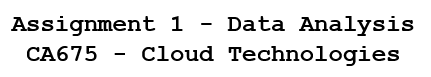
****

|  |  |
| --- | --- |
| **Name** | Rohit Shinde |
| **Student ID** | 22260213 |
| **E-mail** | [rohit.shinde2@mail.dcu.ie](mailto:rohit.shinde2@mail.dcu.ie) |

Git Repository link: <https://gitlab.computing.dcu.ie/shinder2/ca675-cloud-technology-assign1>

Project link on cloud: <https://console.cloud.google.com/dataproc/clusters?region=us-central1&organizationId=999744533918&project=weighty-volt-363313>

Took dataset from Kaggle named Amazon Product Review (Electronics category)

Dataset [Link](https://www.kaggle.com/datasets/naveedhn/amazon-product-review-spam-and-non-spam?select=Electronics)

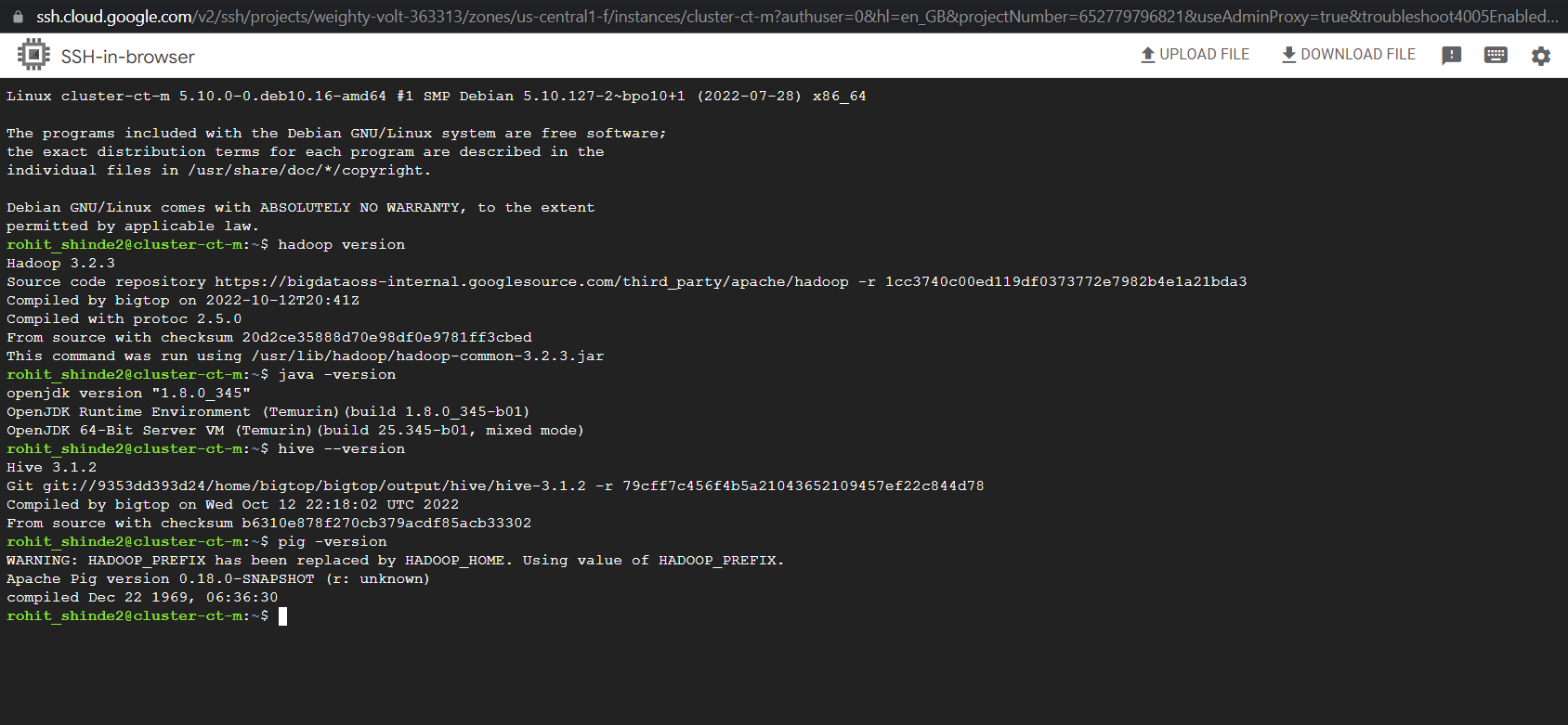
All the queries used in pig, hive, Hadoop are uploaded in gitlab repository [Link](https://gitlab.computing.dcu.ie/shinder2/ca675-cloud-technology-assign1)

****



Used GCP Dataproc to create cluster named “cluster-ct” with 1 master node and 3 slave nodes.









Dataset from Kaggle- [Link](https://www.kaggle.com/datasets/naveedhn/amazon-product-review-spam-and-non-spam?select=Electronics) then reduce that data size using Jupyter notebook python and divided data into chunks as follows:

import pandas as pd

import numpy as np

df=pd.read\_json('Electronics.json', lines=True, nrows=500)

df

cnt=0

for df in pd.read\_json('D:\CA675\Assignment 1\Electronics.json', lines=True, chunksize=200000):

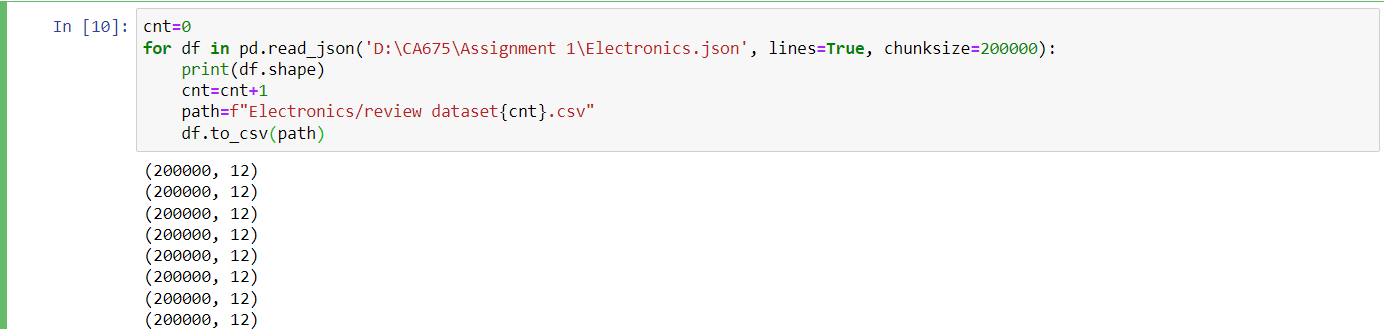
print(df.shape)

cnt=cnt+1

path=f"Electronics/review dataset{cnt}.csv"

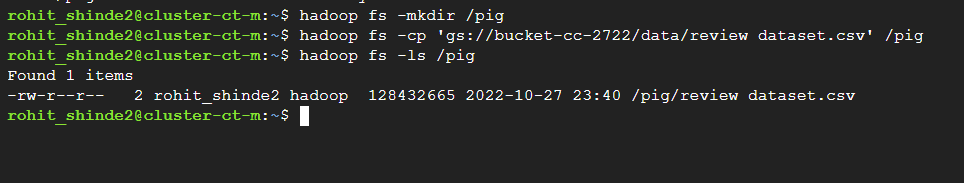
df.to\_csv(path)

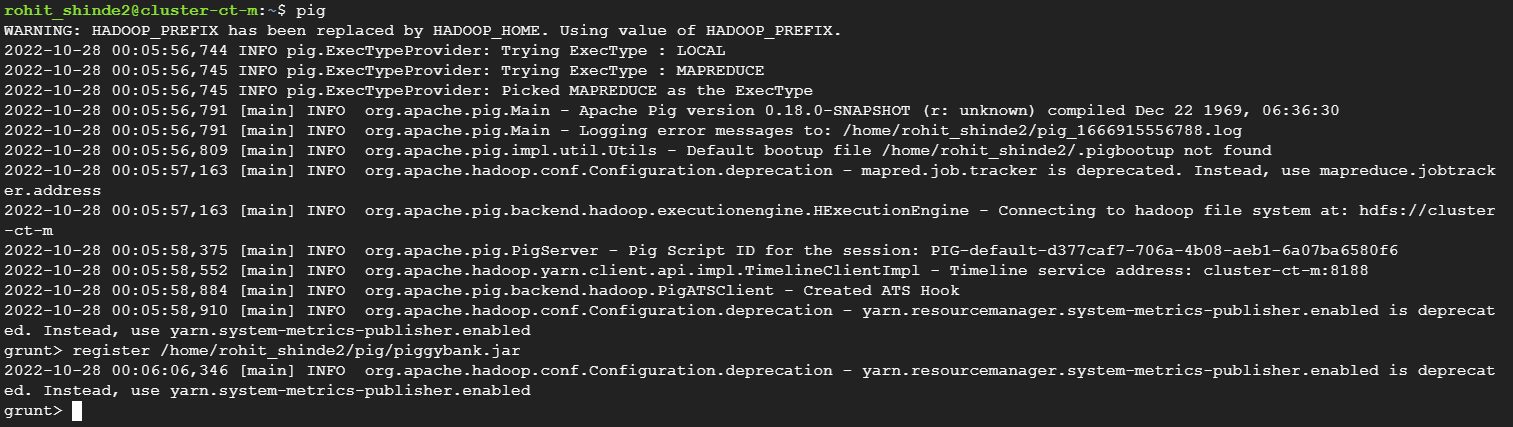
Took file number 6 as random file named “review dataset.csv” and then uploaded it to bucket for further process

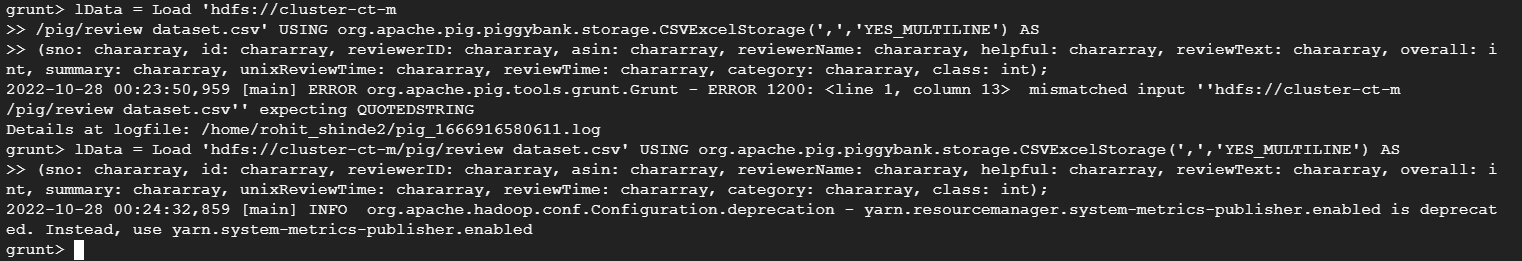




For this task, uploaded the file “review dataset.csv” to bucket and then used Hadoop commands to load the data from bucket to pig directory







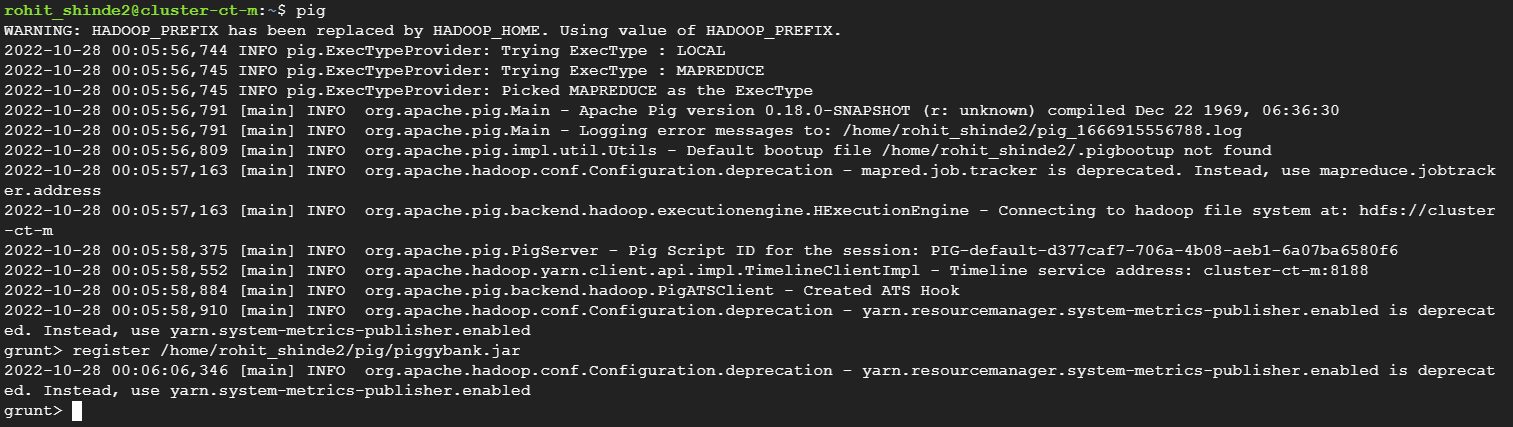


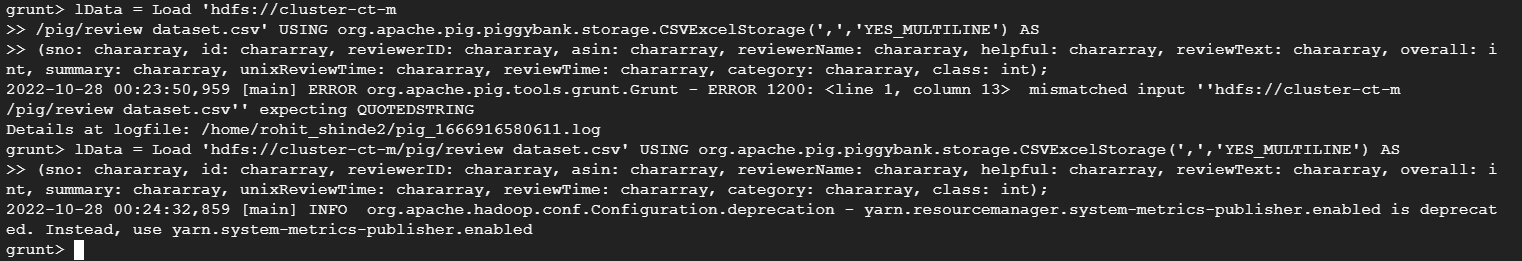
Cleaned the data with the use of pig as it is faster as compared to hive and it uses a multi-query approach. The data had line-break characters and commas so to overcome this, used piggybank library- [Link](https://github.com/escanor797/CA675-Rohit-Shinde/blob/main/piggybank.jar) and registered it in pig.

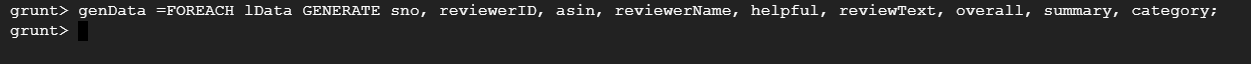
To enable the CSV read multi-line in pig, registered the piggybank.jar file.

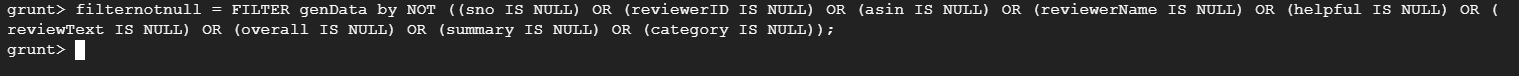
grunt> register /home/rohit\_shinde2/pig/piggybank.jar

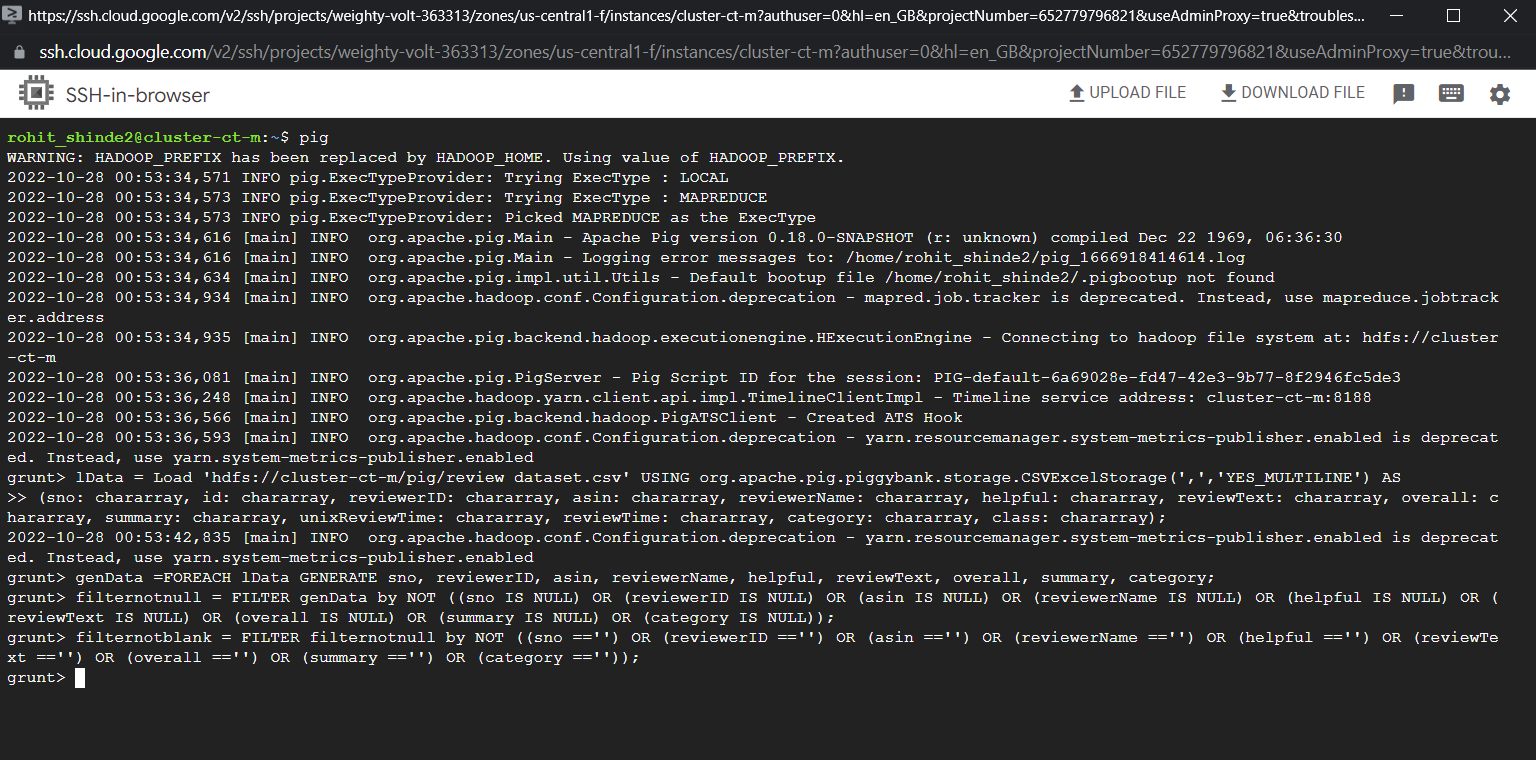
The data is cleaned in pig by checking null values, “N/A” values, blank values, etc. and finally the cleaned data is stored with store query. All the queries used for cleaning data using pig is uploaded on git repository- Link.

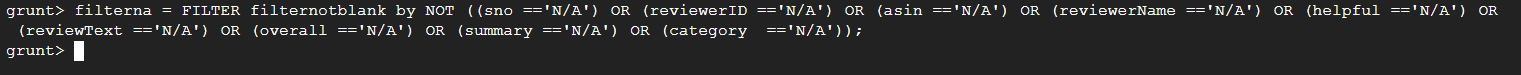






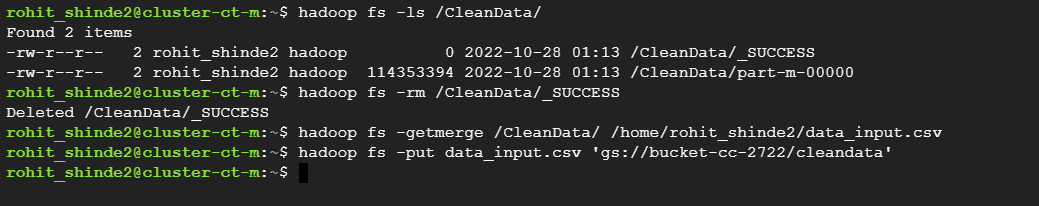




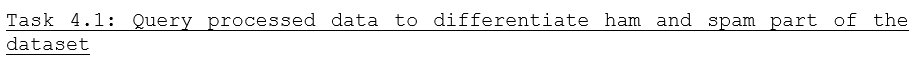




Then put the cleaned data named “data\_input.csv” to local path directory as well as in google bucket.



****

****

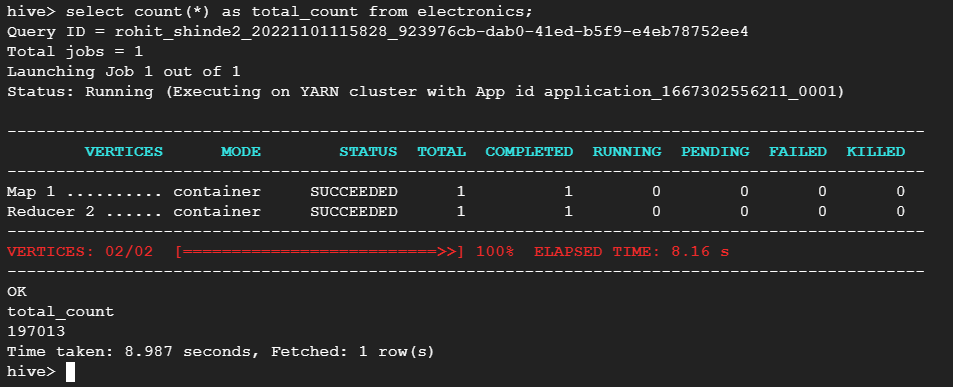
Used hive for this task.

# Created database named db to start with the hive queries:

hive> create database db;

hive> use db;

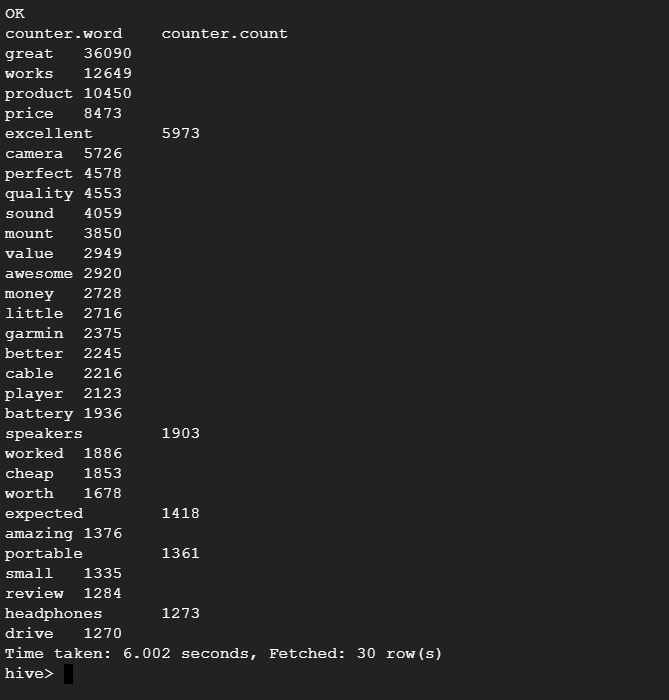
# Created table electronics and loaded the cleaned data from pig:



Downloaded bag of words from [link](https://ortto.com/blog/email-spam-trigger-words/), extracted them with [link](https://toolsyep.com/en/webpage-to-plain-text/) and uploaded it in bucket then

Then created bag of words table named bag2 and loaded the bag of words into it. Also created table named wordcount and counter to count the words in the bag. The queries are uploaded in gitlab repository- [Link](https://gitlab.computing.dcu.ie/shinder2/ca675-cloud-technology-assign1).

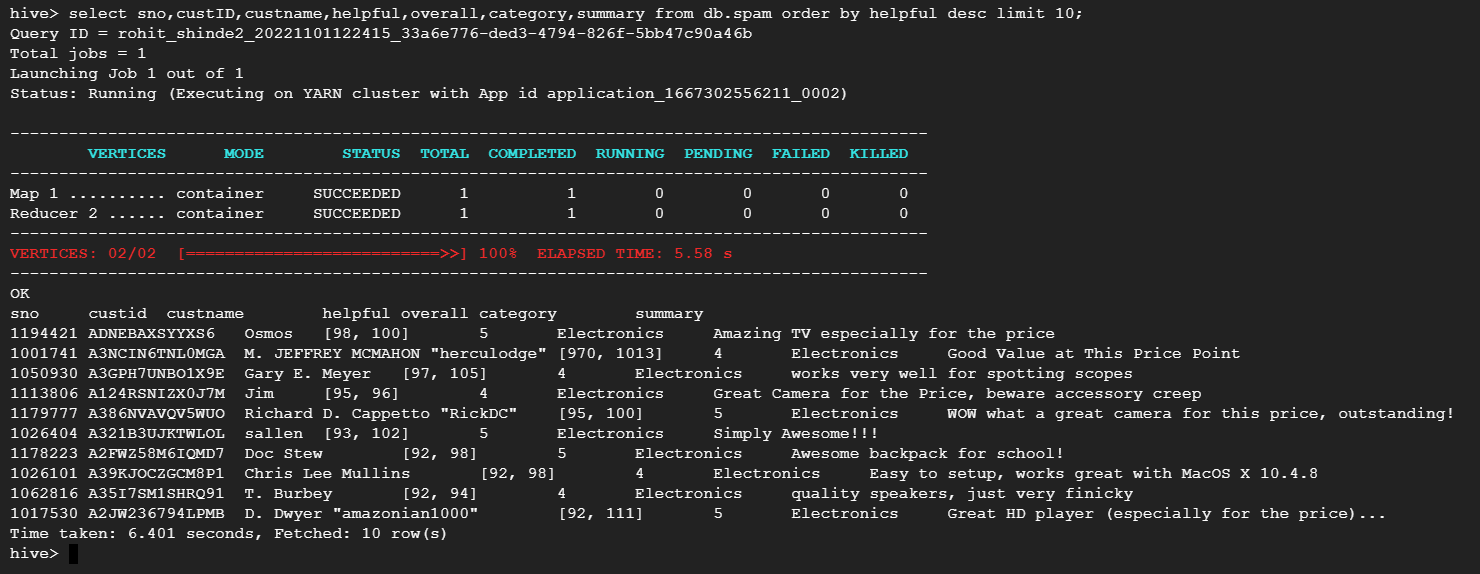
After doing this, the words are selected from bag of words on the basis of their count with respect to summary column from the table electronics.



Created spam and ham table separately using bag of words (Words used from bag of words as spam words are excellent, awesome, speakers, works, product, price). Queries are uploaded on gitlab repository- [Link](https://gitlab.computing.dcu.ie/shinder2/ca675-cloud-technology-assign1).

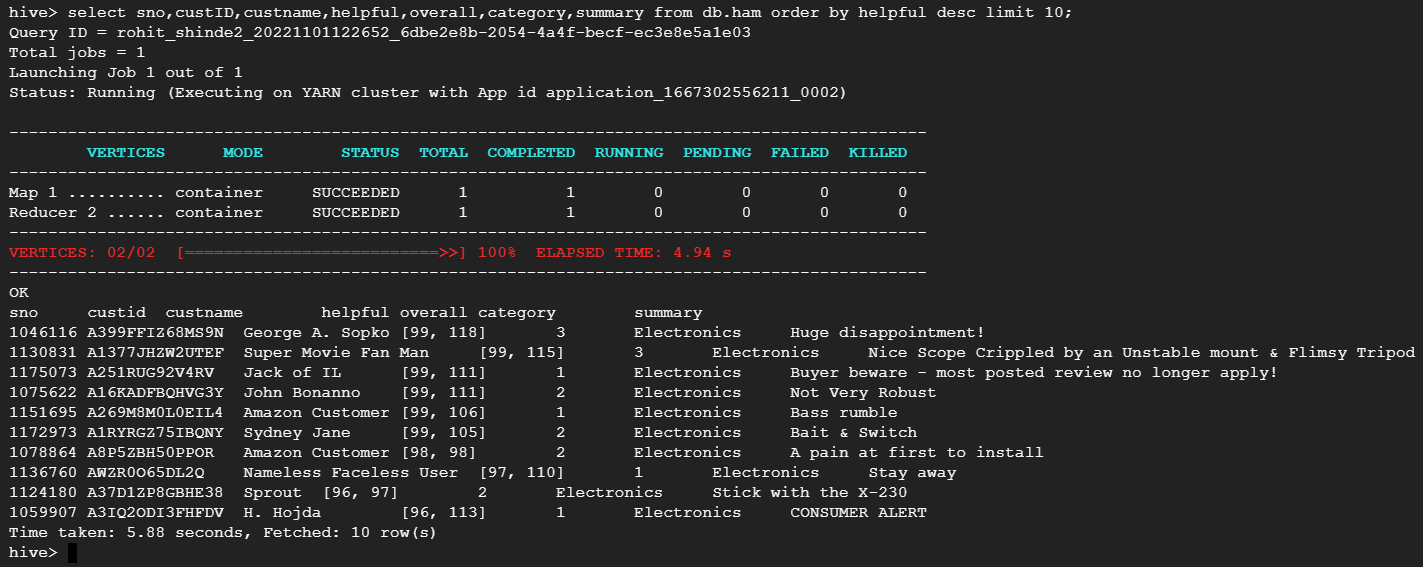


hive> select sno, custID, custname, helpful, overall, category, summary from db.spam order by helpful desc limit 10;





hive> select sno, custID, custname, helpful, overall, category, summary from db.ham order by helpful desc limit 10;



****

****

By using pig cleaned and processed the CSV file which was exported from spam table and this data is fed to mapper and reducer script in python.

--Hive—

In hive, created the table named spamtf and inserted the data into it from table spam.

--Pig—

With the help of pig data is loaded and cleaned again in which null values, blank values and “N/A” values, etc are removed. Final data was stored in local path /pigstore2

Then for mapper reducer the python files are copied from bucket folder pyfiles to local directory /python.

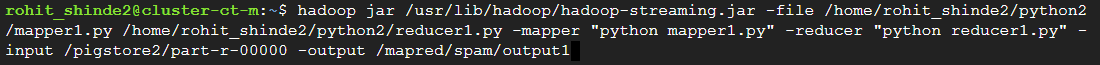
-- for copying file from bucket to hadoop file system

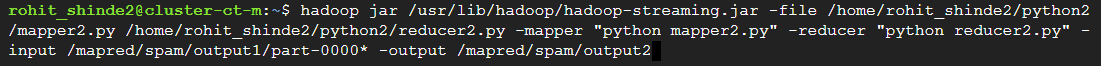
hadoop fs -cp 'gs://bucket-cc-2722/pyfiles/\*' /python

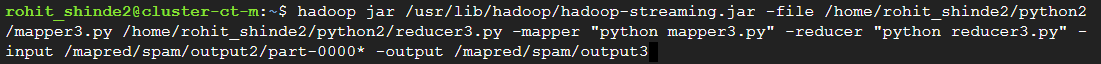
-- for copying file from hadoop file system to local directory

hadoop fs -get /python\* /home/rohit\_shinde2/python2/

Hadoop commands to execute mapper and reducer:

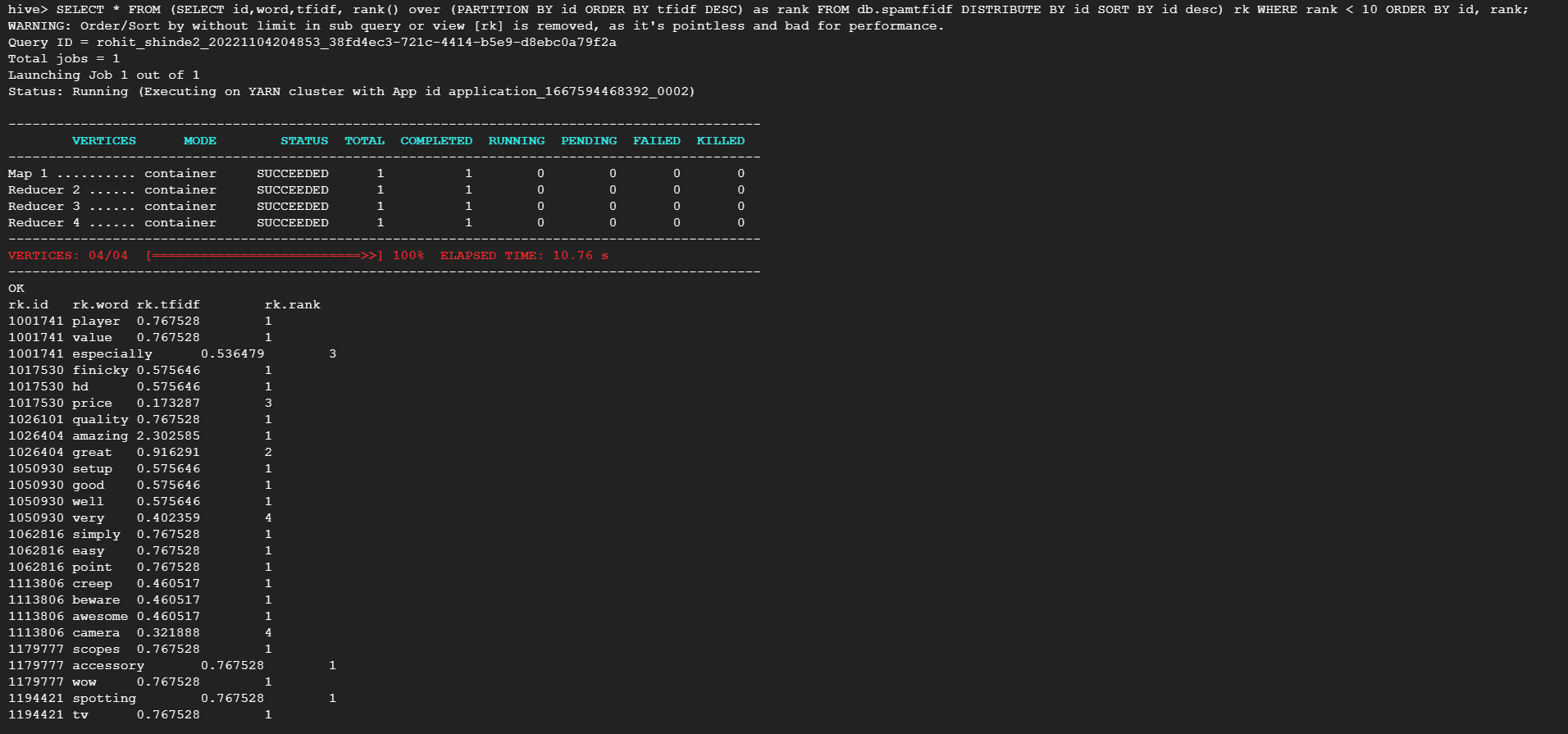








For final outcome, created table named spamtfidf and loaded the final data from mapper and reducer to spamtfidf and extracted top 10 spam accounts with their tf-idf values as below:





The same steps are used to find out top 10 ham records with tf-idf values.

