Computer Sci Electrical Engr 5590 0004
Big Data Programming

Project Increment – 2

Project Title

Stock Market Analysis using Hadoop Ecosystem

Team Members

- Jaisekhar Koya
- Sri Sai Nikhil Kantipudi
- Sai Rohith Guntupally
- Aarthi Nagireddy

Introduction

Study the stock market with graphs, news, important dates, and many more is always hard which leads to inaccuracy in analysis and investment in particular stock. This project helps to analyze the stock market using stock market data and twitter data using Hadoop ecosystem. It collects the twitter data associated with the stock symbols and analyze/predict the stock movement.

Background

There are many articles which tells about the handling of stock market using Hadoop Technologies. But there is no depth analysis of data. Only few articles talked about twitter data and did analysis only for couple of days and not used NLP Techniques. For example: https://www.3pillarglobal.com/insights/analyze-big-data-hadoop-technologies

Goals & Objectives

The main goal of this project is to analyze the stock market with the help of stock market data and also the social data(twitter data in this case) using Hadoop ecosystem. This system collects the twitter data associated with the stock symbols and calculate the sentiment of the stock based on the user tweets.

- ## Motivation

We have tried hard to study the stock market with graphs, news, important dates, and many more, but we failed many times which leads to loss of profits and sometimes even the investments. We always have an extremely hard time analyzing that huge amount of data and coming to a decision based on the analysis. And analyzing that huge data takes a lot of time and got to concentrate on various works too. So, to do this task for us we need some sort of Big data tools to analyze and give us filtered results which we can cross-check and invest in that stock confidently. Not only that as big data tools analyze data extremely fast, we can make decisions quicker than we do. We can also avoid human error using these tools.

- ## Significance
    - As more than half of the population are investing in stocks it is one of the most significant topics to be considered.
    - 30% of investors failing to analyze the stock and predicting the uptrend or downtrend of stock is one of the major reasons.
    - Big data tools help to analyze the huge data which helps to provide the efficient results.
    - It reduces the analysis time of investors and helps in making decisions faster and error free.

- ## Objectives

To create a stock market analysis system, we will be collecting the twitter data using NLP techniques and processing the data. Different tools in Hadoop ecosystem will be used to analyze the data and help in predicting the movement of the stock.

- ## Features

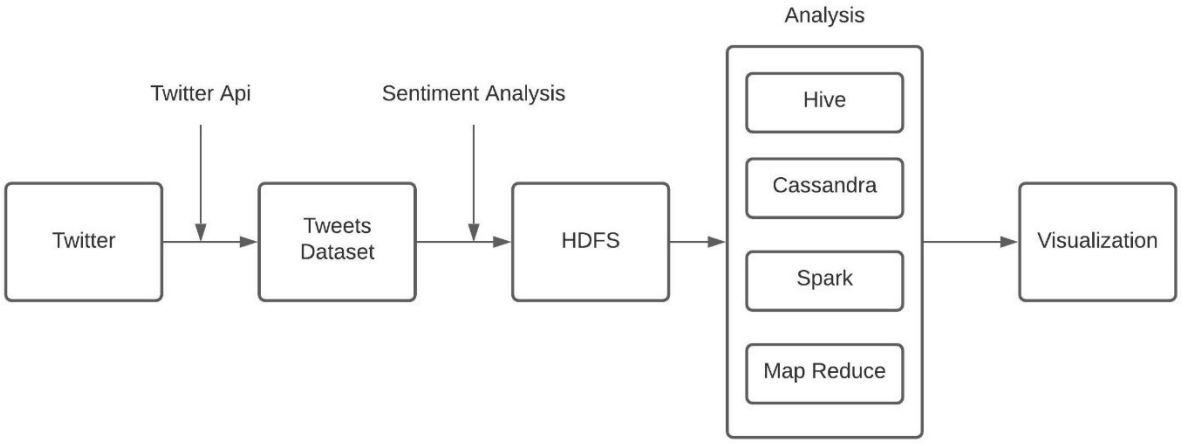
The main features of this project is twitter data extraction, sentimental analysis on the extracted data. Using HDFS to store the data. Apache spark to process the data and Hive for querying the data. A visualization tool for showing out the results.

Dataset:

Twitter Dataset:

https://mailmissouri-my.sharepoint.com/:u:/g/personal/skgyc_umsystem_edu/EXl5i-g3529CmwhdsDDSHoABtW7cOW4U_UtctybYaAsnyQ?e=WGriiy

Design:



Analysis:

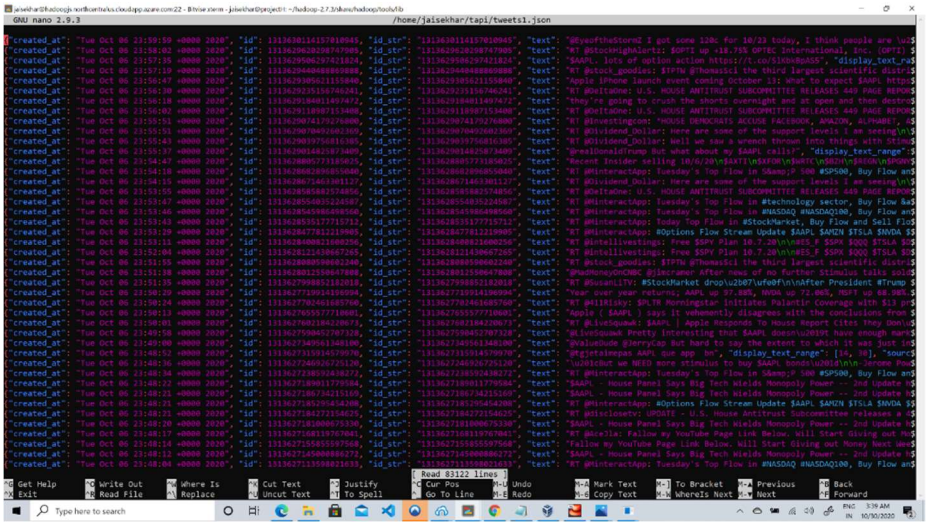
Twitter is one of the best data sources as it contains huge public opinions. Various number of users/ companies tweets the information regarding many things. This dataset is about AAPL stock information. Hence, we extracted all the tweets related to the AAPL stock. Since, our project is to analyze the previous data and predict/analyze the stock movement we are considering the tweets from October 1st to October 28th. We extracted more than 100K tweets and every tweet object contains complete information about the specific tweet like created_at, retweet_count, user details, entity details etc.

Keyword: AAPL

Date Range: 10/01/20 – 10/28/20

No. of tweets: 100k

The dataset will look like below

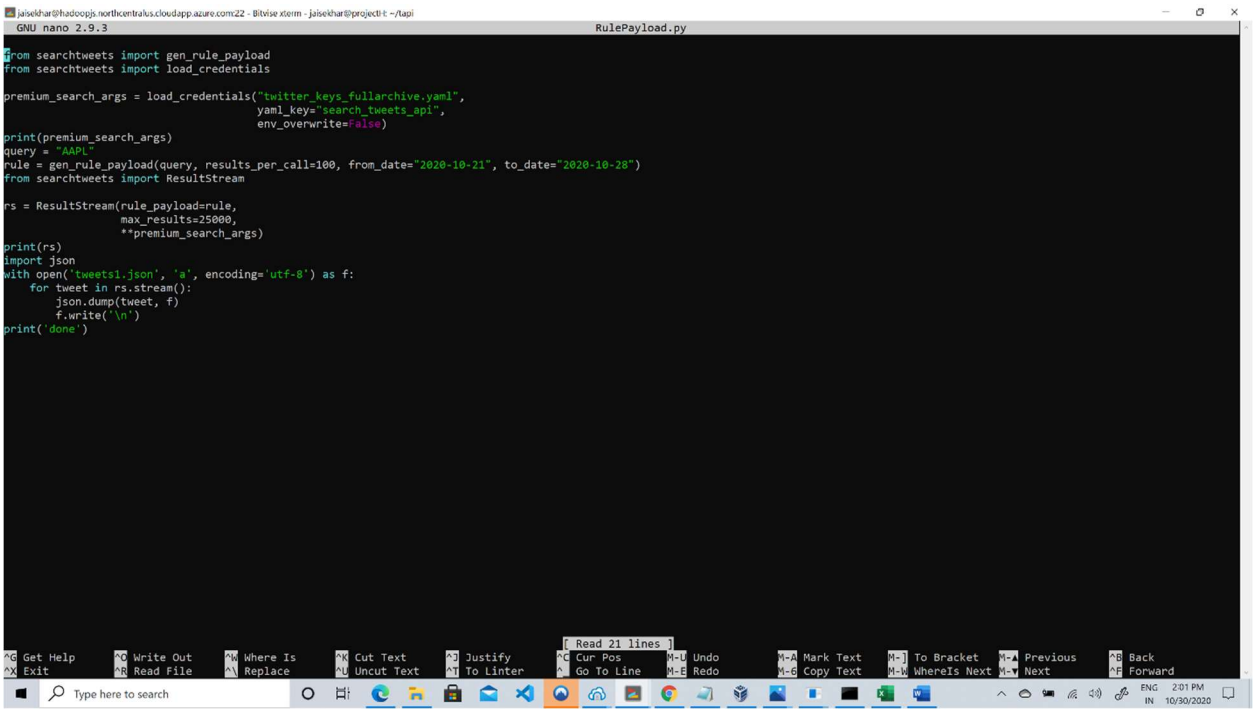


IMPLEMENTATION:

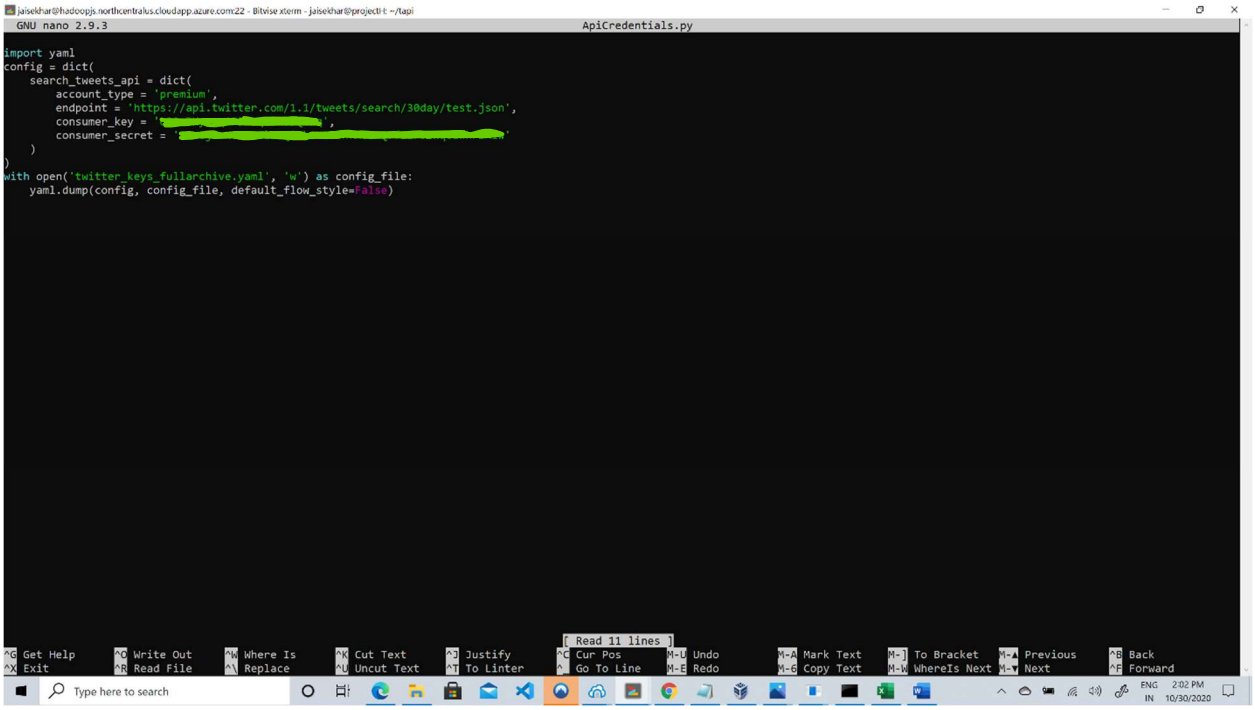
- ## Tweets Extraction:

To extract the tweets we used different API's or library's in the python. Tweepy in the python is the widely used library to extract the tweets but it comes with limitations where we cannot extract tweets based on the data rangle. Even there were other libraries like twint or GetOldTweets3 but those were not working with the recent endpoint update at twitter api. Hence, we used searchtweets library which is nothing but a static library which makes direct API call to the twitter.

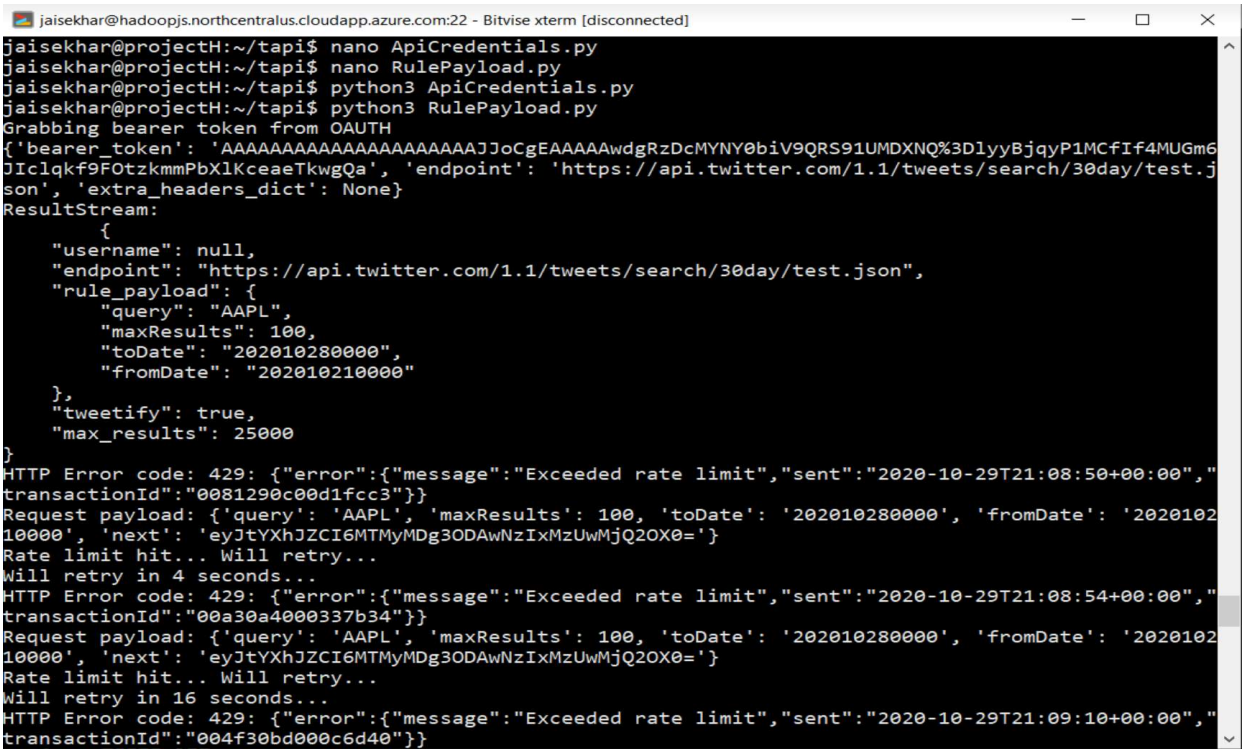
The below code is used to retrieve the tweets using searchtweets library. This code is importing the credentials from the other file and sending query(here we are querying AAPL), results_per_call, from_date, to_date (October month's) parameters to the api with max_results constraint which is set to 25000 here. As we used four keys we are able to retrieve approx 100k tweets by running four times of below code using four different twitter keys of ours.



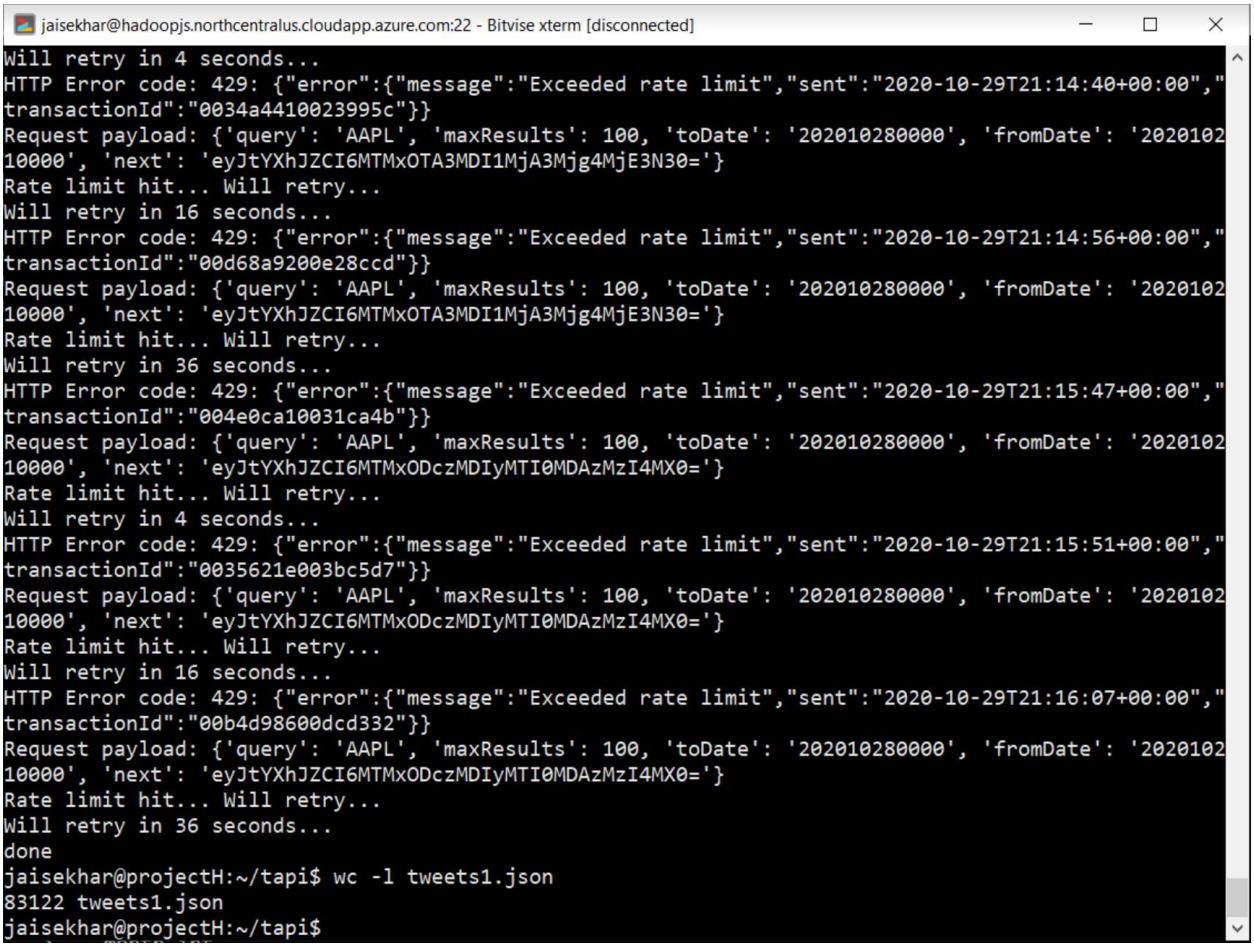
These are the credentials which are used to as the args to the searchtweets api .



Below you can see the execution of all the python files to extract the tweets.



The results are saved in tweets1.json and number lines in the below snippet is nothing but the number of tweets in the json file as the every line indicates one tweet object.

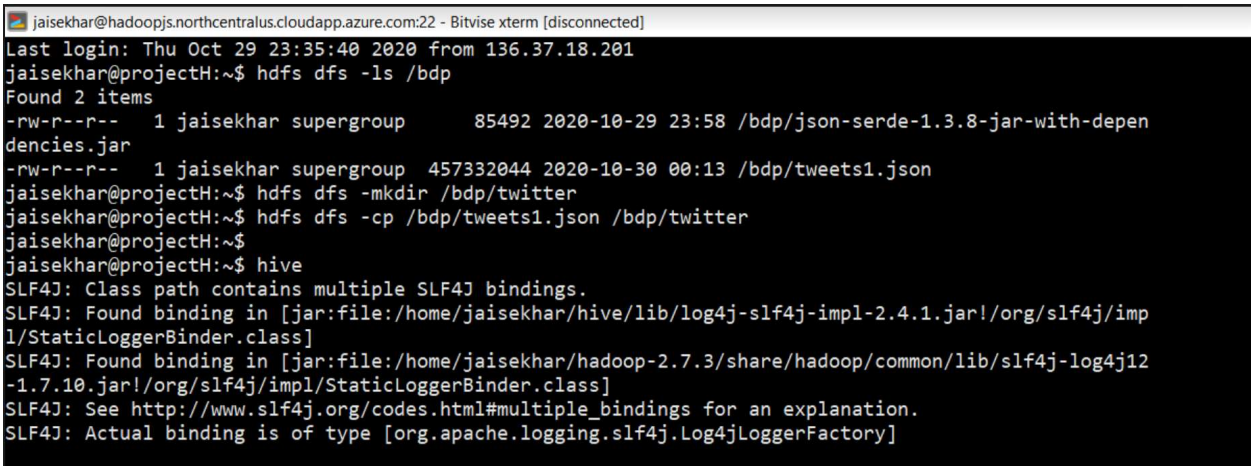


- ## Storing in HDFS:

The tweets file (tweets1.json) is copied to HDFS for further analysis using the the below commad

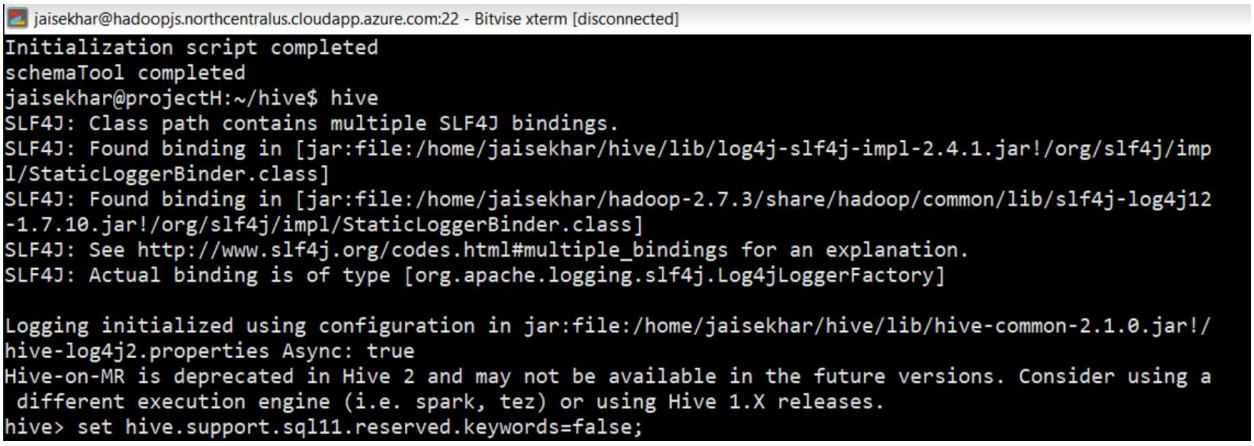
hdfs dfs -copyFromLocal ~/tapi/tweets1.json /bdp

Later, the file is moved to other directory which only contains that particular json file that helps while creating table in hive.

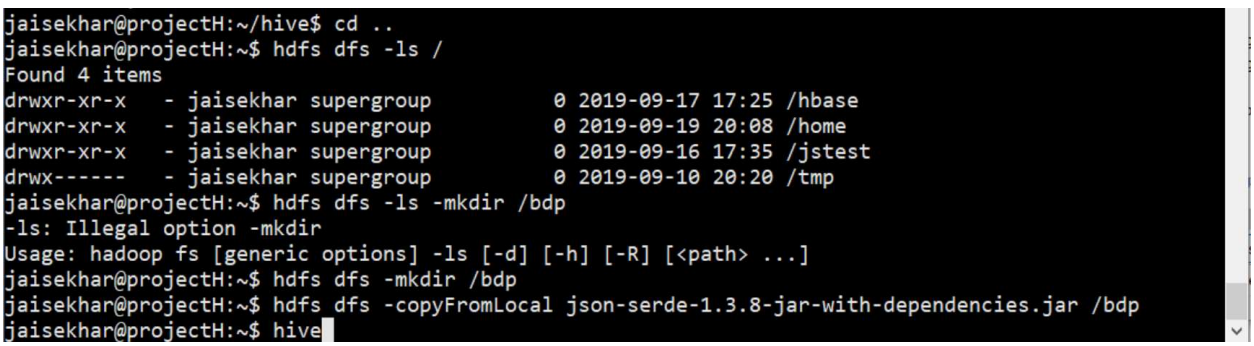


- Hive Operations: (initialization and Loading Data)

To start hive, we ran schematool to initialize the metastore_db and then we started the hive.



Later we added json serde dependency jar to the hive using following command. JSON SerDe is used to map the json object to the table schema. Thus, it helps in mapping the json data to the table.

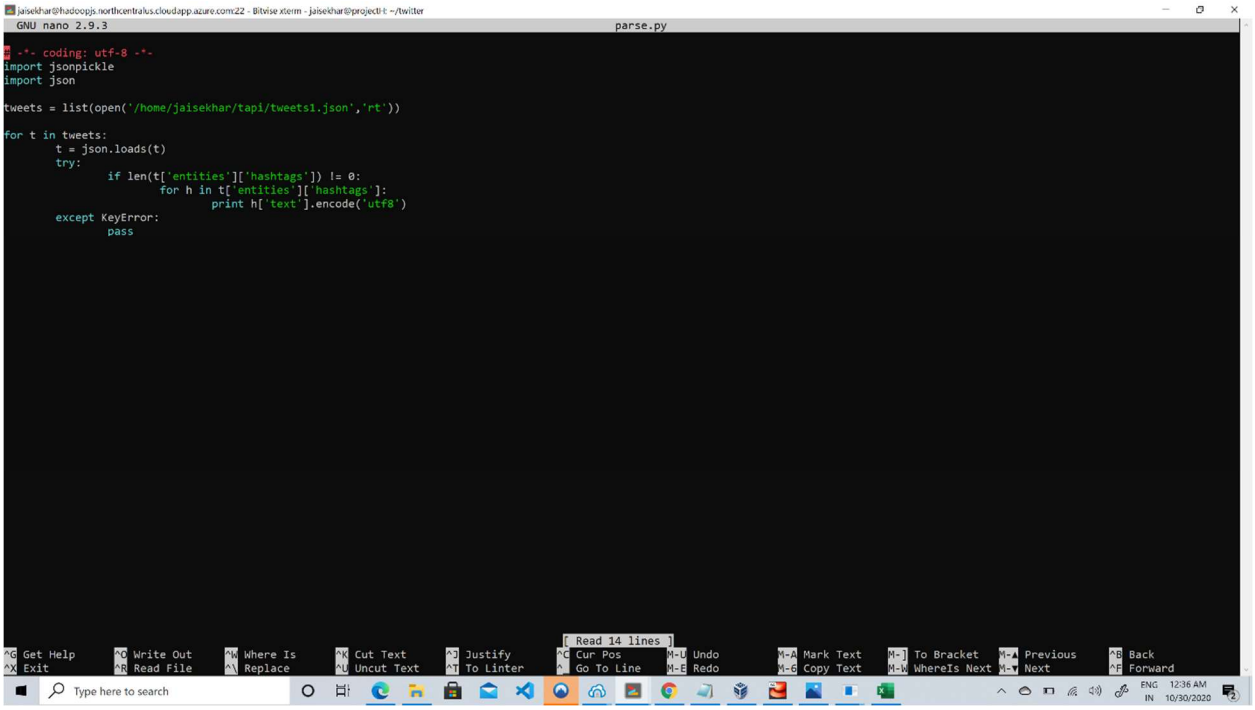


Next, we created the table in hive tweets_raw with the few required attributes from the tweet object which we extracted from twitter api. If you see here, the rows were delimited by serde which points to the added jar of json serde and location is pointing to the directory in hdfs which contains the tweets json file.

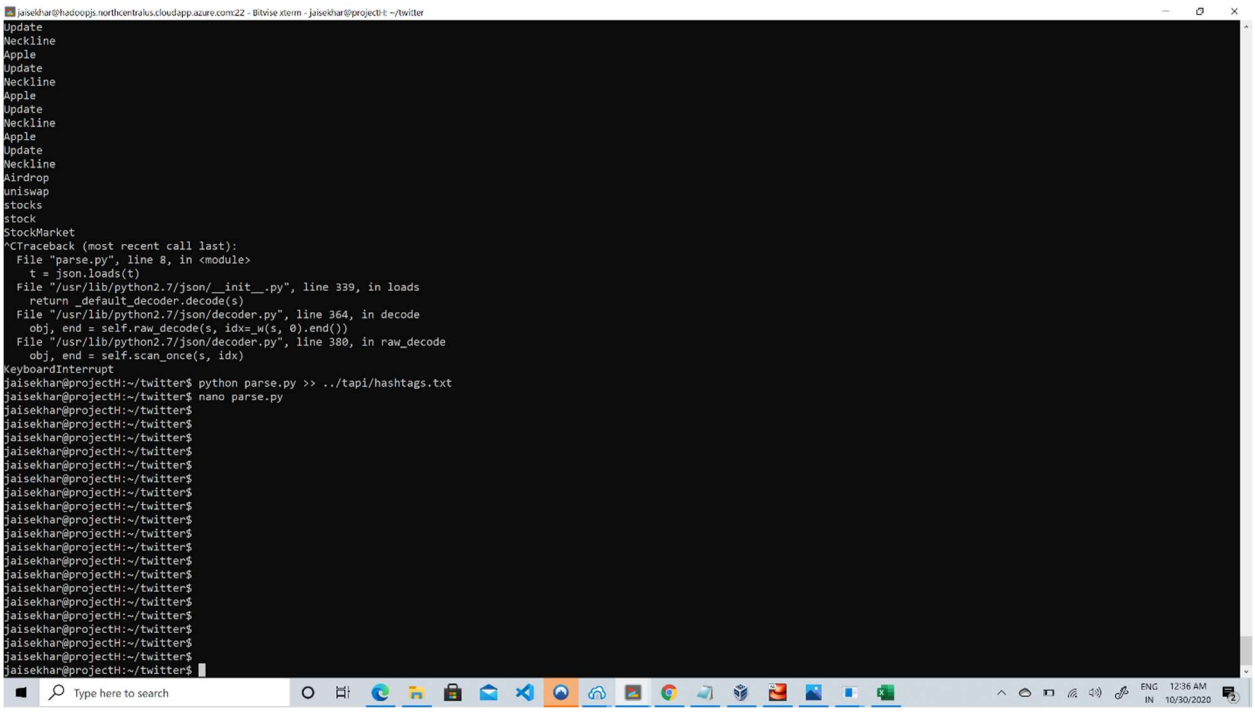
Now, the extracted tweets data is successfully imported to table in hive. Next, we will perform analysis on the data which we extracted into hive and hdfs.

- ## MapReduce – wordcount:

The below python code is used to parse the hashtags from the entities in tweets1.json



The python code is executed and saved the output in hashtags.txt using out operator.

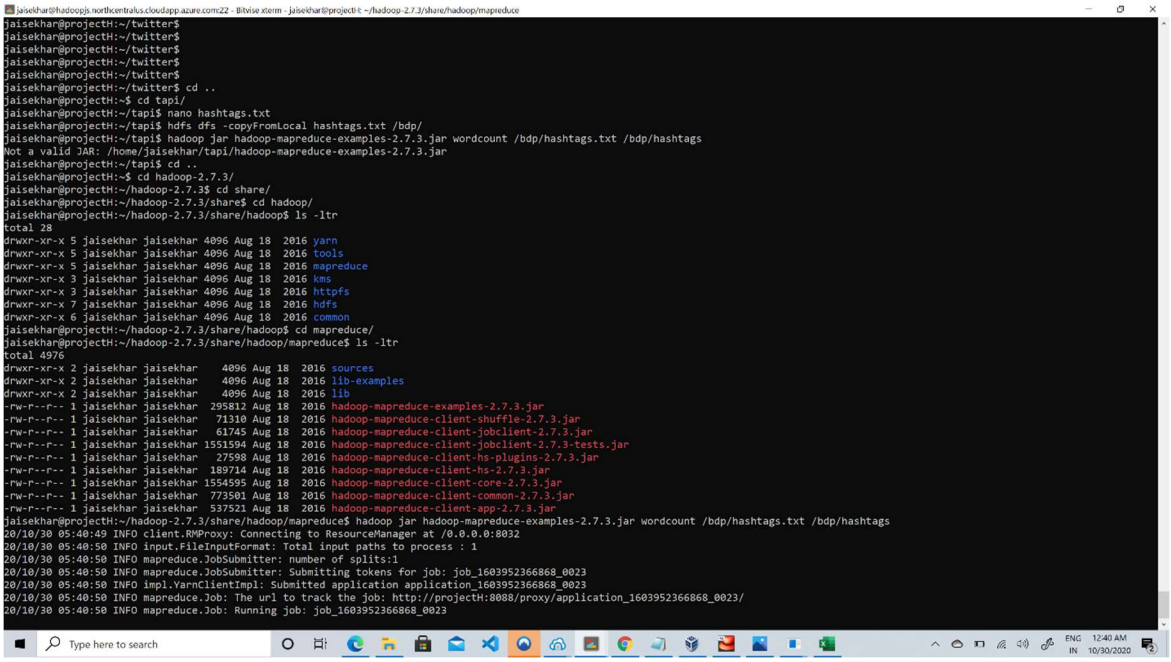


We successfully got the hashtags into the hashtags.txt. And as shown below the hashtags.txt is copied to hdfs using the below command

hdfs dfs -copyFromLocal hashtags.txt /bdp

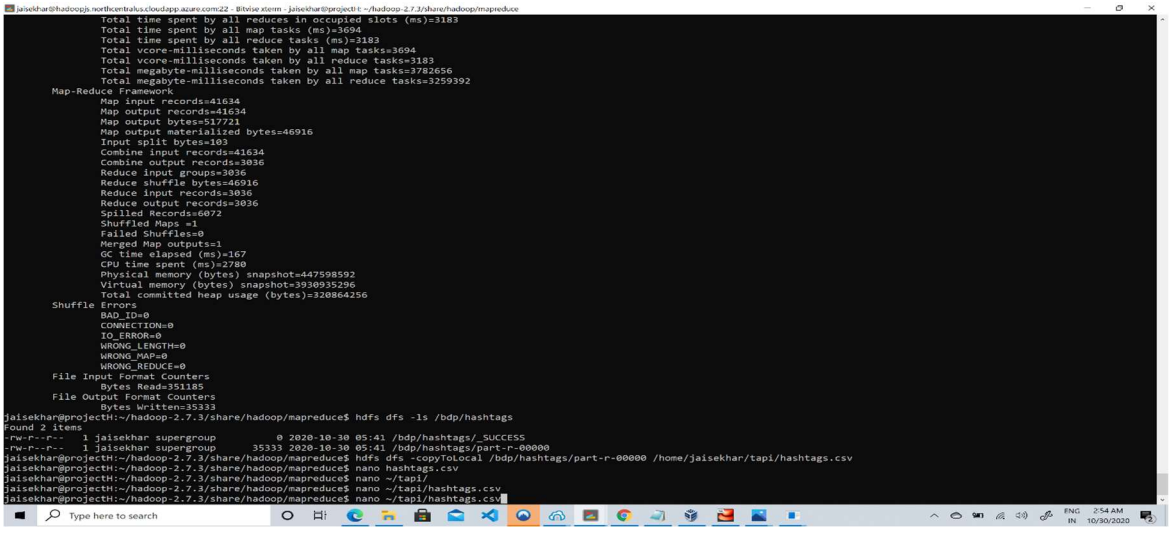
Now wordcount operation is performed using hadoop inbuilt example jar with hashtags.txt as input and saved the ouput in /bdp/hashtags directory in hdfs.

hadoop jar hadoop-mapreduce-examples-2.7.3.jar wordcount /bdp/hashtags.txt /bdp/hashtags

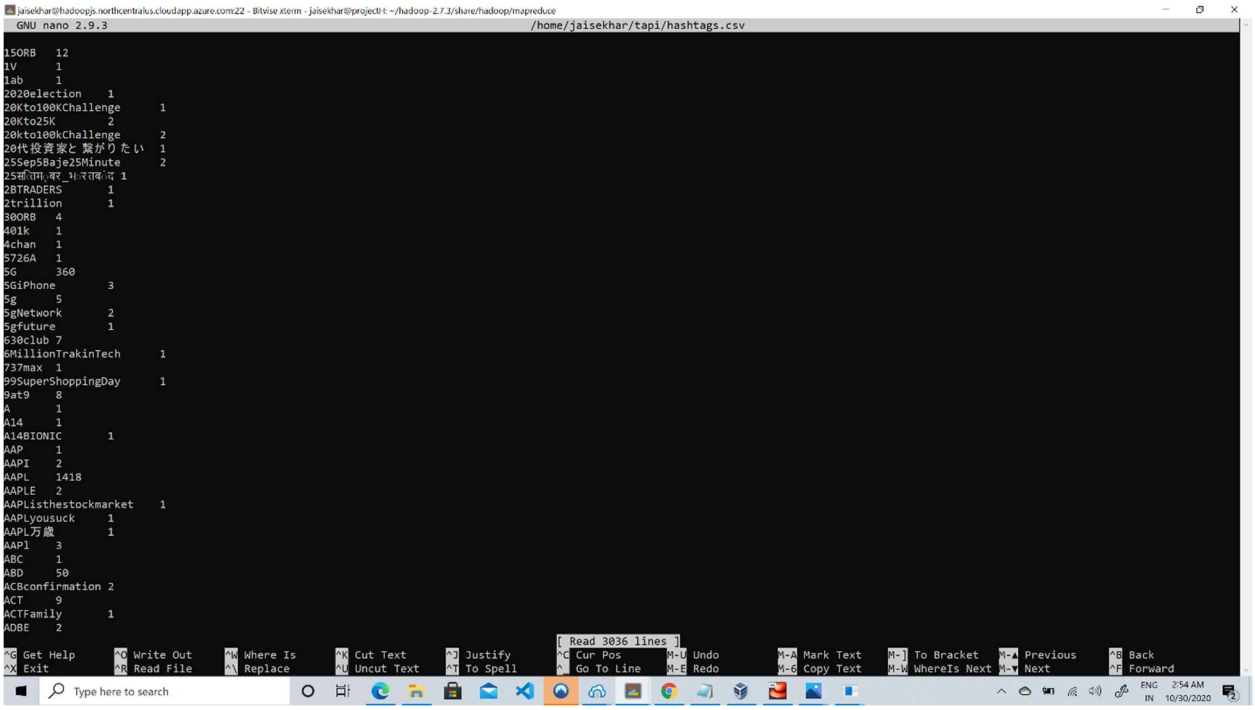


After the wordcount using the mapreduce, the saved output was sent to local for future analysis using the below commands.

hdfs dfs -copyToLocal /bdp/hashtags/part-r-00000 /home/jaisekhar/tapi/hashtags.csv



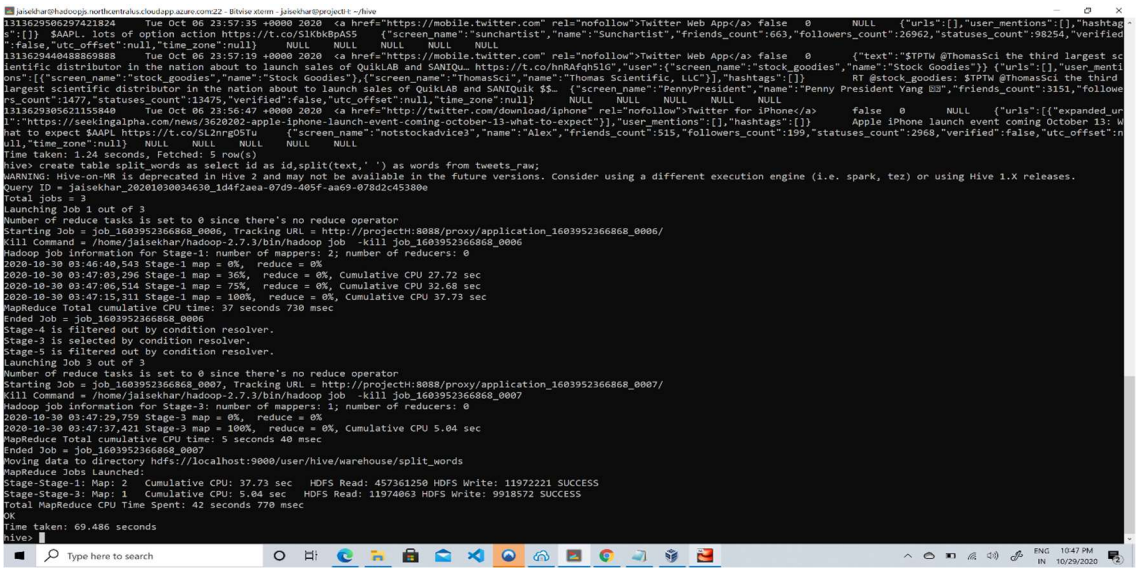
After performing the wordcount the output is like below where it shows the hashtag and their number of occurrences.



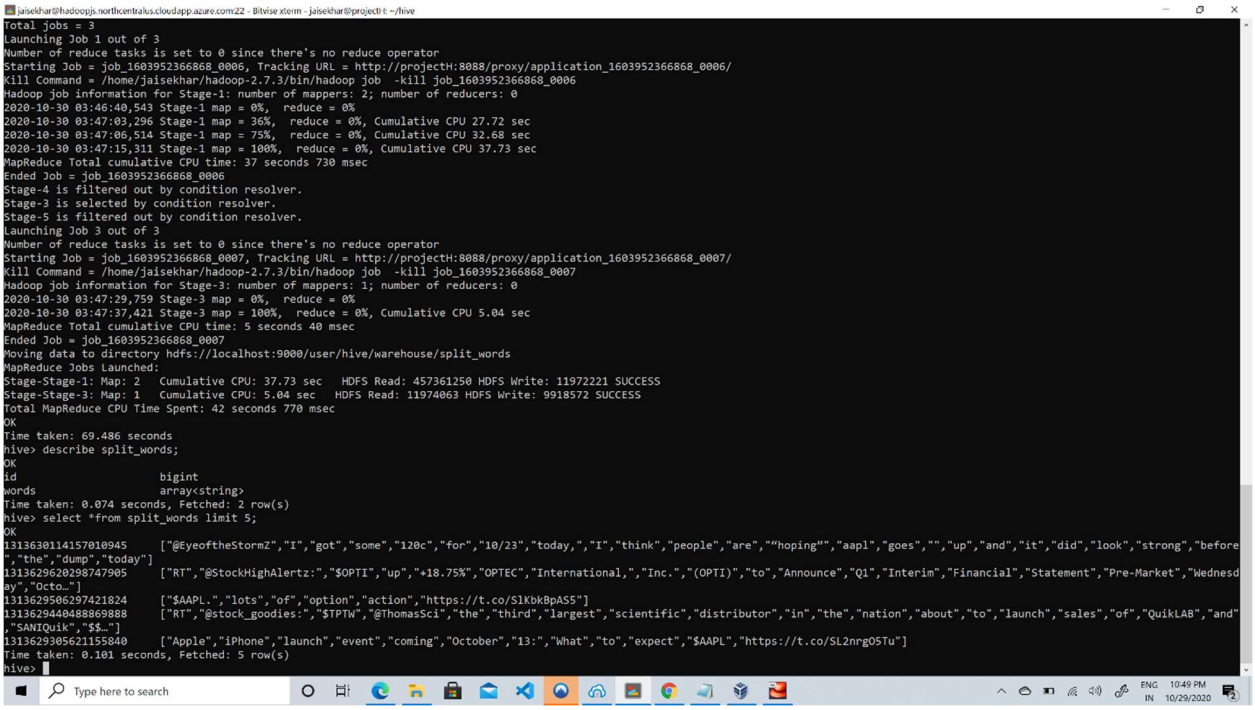
- ## Sentiment Analysis using Dictionary:

To perform sentiment analysis, we are using AFINN dictionary which contains more 2.5K words and values of polarity which ranges from -5.0 to +5.0 (Negative to Positive).

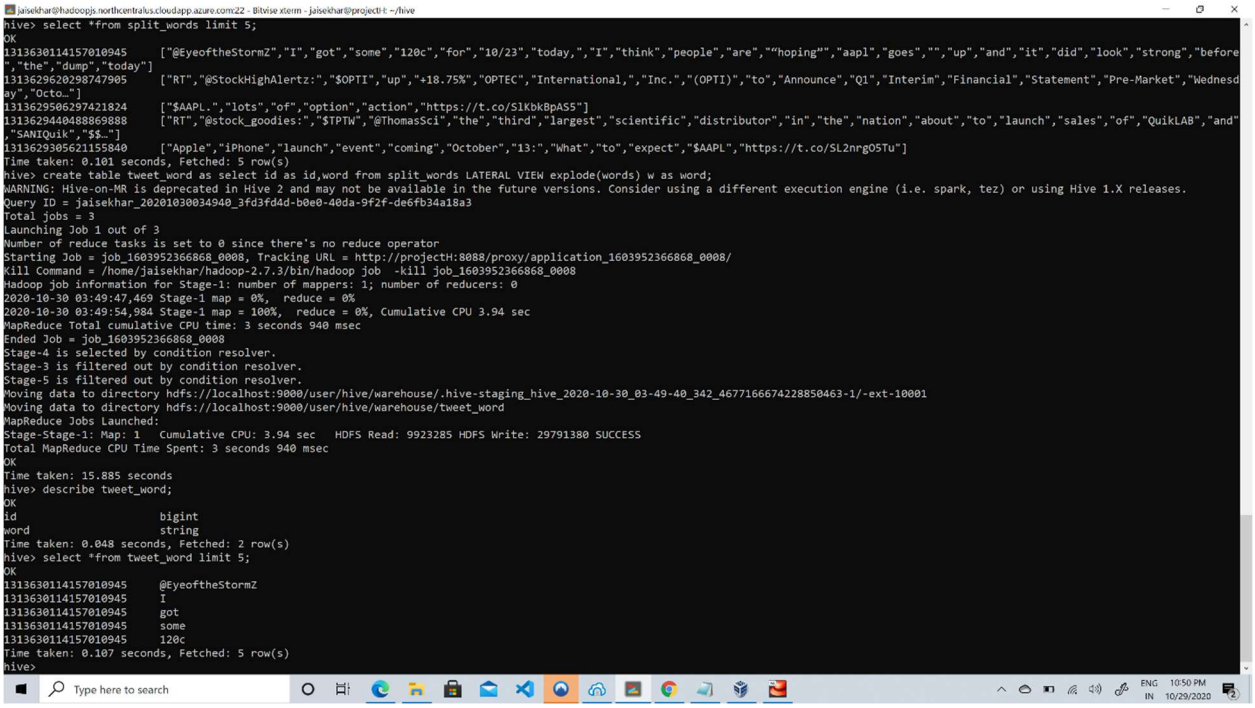
Initially, we separated the word using split function on text data in tweets _raw table and saved into split_words table using below command.



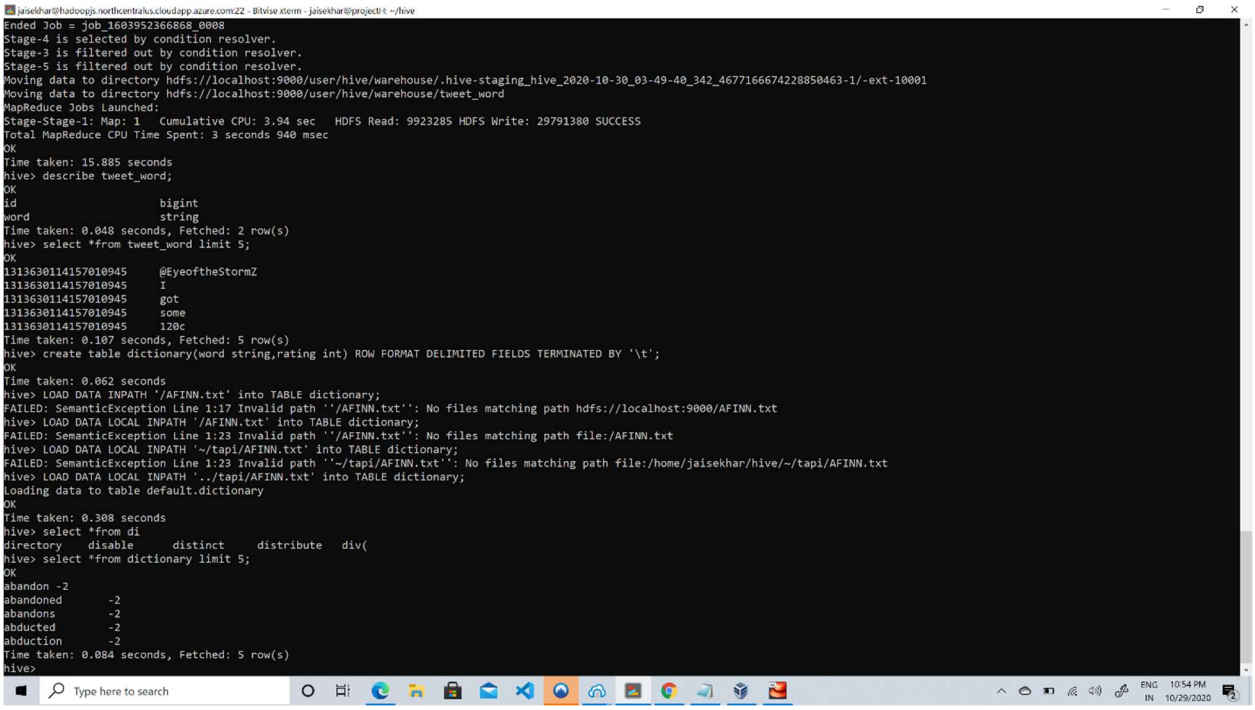
The split words in split_words table will look like below



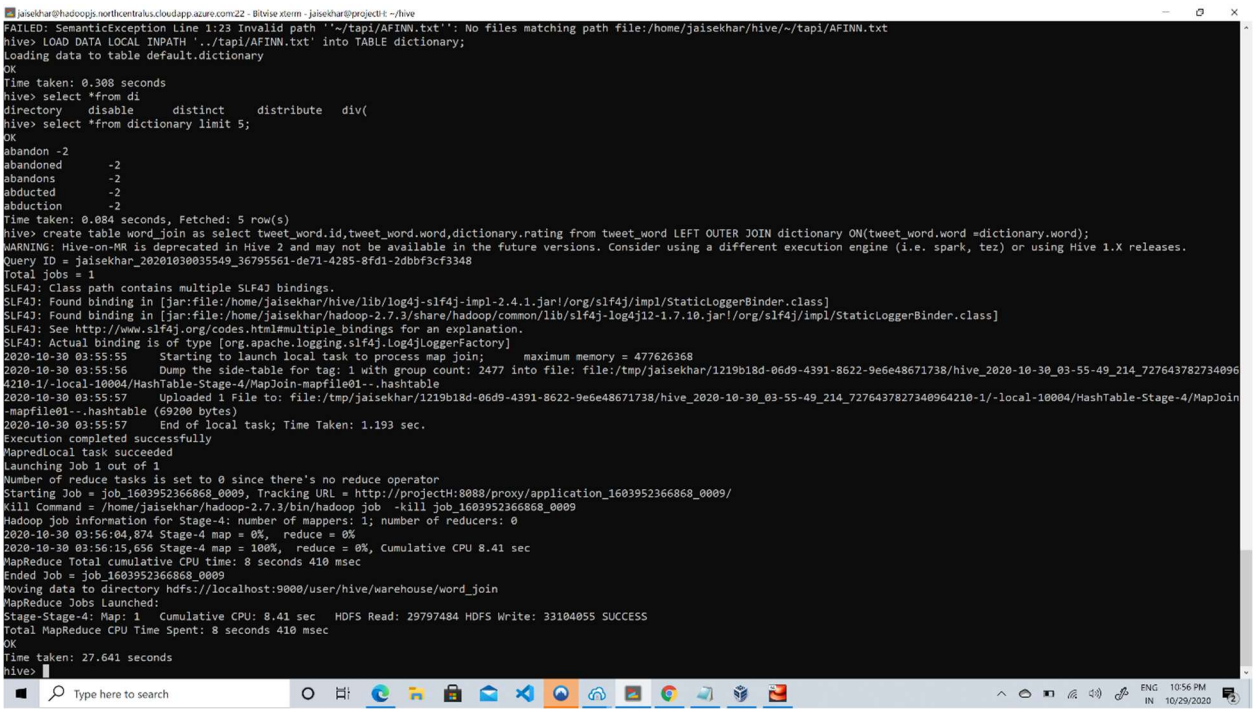
Now, we are splitting each word in the array as the new row. For this, we are using explode function which extracts the element from array into new row. Since it got some limitations we are using LATERAL VIEW. The output also shown below.



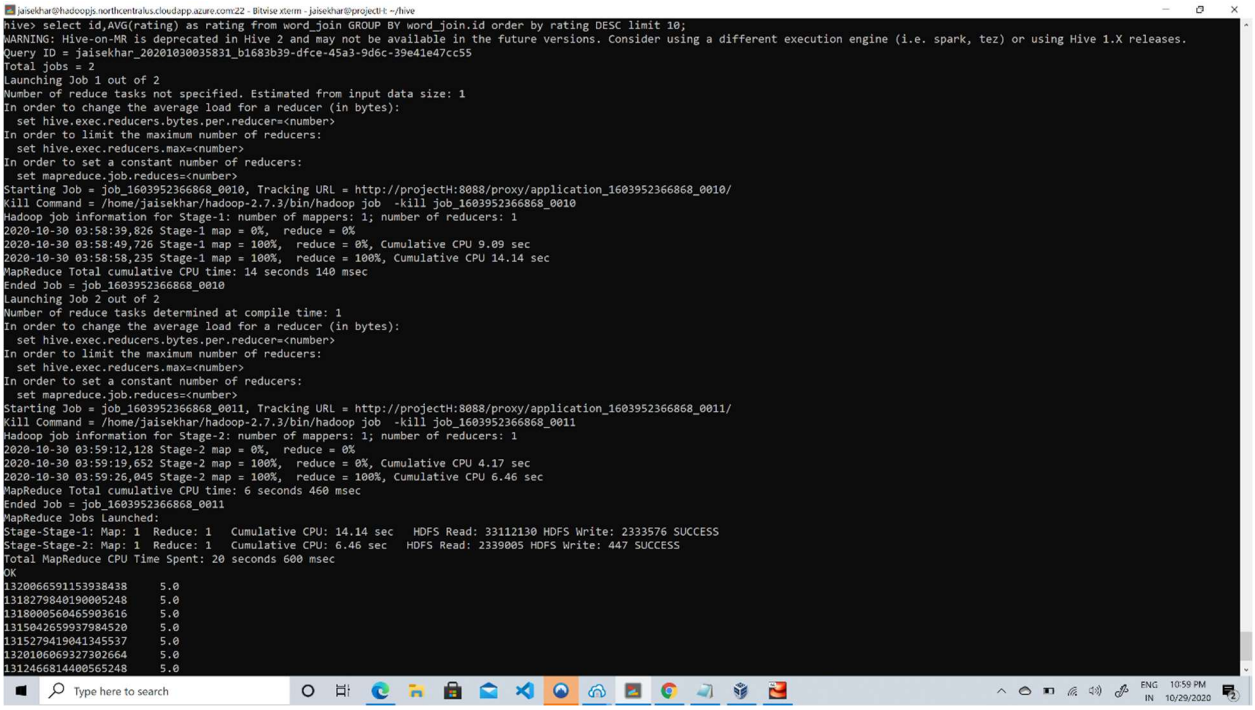
We got every word in new rows. Now, we will create new table named dictionary and will load the AFINN dictionary data into that table as as show below.



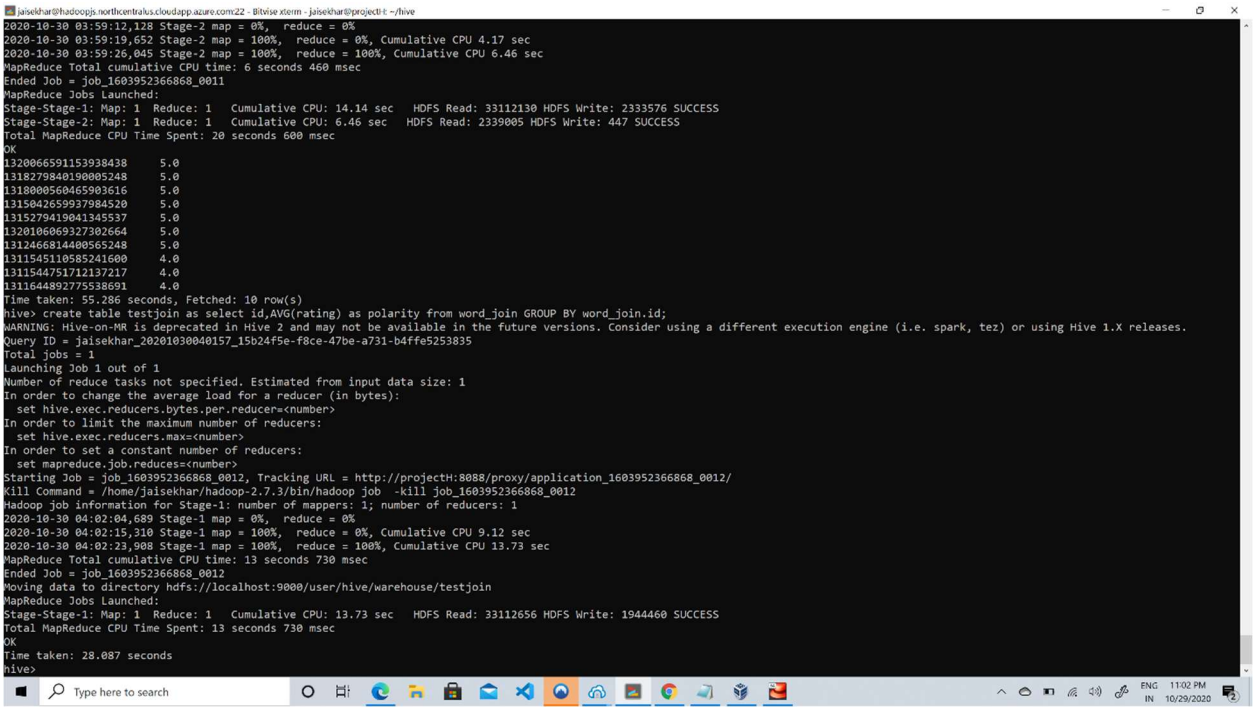
Create new table word_join by joining dictionary and tweet_word tables as shown below.



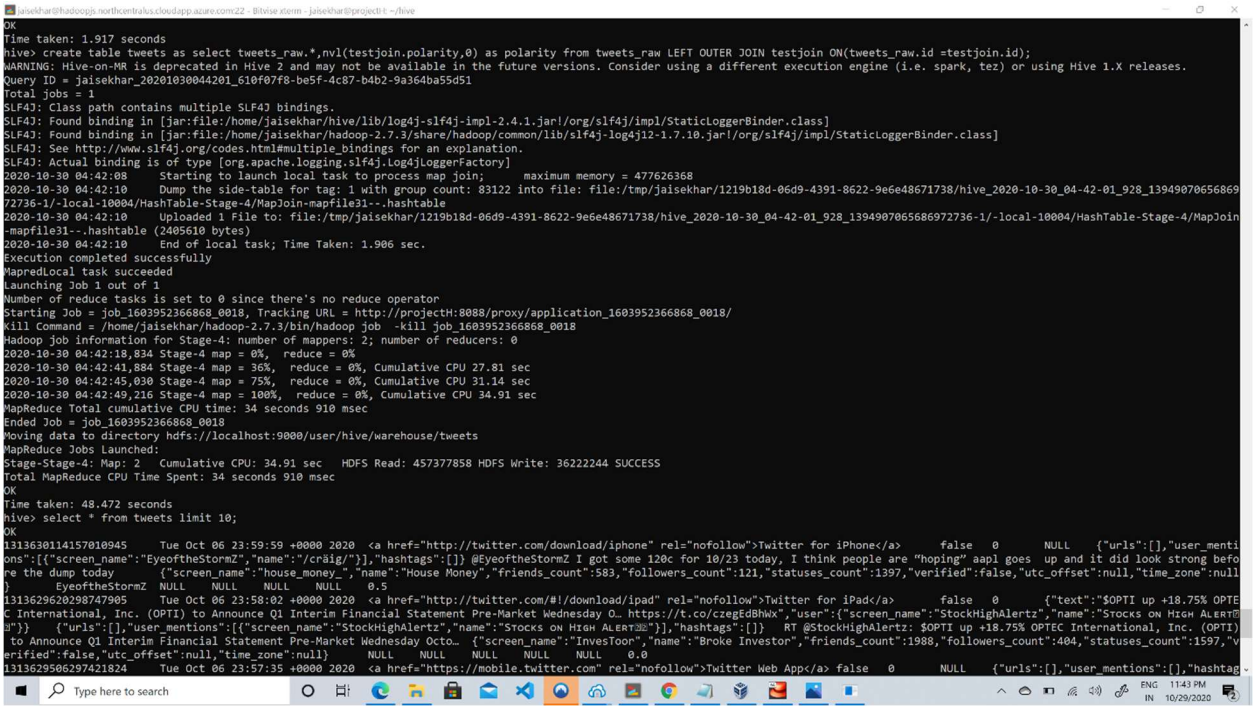
After joining now, we got the polarity value for each tweet in the word_join table which is shown below.



Now, we are saving the same id based grouped data into the testjoin table.



At last, we are joining the tweets_raw table and test join table based on id and created new table ‘tweets’.



Below, we can the see sample data which the results shows the tweet text and polarity value. This is one of the important part in this project. If you see the first tweets, it says holding the AAPL for overnight is risky hence it showed negative polarity value.

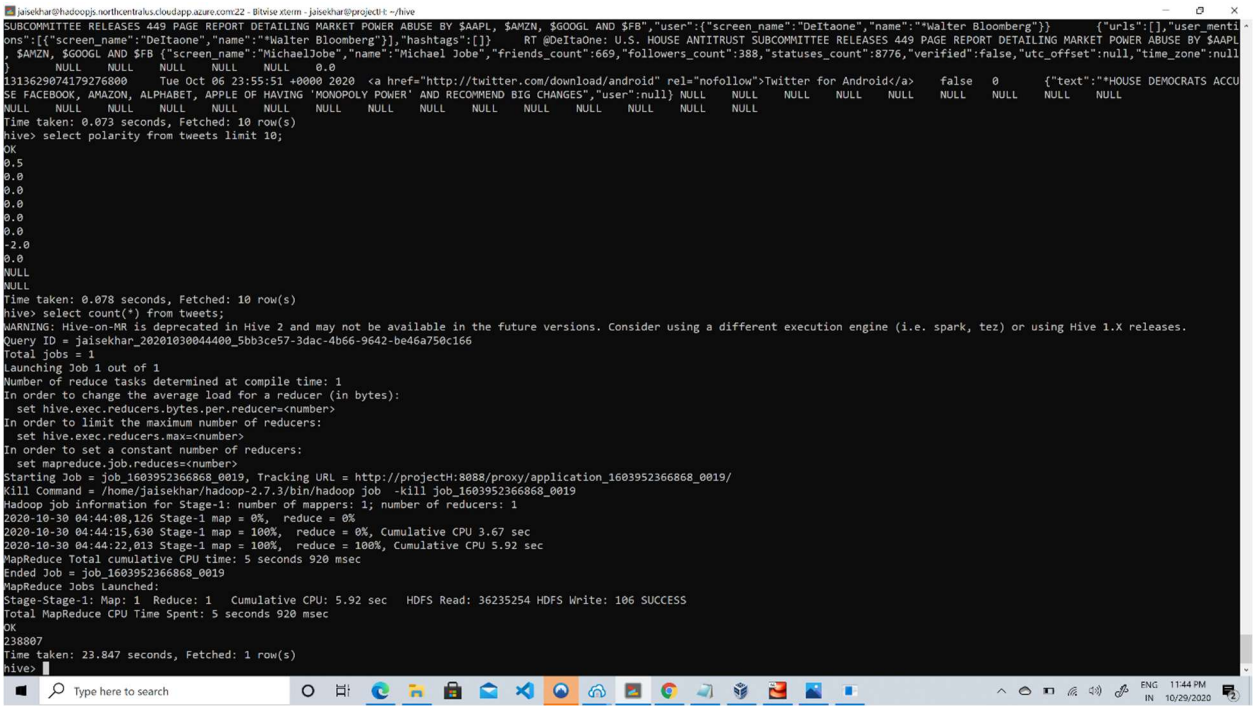
```
hive> select distinct(text),polarity from finaltweets where polarity is not null limit 10;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = jaisekhar_20201030040926_65b52753-6dc4-49ea-ac1f-5293b2c25413
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1603952366868_0014, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0014/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job  -kill job_1603952366868_0014
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-10-30 04:09:35,503 Stage-1 map = 0%,  reduce = 0%
2020-10-30 04:09:42,984 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.36 sec
2020-10-30 04:09:49,405 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.36 sec
MapReduce Total cumulative CPU time: 4 seconds 360 msec
Ended Job = job_1603952366868_0014
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 6.95 sec   HDFS Read: 36170628 HDFS Write: 1696 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 950 msec
OK
!ALERT! Bought $AAPL $126c @ $1.95 with some profits from earlier. Holding overnight going to risk it for a gap up into the event tomorrow.     -2.0
!ALERT! Bought $AAPL $127.50c @ $1.55 with some profits from earlier. Holding overnight going to risk it for a gap… https://t.co/imjIWboqOD     -2.0
" . . .  just like the Aug turn into Sep expiry (Note: the same thing would occur if the client were to unwind the… https://t.co/2pukITSPqm     2.0
" Hope Investors" in ITC don't get it:Contrarian Investing doesn't work this way: Identify beaten down names.Wait f… https://t.co/wO7yZQnu6E     -2.0
"Alerted live in chat with entry and exit. A cool 100% gain on this one and letting the runners go to work."… https://t.co/0zuun5z8Zx     1.5
"Alerted live in chat. Solid discipline. Give the stock time to play out your plan. This one never hit the stop so… https://t.co/1DfuI95mtF     -1.0
"Alerted live in chat. Solid discipline. Give the stock time to play out your plan. This one never hit the stop so… https://t.co/bNg1sSwe4O     -1.0
"Alerted live in chat. This one had multiple chances to buy low and sell for profit. Getting the right direction is… https://t.co/zgZI94fqqs     2.0
"Alerted live in chat. This one made a solid move up for the rest of the day. Don't forget to take profit.… https://t.co/TsjxMsmmNs     0.5
"Alerted live in chat. You would have had multiple chances to enter lower than our analyst. The important part of o… https://t.co/tknf28QxRl     2.0
Time taken: 24.68 seconds, Fetched: 10 row(s)
hive>
```

- ## Hive Analysis:

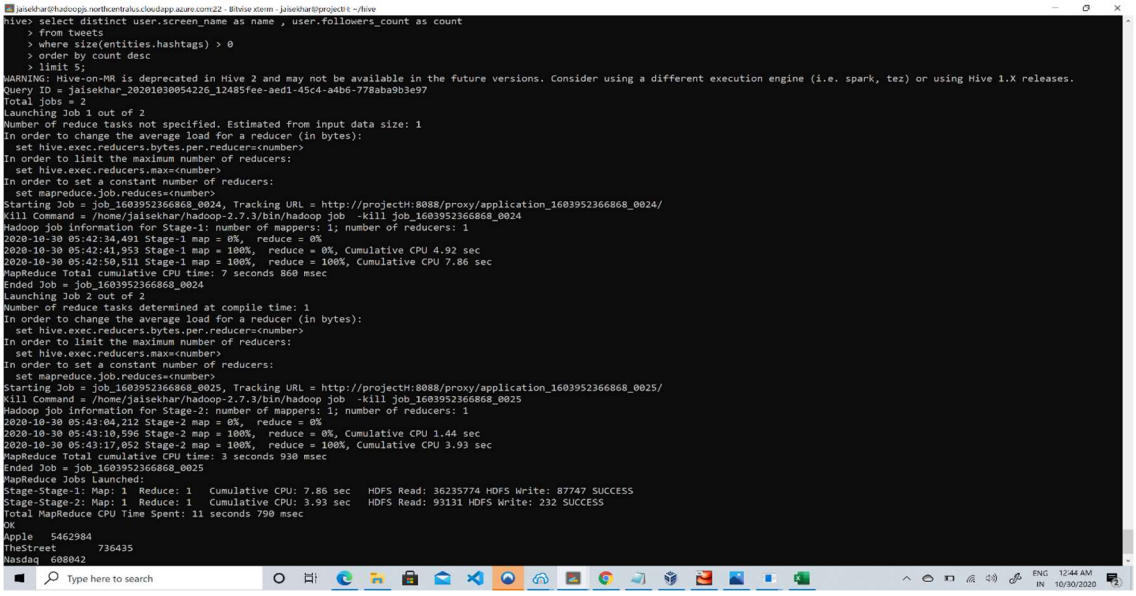
Below are the Queries for analysis using Hive.

1. The below simple query shows number of records in the tweets table.

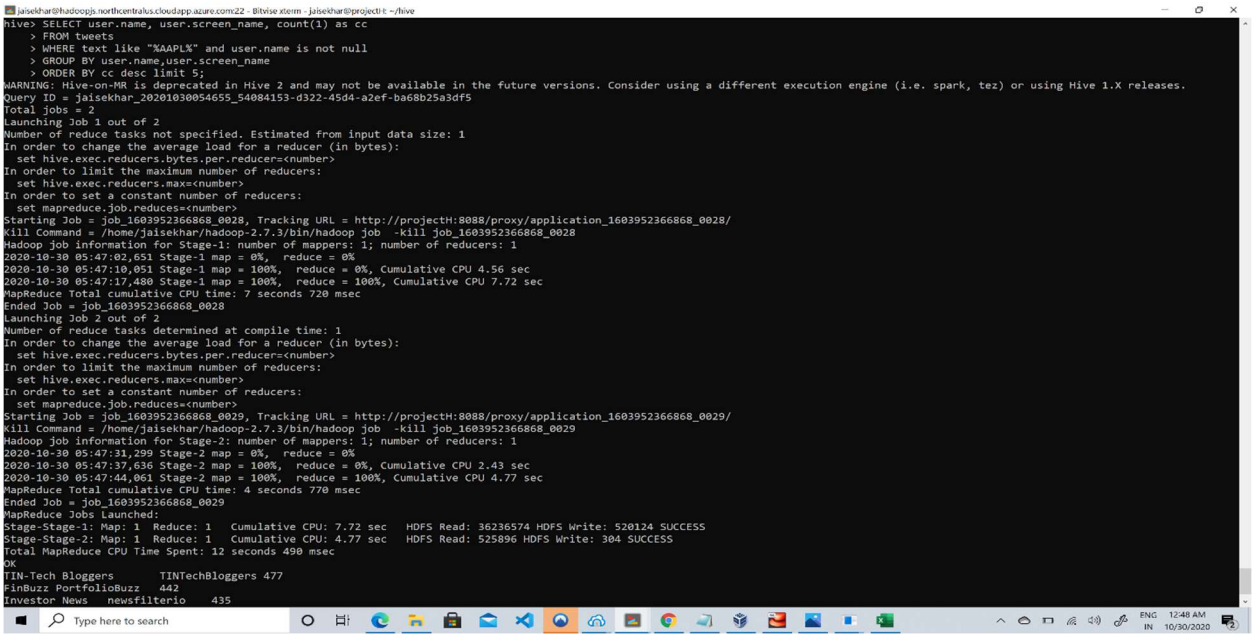
   select count(*) from tweets;



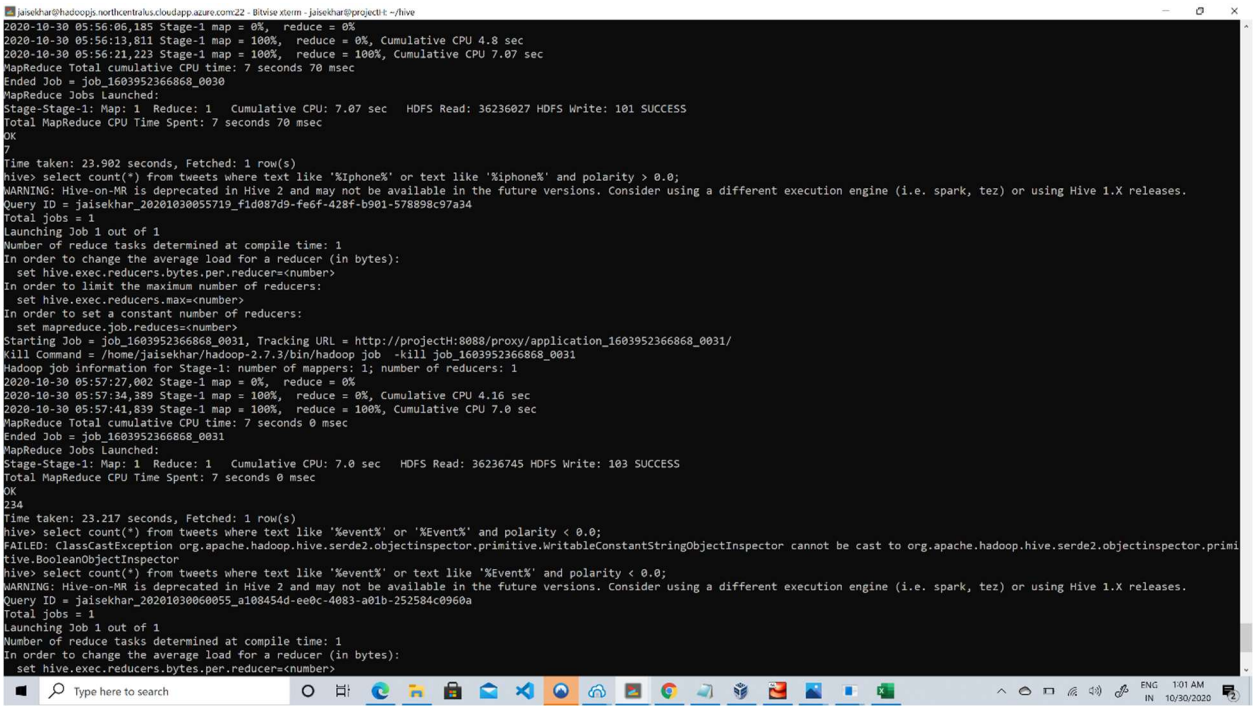
2. The below query shows the screenname of the user and there followers count in the descending order in whole dataset. In this dataset, Apple user has the more number of followers.



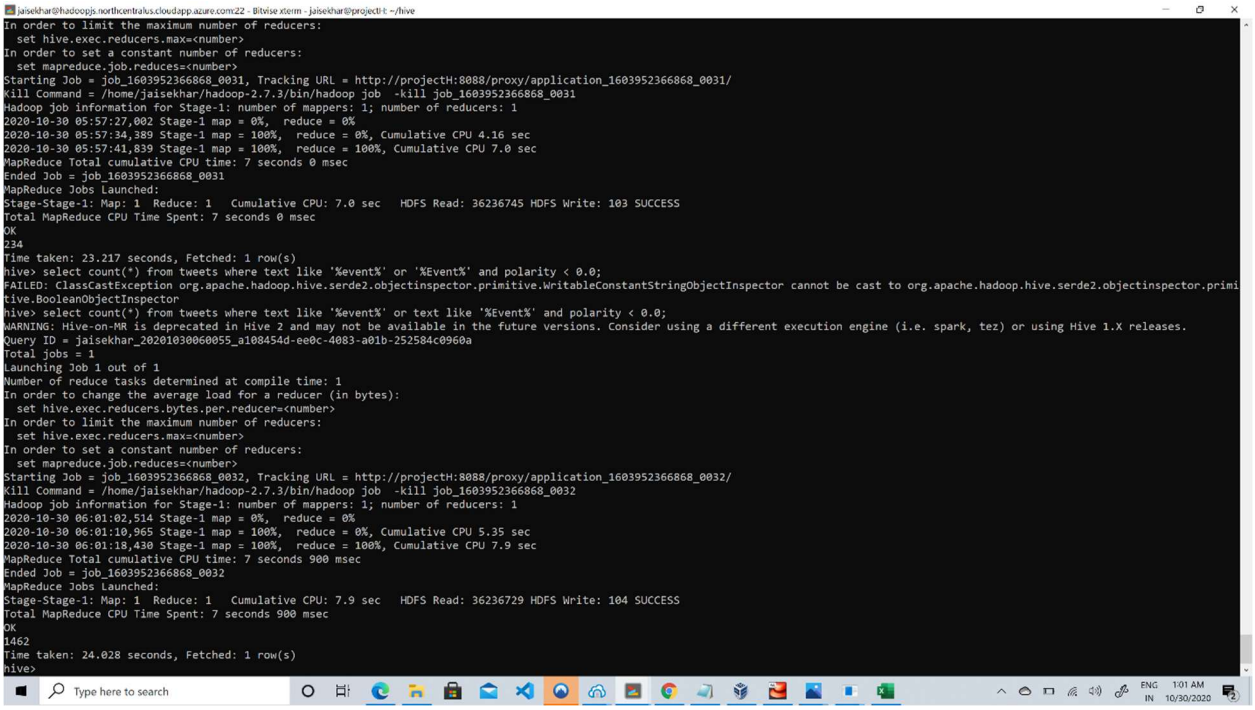
3. Using the below query, we are fetching the user screennames and number of tweets tweeted by them in the descending order. In this dataset, TIN-Tech Bloggers has tweeted more about AAPL.



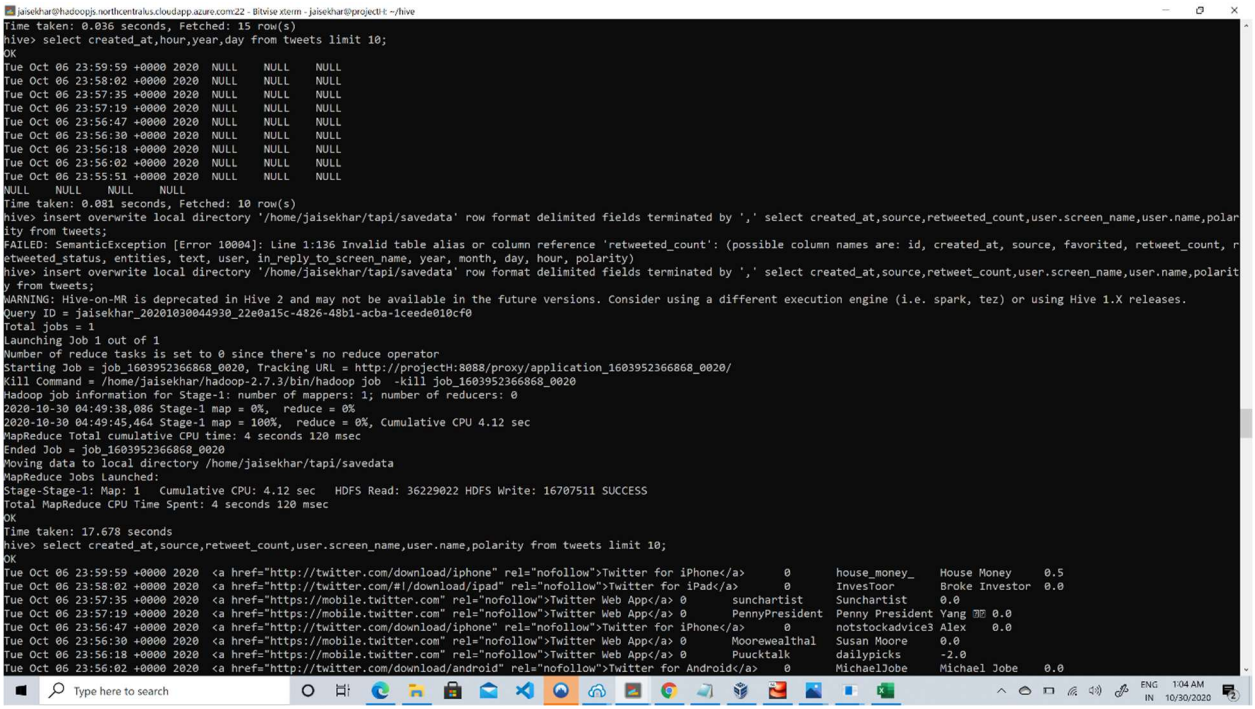
4. In the below query, we are getting the number of positive tweets which contains the keyword ‘Iphone’. There are 234 positive tweets about Iphone.



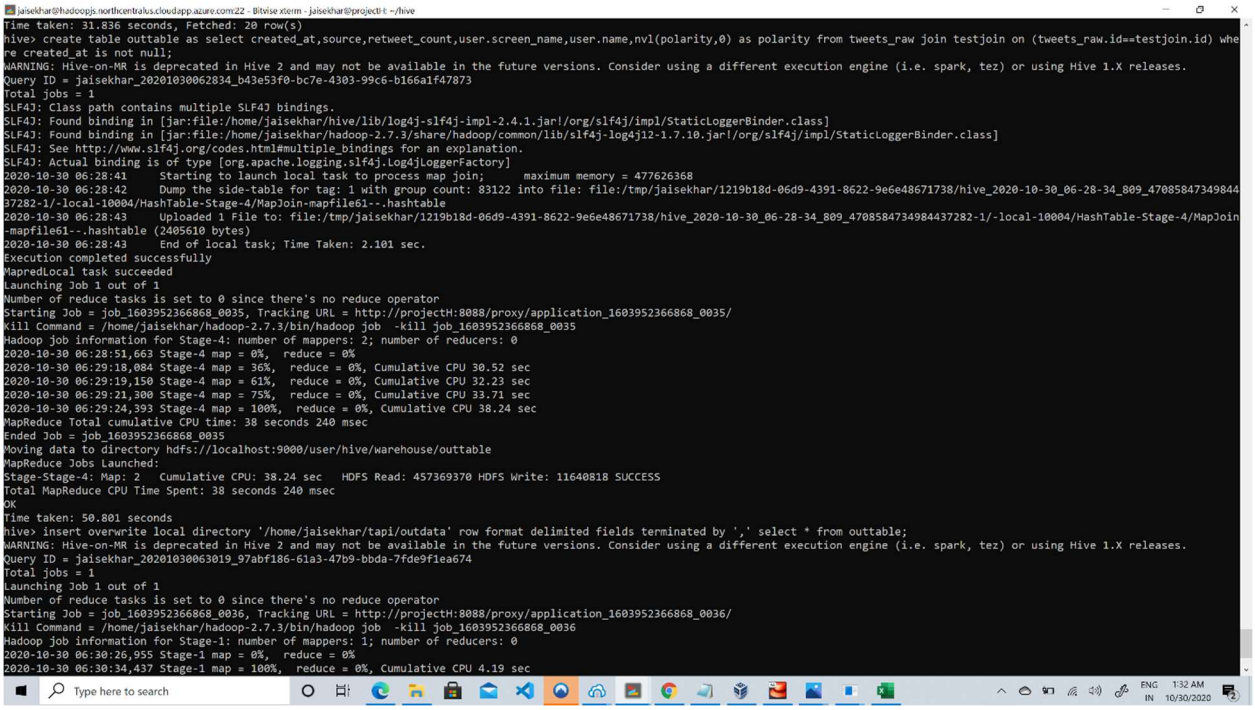
5. Here, we are getting the number of negative tweets which contains the keyword ‘Event’ and there are 1462 tweets in this dataset.



6. The below basic query returns the created time, source of the tweet, screenname, username and their polarities .

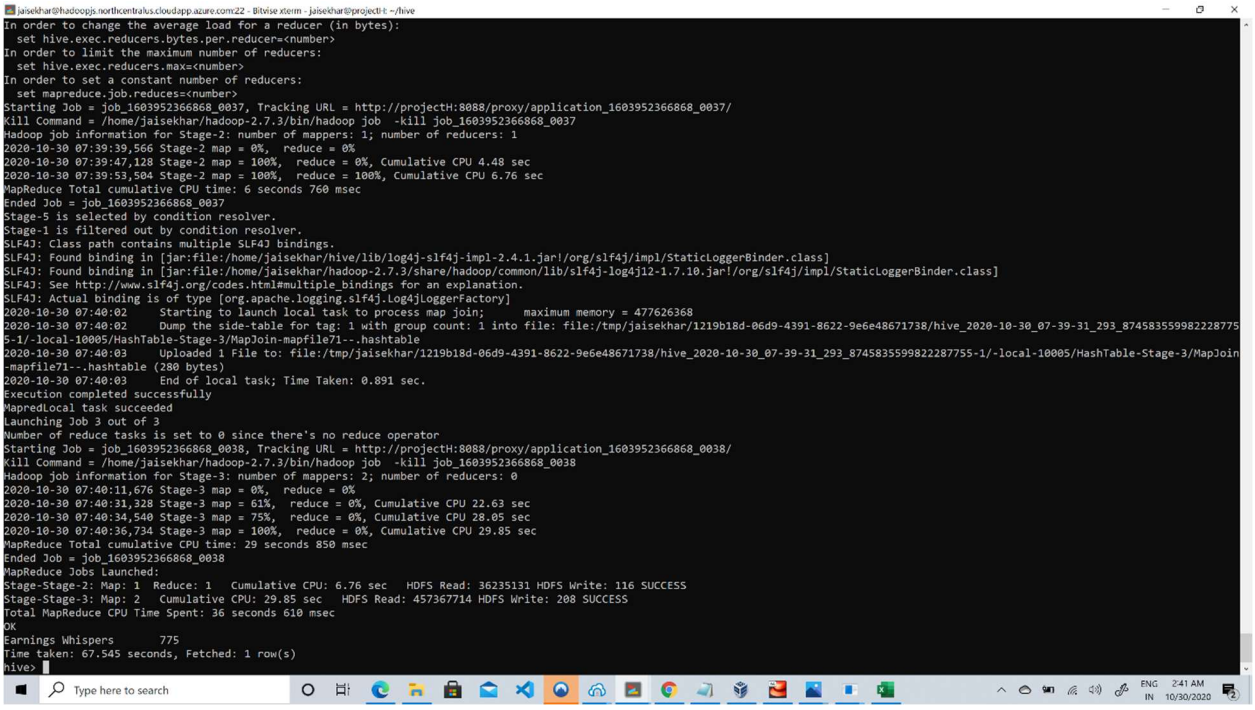


And the same data is stored in other csv file which helps in later part of analysis (Cassandra, visualization)

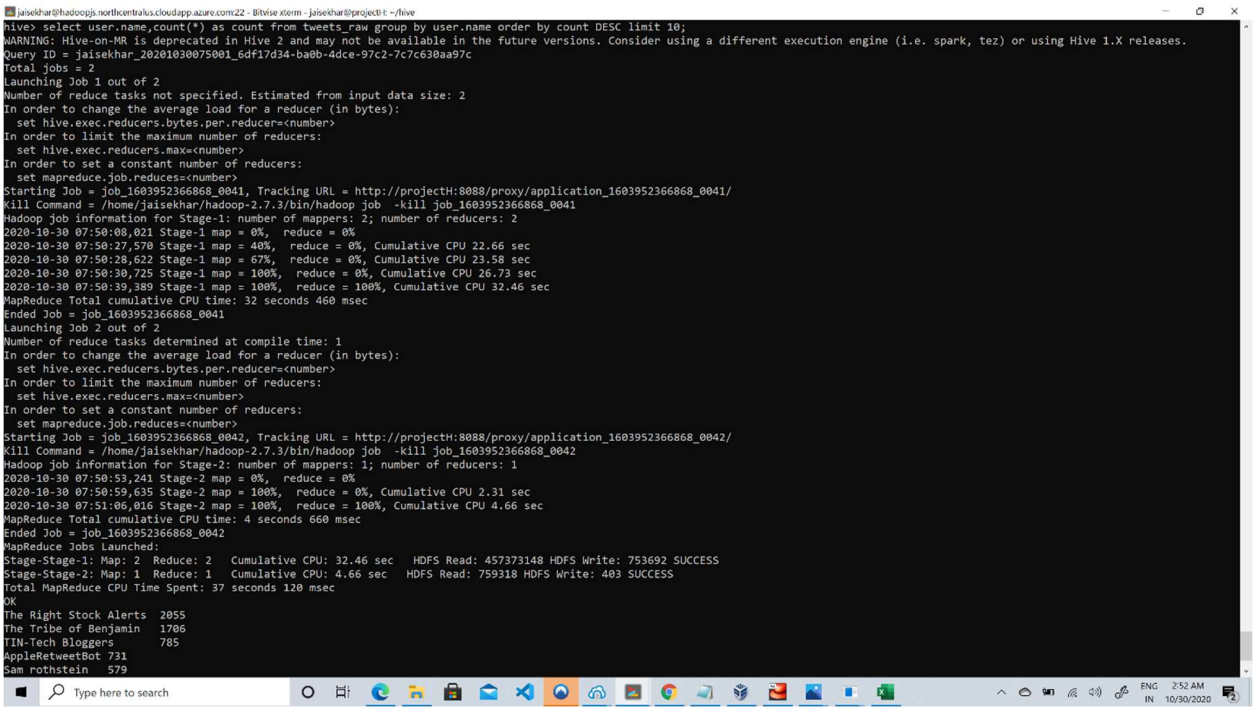


7. The below query shows the user's name whose tweets has maximum number of retweet count. In this dataset, Earnings Whispers tweet has more tweet count.

```
hive> select user.name, retweet_count from tweets_raw where retweet_count in (select max(retweet_count) from tweets);
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution en
Query ID = jaisekhar_20201030073931_8dc5c22d-0da3-4921-8945-61633365e4e8
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1603952366868_0037, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0037/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job  -kill job_1603952366868_0037
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2020-10-30 07:39:39,566 Stage-2 map = 0%,  reduce = 0%
2020-10-30 07:39:47,128 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 4.48 sec
2020-10-30 07:39:53,504 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 6.76 sec
MapReduce Total cumulative CPU time: 6 seconds 760 msec
Ended Job = job_1603952366868_0037
Stage-5 is selected by condition resolver.
Stage-1 is filtered out by condition resolver.
```

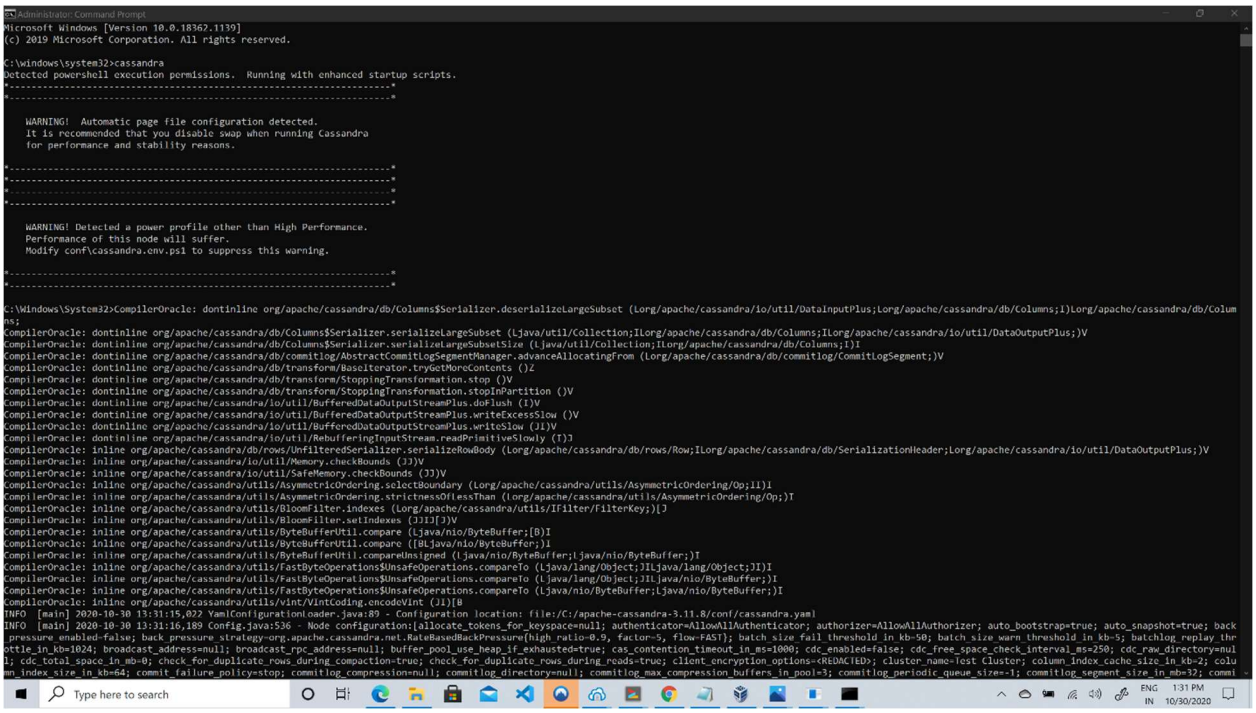


8. Using this query, we are getting User's name and their total number of tweets in the descending order

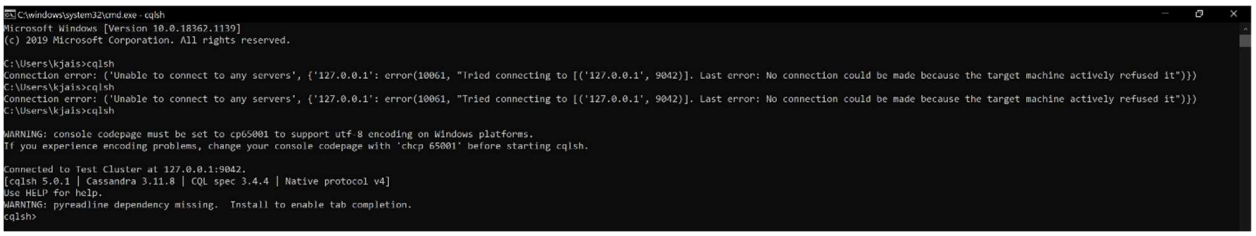


- ## Cassandra Analysis:

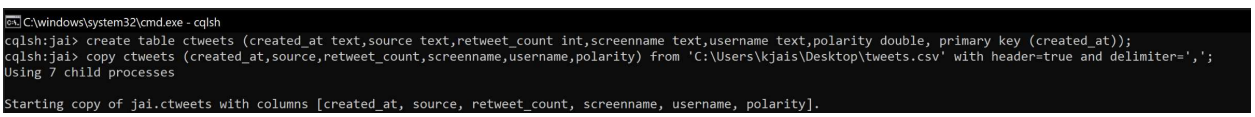
  Cassandra is started in command prompt administrator mode.



  Started cqlsh.



  Created table in Cassandra with required fields and copied the data from the tweets.csv into the ctweets table.



9. The below query gives the count of number of most positive tweets which means the polarity values is more than 3.

```
cqlsh:jai> select count(*) as positive from ctweets where polarity > 3 allow filtering;

 positive
----------
      414

(1 rows)

Warnings :
Aggregation query used without partition key
```

10. The below query gives the count of number of positive tweets whose polarity value is more than 0.

```
cqlsh:jai> select count(*) as positive from ctweets where polarity > 0 allow filtering;

 positive
----------
    12601

(1 rows)

Warnings :
Aggregation query used without partition key
```

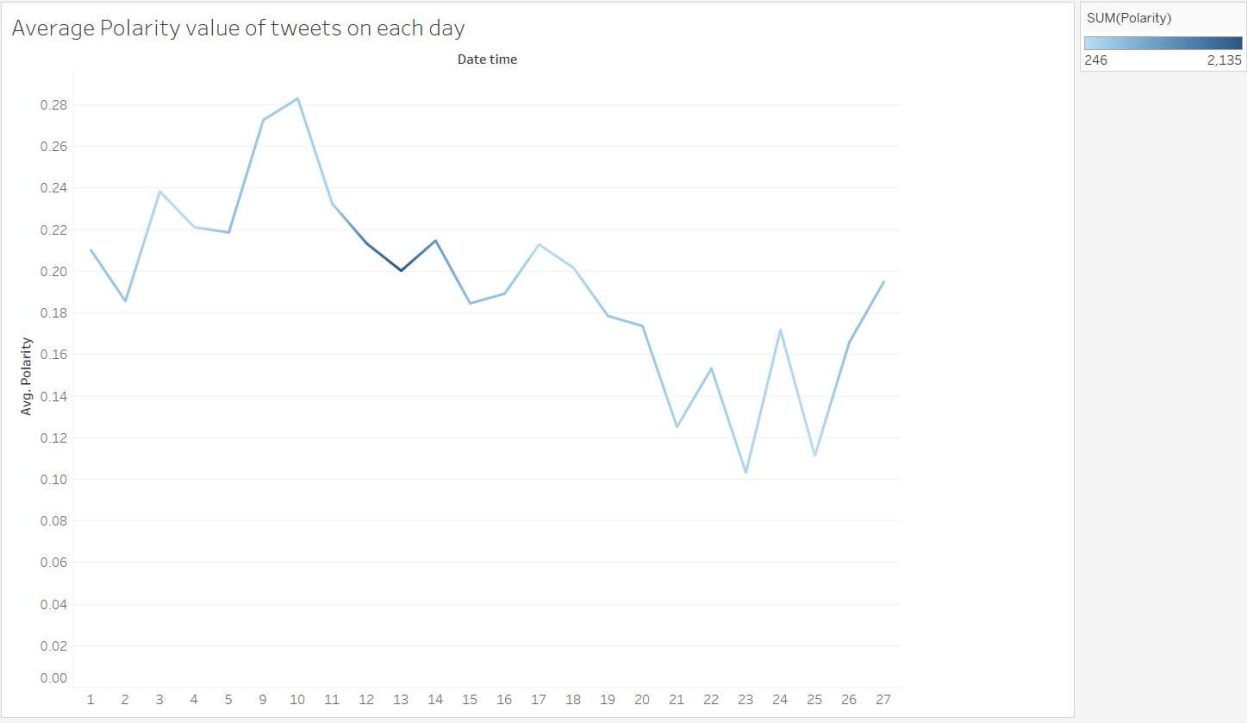
11. The below query gives the count of total negative tweets whose polarity value is less than 0.

```
cqlsh:jai> select count(*) as negative from ctweets where polarity < 0 allow filtering;

 negative
----------
     4896

(1 rows)

Warnings :
Aggregation query used without partition key
```

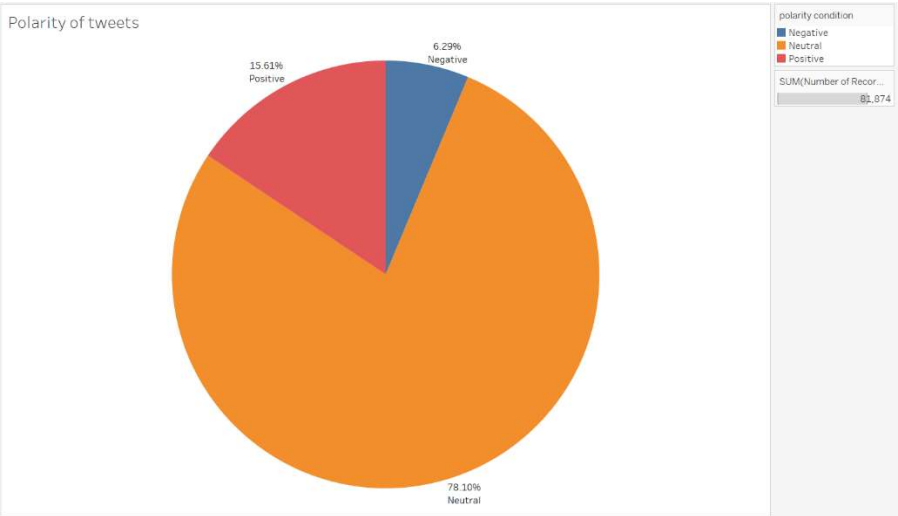
Visualization

We used Tableau for the visualilzation

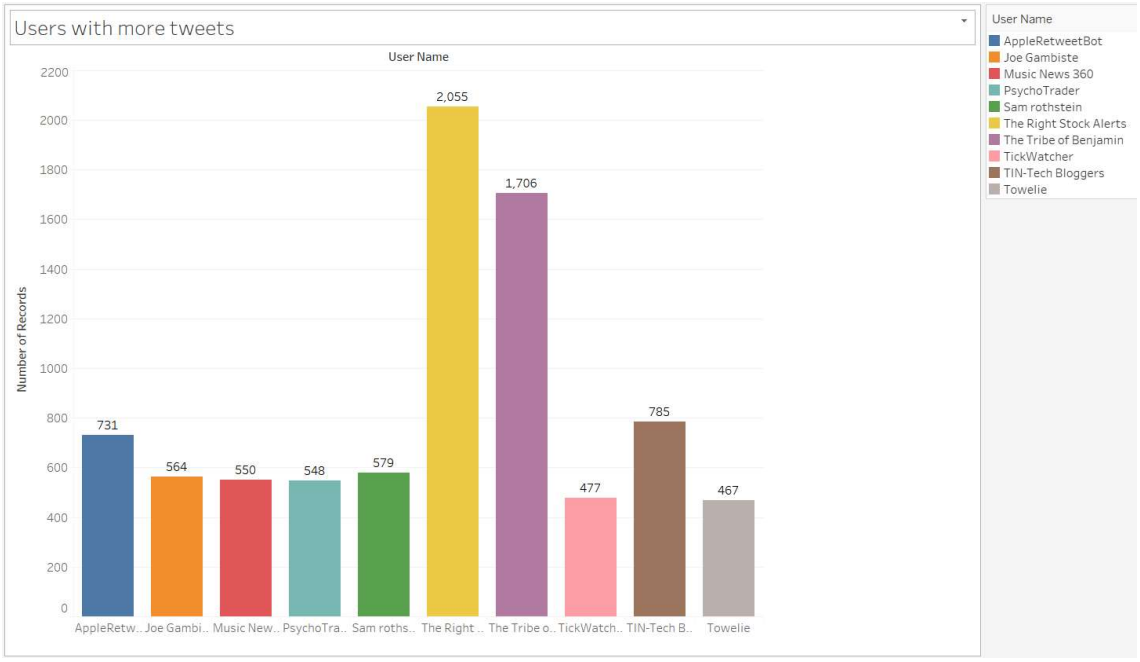
12. The Average polarity of tweets based on keyword **AAPL** on each day of October month. From October 5th to 10th there is a gradual increase in the polarity value. The highest average polarity value is recorded on October 10th. From 10th of the month polarity value gradually decreases.



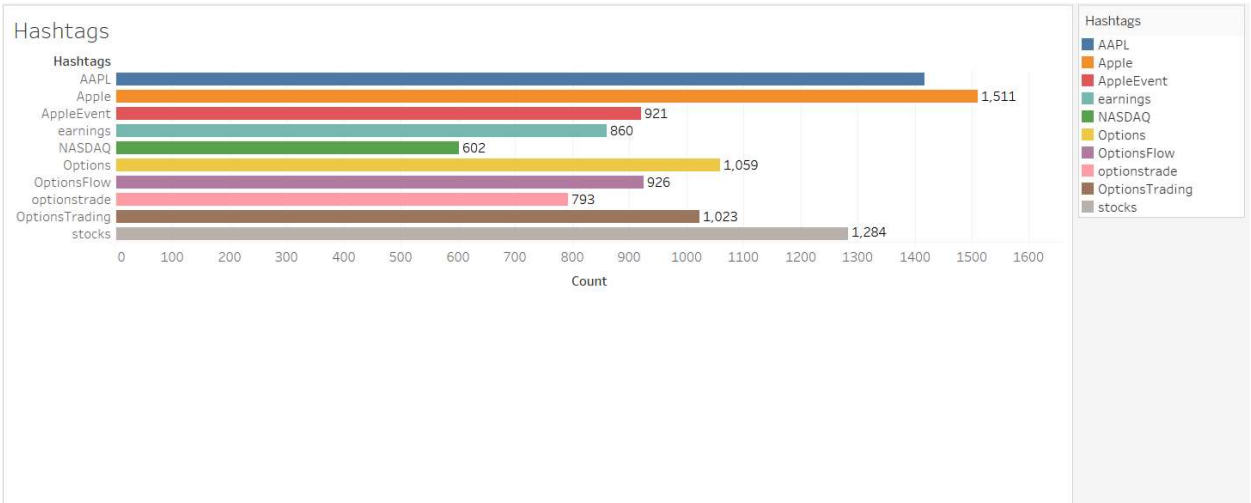
13. Polarity value of each tweet is calculated by using sentiment analysis on each tweet. We get values in the rage between -5 to +5. If the polarity value lies between -5 and 0 it is Negative. If the polarity value lies between 0 and 5 it is Positive. If the polarity value is 0 and 1 it is Neutral.
The data shows 78% tweets are neutral and remaining 22% of tweets are positive and negative.



14. Data shows the top 10 users with a greater number of tweets in the month of October. User with username **The Right Stock Alerts** has more tweets on AAPL stock in the month of October.



15. Data shows that Apple is the hashtag that has highest wordcount in the month of October



Project Management

- ## Work completed:
    1. Extracted tweets using python and twitter API for the dataset.
    2. Loading the data into the HDFS.
    3. Sentiment Analysis done on every tweet in the dataset.
    4. Wordcount on hashtags using mapreduce.
    5. Creation of table and loading of data into Hive.
    6. Analysis of data using Hive
    7. Analysis of data using Cassandra
    8. Visualization using Tableau.

- ## Work to be completed:
    1. Initialization and loading the data into Apache Spark.
    2. Using other components which will be covered in further classes.

- ## Contribution:

Jaisekhar Koya : 25%

Extracting tweets using Python. Done sentimental analysis using AFINN dictionary which gives polarity value. Extracted tweets were stored in HDFS. Created and loaded data to hive. Performed few queries in Hive and also Cassandra.

Sri Sai Nikhil Kantipudi: 25%

Helped in doing sentimental analysis on every tweet in the dataset. Done wordcount on extracted hashtags using mapreduce. Contributed in loading and analysing the data in Hive. Performed few queries in Hive and also in Cassandra

Sai Rohith Guntupally: 25%

Tweets extracted and parsed for hashtags using python. Done hdfs operations on the extracted dataset. Performed few queries on the dataset using hive and Cassandra. Part of visualization in tableau.

Aarthi Nagireddy: 25%

Sentiment Analysis using dictionary system. Hive initialization and loading of data of the dataset. Performed few queries in Hive and Cassandra for data analysis. Part of Visualization work in tableau.

Story Approach

- ### Who?

  The dataset is about the users who tweeted with the keyword 'AAPL' (in this case) in the twitter. There are many numbers of users who tweets in twitter. Thus, it help us in analyzing more data. This dataset contains only public available information . It doesn't contain any undisclosable information. These users are the main representatives stated in the assignment 1.

- ### What?

  The dataset contains the tweets extracted from the twitter which contains user information, tweet text, hashtags, created time and their entity information. This are the important variables used in the project. This covers all the required variables stated in the assignment 1.

- ### When?

  The data is generated when the user tweets about AAPL stock in the twitter. There will be a greater number of users who tweets same. The data in out dataset consists of all the tweets which contains AAPL keyword in Oct 1st to Oct 28th. Hence this is the old data not the real time data. This is permanent data which might need to extract more data in future for further analysis of other data range. It satisfied the abstract stated in assignment 1.

- ### Where?

  The dataset consists of tweets which are extracted from twitter. Twitter consists of huge amount of data. It is an online social media platform. It has more number of users. This data set contains geographical information of the tweeted location. This dataset doesn't have geographical limitations. This contains the data as stated in Assignment 1.

- ### Why?

  Twitter is one of the important sources for the data. Many number of people puts their opinion in the twitter. In the same way, many companies or investors puts their news and opinions in twitter which effects in stock trend. Hence we are using their opinions(tweets) to analyze the stock movement.

- ### How?

  Most of this part was already covered above. But to be precise, we are collecting the tweets from twitter and doing sentiment analysis to find whether their opinion is positive or negative and do the analysis on the whole data using hadoop ecosystem.

- ### Calibration 1: Every aspect covered in Assignment 1 aligns with the Assignment 2. We have choose right dataset in Assignment 2 for analysis as mentioned in the Assignment 1. And, this is the relevant dataset to the project.

References

- ANALYSIS AND PREDICTION OF STOCK MARKET USING BIG DATA TECHNOLOGY https://www.academia.edu/36679481/ANALYSIS_AND_PREDICTION_OF_STOCK_MARKET_USING_BIG_DATA_TECHNOLOGY
- Stock Market Prediction on Bigdata Using Machine Learning Algorithm http://ijesc.org/upload/b91a9a994c4d79a72f5542393ca9469d.Stock%20Market%20Prediction%20on%20Bigdata%20Using%20Machine%20Learning%20Algorithm.pdf