 4: TF-IDF Calculation and top 10 terms by each user from query 2 from Task 3**

User Id for the top 10 users was extracted and saved in a file in hdfs path.

<https://drive.google.com/file/d/1jGJAtfq3Ca49ZmzIph0595wwXJVw1U_N/view?usp=sharing>

<https://drive.google.com/file/d/1vFF6Oqb_dCKF66HCDsCP0LjKFwmv1kkW/view?usp=sharing>

A script in Task 4 folder “TFIDF\_TopTerms.sh” will get the user ids from this file and will run the TFIDF map reduce code on each ID one by one to fetch its top 10 used terms in their posts. More detail on each command is given in the script itself. I have made the script to ignore all junk characters and only consider words from posts.

<https://drive.google.com/file/d/1XMymCk8_7N4Sp_YdQArMXcFuYbrpDUsf/view?usp=sharing>

These terms are finally stored in a hive table in relation with its original userid.

**Code execution:**

You can go through entire code on my github or you can download the zipped file from my Google Drive.

A script “codeexec.sh” in the main zip file will take care of unzipping the entire code, downloading required source files & other jars. This will further execute pig script, load hive tables and execute TFIDF MapReduce code. This script will end with displaying top 10 terms used by each of top 10 users.

The complete code execution takes around 20 minutes to finish considering the amount of data and number of posts this code has to process to achieve its final target.