

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**

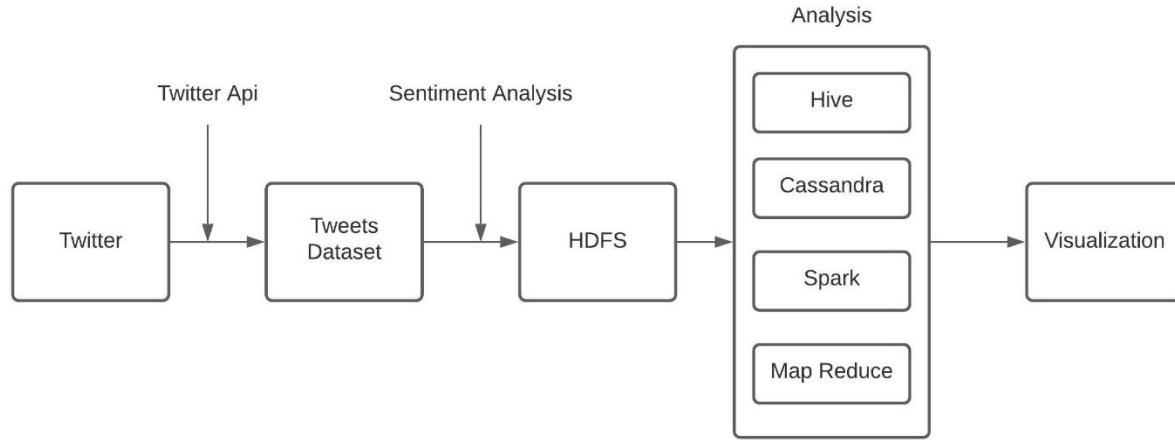
The main features of this project is twitter data extraction, sentimental analysis on the extracted data. Using HDFS to store the data. Apache spark to process the data and Hive for querying the data. A visualization tool for showing out the results.

Dataset:

Twitter Dataset:

https://mailmissouri-my.sharepoint.com/:u/g/personal/skgyc_umsystem_edu/EXI5i-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

```
curl -X GET "https://api.twitter.com/2/tweets/search/recent?query=AAPL&start_time=2020-10-01T00%3A00%2B0000&end_time=2020-10-28T23%3A59%2B0000&tweet_mode=extended&max_results=100000" -H "Authorization: Bearer <REDACTED>" -H "User-Agent: Python-requests/2.27.1 aiohttp/3.8.0" -H "Accept: application/json"
```

Sample JSON output (tweets1.json):

```
[{"created_at": "Tue Oct 06 23:09:50 -0000 2020", "id": "1136290813415791044", "id_str": "1136290813415791044", "text": "#AAPL #stocks I got some like for 10/23 today, I think people are like \"", "created_at": "Tue Oct 06 23:09:52 -0000 2020", "id": "1136290813415791045", "id_str": "1136290813415791045", "text": "RT @jaiselkar: #MFT up ~18.7% QPTEC International, Inc. (OPTZ)", "created_at": "Tue Oct 06 23:13:35 -0000 2020", "id": "11362908137411824", "id_str": "11362908137411824", "text": "#AAPL lots of option action https://t.co/5lxhRqB55", "display_text_range": "11362908137411824", "entities": {"urls": [{"url": "https://t.co/5lxhRqB55"}], "hashtags": [{"tag": "#AAPL"}]}, "created_at": "Tue Oct 06 23:16:47 -0000 2020", "id": "11362908138155844", "id_str": "11362908138155844", "text": "RT @jaiselkar: U.S. HOUSE ANTITRUST SUBCOMMITTEE RELEASES 640 PAGE REPORTS", "display_text_range": "11362908138155844", "entities": {"urls": [{"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}], "hashtags": [{"tag": "#AAPL"}]}, "created_at": "Tue Oct 06 23:18:10 -0000 2020", "id": "11362908140793481", "id_str": "11362908140793481", "text": "RT @jaiselkar: U.S. HOUSE ANTITRUST SUBCOMMITTEE RELEASES 640 PAGE REPORTS", "display_text_range": "11362908140793481", "entities": {"urls": [{"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}], "hashtags": [{"tag": "#AAPL"}]}, "created_at": "Tue Oct 06 23:18:51 -0000 2020", "id": "1136290814097923", "id_str": "1136290814097923", "text": "RT @jaiselkar: #Tech Stocks are going to crush the shorts overnight and at open and the destiny is ours", "display_text_range": "1136290814097923", "entities": {"urls": [{"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}], "hashtags": [{"tag": "#AAPL"}]}, "created_at": "Tue Oct 06 23:18:51 -0000 2020", "id": "11362908141172600", "id_str": "11362908141172600", "text": "RT @jaiselkar: #Tech Stocks are going to crush the shorts overnight and at open and the destiny is ours", "display_text_range": "11362908141172600", "entities": {"urls": [{"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}], "hashtags": [{"tag": "#AAPL"}]}, "created_at": "Tue Oct 06 23:18:51 -0000 2020", "id": "11362908141172601", "id_str": "11362908141172601", "text": "RT @jaiselkar: #Tech Stocks are going to crush the shorts overnight and at open and the destiny is ours", "display_text_range": "11362908141172601", "entities": {"urls": [{"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}], "hashtags": [{"tag": "#AAPL"}]}, "created_at": "Tue Oct 06 23:18:51 -0000 2020", "id": "11362908141172602", "id_str": "11362908141172602", "text": "RT @jaiselkar: #Tech Stocks are going to crush the shorts overnight and at open and the destiny is ours", "display_text_range": "11362908141172602", "entities": {"urls": [{"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}], "hashtags": [{"tag": "#AAPL"}]}, "created_at": "Tue Oct 06 23:18:51 -0000 2020", "id": "11362908141172603", "id_str": "11362908141172603", "text": "RT @jaiselkar: #Tech Stocks are going to crush the shorts overnight and at open and the destiny is ours", "display_text_range": "11362908141172603", "entities": {"urls": [{"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}], "hashtags": [{"tag": "#AAPL"}]}, "created_at": "Tue Oct 06 23:18:51 -0000 2020", "id": "11362908141172604", "id_str": "11362908141172604", "text": "RT @jaiselkar: #Tech Stocks are going to crush the shorts overnight and at open and the destiny is ours", "display_text_range": "11362908141172604", "entities": {"urls": [{"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}], "hashtags": [{"tag": "#AAPL"}]}, "created_at": "Tue Oct 06 23:18:51 -0000 2020", "id": "11362908141172605", "id_str": "11362908141172605", "text": "RT @jaiselkar: #Tech Stocks are going to crush the shorts overnight and at open and the destiny is ours", "display_text_range": "11362908141172605", "entities": {"urls": [{"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}], "hashtags": [{"tag": "#AAPL"}]}, "created_at": "Tue Oct 06 23:18:51 -0000 2020", "id": "11362908141172606", "id_str": "11362908141172606", "text": "RT @jaiselkar: #Tech Stocks are going to crush the shorts overnight and at open and the destiny is ours", "display_text_range": "11362908141172606", "entities": {"urls": [{"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}], "hashtags": [{"tag": "#AAPL"}]}, "created_at": "Tue Oct 06 23:18:51 -0000 2020", "id": "11362908141172607", "id_str": "11362908141172607", "text": "RT @jaiselkar: #Tech Stocks are going to crush the shorts overnight and at open and the destiny is ours", "display_text_range": "11362908141172607", "entities": {"urls": [{"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}], "hashtags": [{"tag": "#AAPL"}]}, "created_at": "Tue Oct 06 23:18:51 -0000 2020", "id": "11362908141172608", "id_str": "11362908141172608", "text": "RT @jaiselkar: #Tech Stocks are going to crush the shorts overnight and at open and the destiny is ours", "display_text_range": "11362908141172608", "entities": {"urls": [{"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}], "hashtags": [{"tag": "#AAPL"}]}, "created_at": "Tue Oct 06 23:18:51 -0000 2020", "id": "11362908141172609", "id_str": "11362908141172609", "text": "RT @jaiselkar: #Tech Stocks are going to crush the shorts overnight and at open and the destiny is ours", "display_text_range": "11362908141172609", "entities": {"urls": [{"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}], "hashtags": [{"tag": "#AAPL"}]}, "created_at": "Tue Oct 06 23:18:51 -0000 2020", "id": "11362908141172610", "id_str": "11362908141172610", "text": "RT @jaiselkar: #Tech Stocks are going to crush the shorts overnight and at open and the destiny is ours", "display_text_range": "11362908141172610", "entities": {"urls": [{"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}], "hashtags": [{"tag": "#AAPL"}]}, "created_at": "Tue Oct 06 23:18:51 -0000 2020", "id": "11362908141172611", "id_str": "11362908141172611", "text": "RT @jaiselkar: #Tech Stocks are going to crush the shorts overnight and at open and the destiny is ours", "display_text_range": "11362908141172611", "entities": {"urls": [{"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}], "hashtags": [{"tag": "#AAPL"}]}, "created_at": "Tue Oct 06 23:18:51 -0000 2020", "id": "11362908141172612", "id_str": "11362908141172612", "text": "RT @jaiselkar: #Tech Stocks are going to crush the shorts overnight and at open and the destiny is ours", "display_text_range": "11362908141172612", "entities": {"urls": [{"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}], "hashtags": [{"tag": "#AAPL"}]}, "created_at": "Tue Oct 06 23:18:51 -0000 2020", "id": "11362908141172613", "id_str": "11362908141172613", "text": "RT @jaiselkar: #Tech Stocks are going to crush the shorts overnight and at open and the destiny is ours", "display_text_range": "11362908141172613", "entities": {"urls": [{"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}], "hashtags": [{"tag": "#AAPL"}]}, "created_at": "Tue Oct 06 23:18:51 -0000 2020", "id": "11362908141172614", "id_str": "11362908141172614", "text": "RT @jaiselkar: #Tech Stocks are going to crush the shorts overnight and at open and the destiny is ours", "display_text_range": "11362908141172614", "entities": {"urls": [{"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}], "hashtags": [{"tag": "#AAPL"}]}, "created_at": "Tue Oct 06 23:18:51 -0000 2020", "id": "11362908141172615", "id_str": "11362908141172615", "text": "RT @jaiselkar: #Tech Stocks are going to crush the shorts overnight and at open and the destiny is ours", "display_text_range": "11362908141172615", "entities": {"urls": [{"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}], "hashtags": [{"tag": "#AAPL"}]}, "created_at": "Tue Oct 06 23:18:51 -0000 2020", "id": "11362908141172616", "id_str": "11362908141172616", "text": "RT @jaiselkar: #Tech Stocks are going to crush the shorts overnight and at open and the destiny is ours", "display_text_range": "11362908141172616", "entities": {"urls": [{"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}, {"url": "https://t.co/1k0XfJFyv"}], "hashtags": [{"tag": "#AAPL"}]}]
```

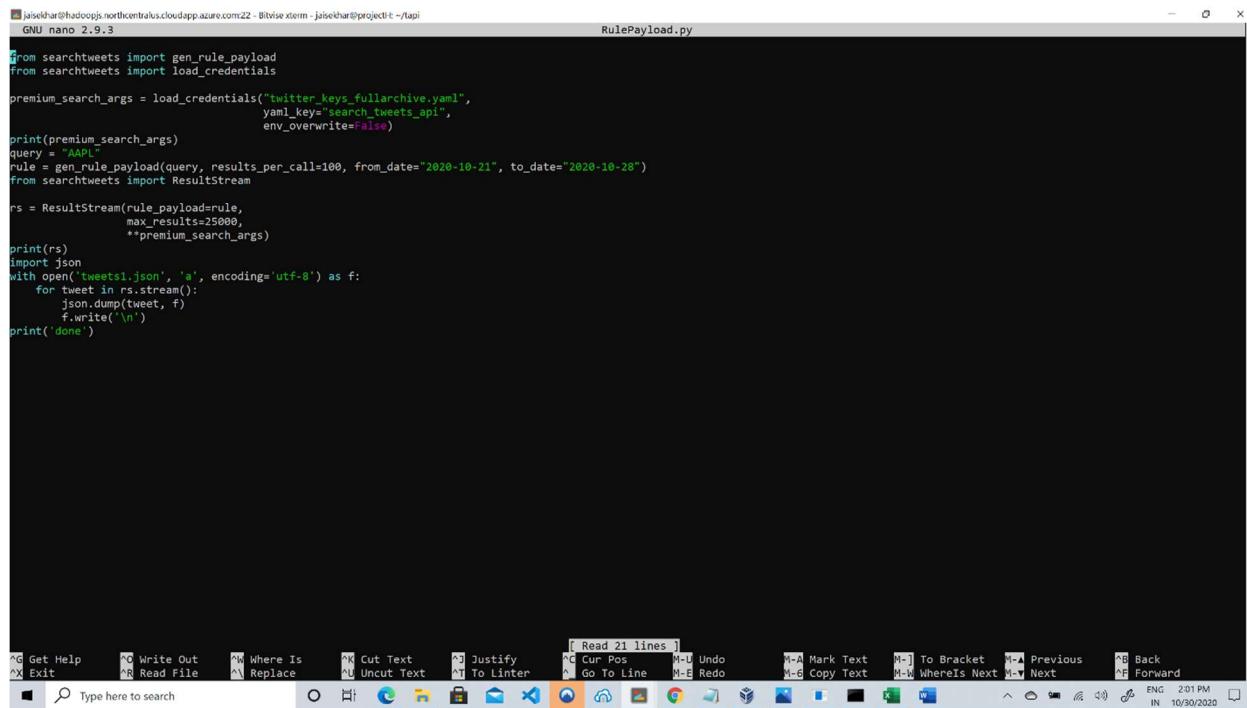
Sample JSON output (tweets1.json):

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 range. 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.



A screenshot of a terminal window titled "RulePayload.py" running on a Windows operating system. The window shows a block of Python code intended to extract tweets related to the stock symbol "AAPL". The code imports necessary modules from the searchtweets library, loads credentials from a YAML file, and defines a search rule with specific parameters like date range and result count. It then creates a ResultStream and writes the retrieved tweets to a JSON file named "tweets1.json". The terminal window has a standard Windows-style menu bar at the top and a toolbar with various file and edit functions at the bottom. The status bar at the bottom right indicates the system is in English, the time is 2:01 PM, and the date is 10/30/2020.

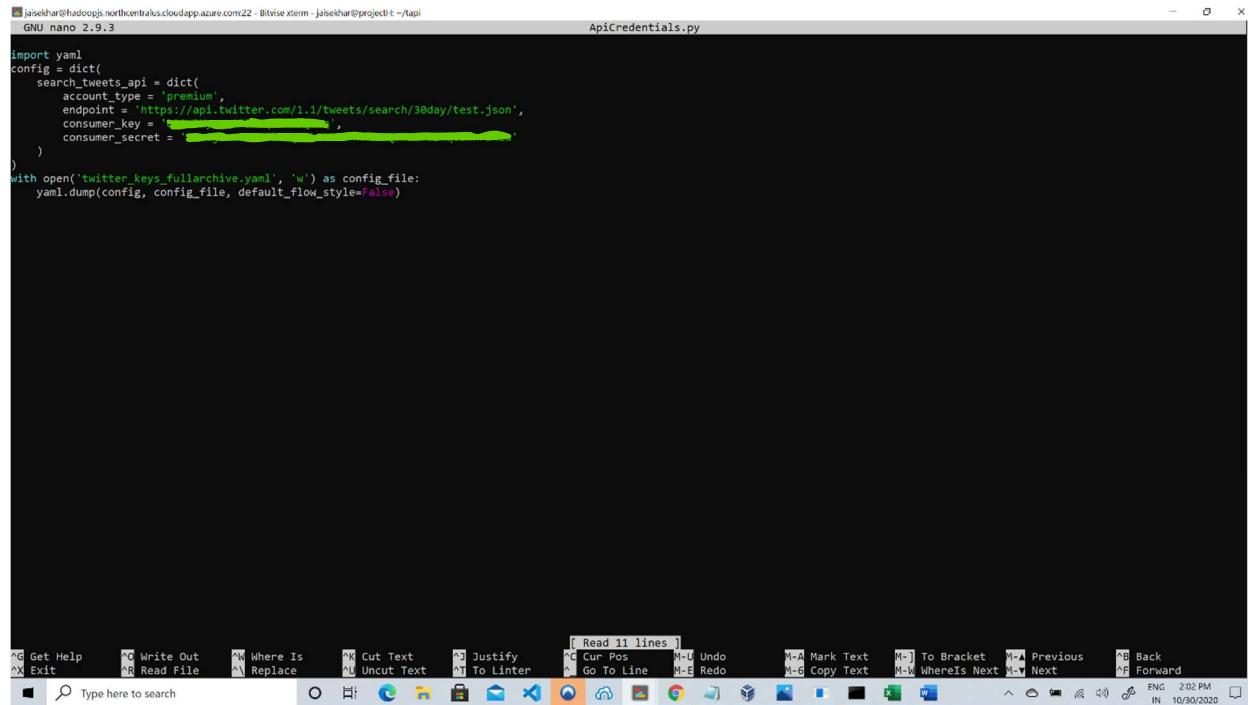
```
jaisekhar@hadoc01:~/Desktop/northcentralus.cloudapp.azure.com22 - Blvise Xterm - jaisekhar@project01: ~/tapi
GNU nano 2.9.3                                         RulePayload.py

from searchtweets import gen_rule_payload
from searchtweets import load_credentials

premium_search_args = load_credentials('twitter_keys_fullarchive.yaml',
                                       yaml_key='search_tweets_api',
                                       env_overwrite=False)
print(premium_search_args)
query = "AAPL"
rule = gen_rule_payload(query, results_per_call=100, from_date="2020-10-21", to_date="2020-10-28")
from searchtweets import ResultStream

rs = ResultStream(rule_payload=rule,
                  max_results=25000,
                  **premium_search_args)
print(rs)
import json
with open('tweets1.json', 'a', encoding='utf-8') as f:
    for tweet in rs.stream():
        json.dump(tweet, f)
        f.write('\n')
    f.write('\n')
print('done')
```

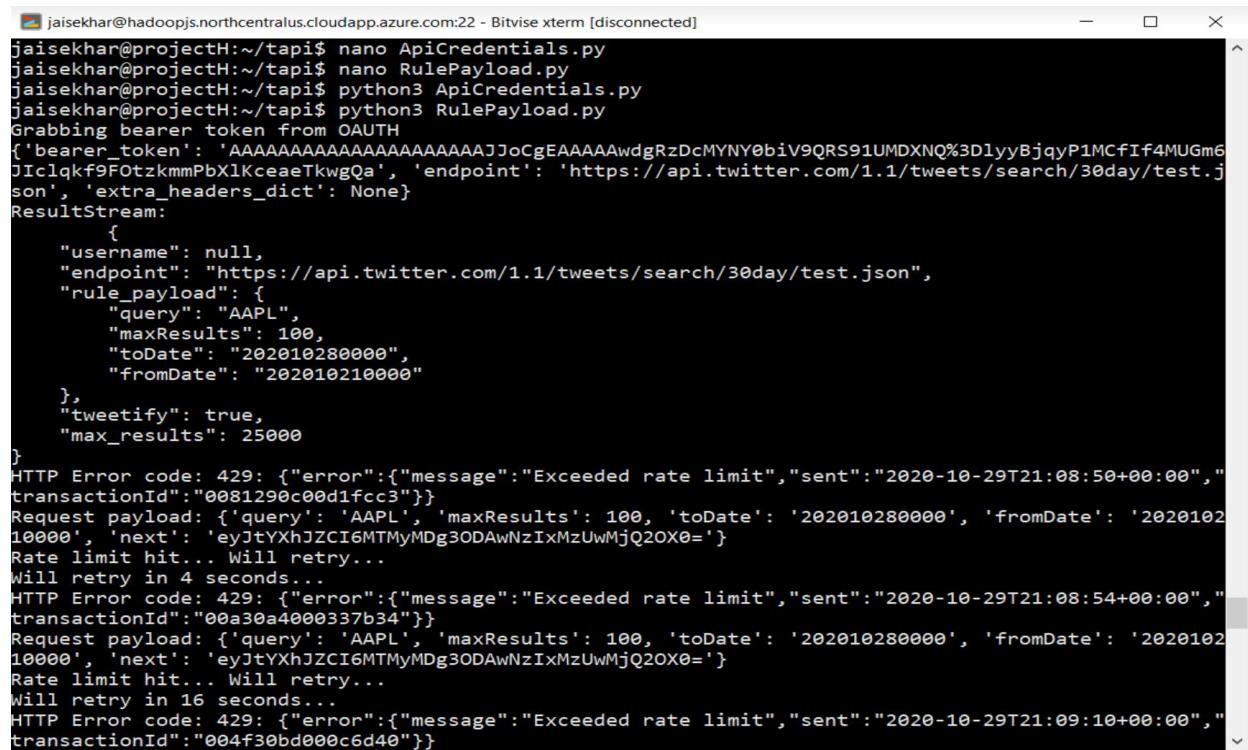
These are the credentials which are used to as the args to the searchtweets api .



```
jaisekhar@hadoopjs:~/tapi$ nano ApiCredentials.py
GNU nano 2.9.3
ApiCredentials.py

import yaml
config = dict(
    search_tweets_api = dict(
        account_type = 'premium',
        endpoint = 'https://api.twitter.com/1.1/tweets/search/30day/test.json',
        consumer_key = 'XXXXXXXXXXXXXX',
        consumer_secret = 'XXXXXXXXXXXXXX'
    )
)
with open('twitter_keys_fullarchive.yaml', 'w') as config_file:
    yaml.dump(config, config_file, default_flow_style=False)
```

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



```
jaisekhar@projectH:~/tapi$ nano ApiCredentials.py
jaisekhar@projectH:~/tapi$ nano RulePayload.py
jaisekhar@projectH:~/tapi$ python3 ApiCredentials.py
jaisekhar@projectH:~/tapi$ python3 RulePayload.py
Grabbing bearer token from OAUTH
{'bearer_token': 'AAAAAAAAAAAAAAJJoCgEAAAABwdgRzDcMYNY0biV9QRS91UMDXNQ%3DlyyBjqyP1MCFIf4MUGm6
J1c1qkf9F0tzkmmPbXlKceaeTkwgQa', 'endpoint': 'https://api.twitter.com/1.1/tweets/search/30day/test.json', 'extra_headers_dict': None}
ResultStream:
{
    "username": null,
    "endpoint": "https://api.twitter.com/1.1/tweets/search/30day/test.json",
    "rule_payload": {
        "query": "AAPL",
        "maxResults": 100,
        "toDate": "202010280000",
        "fromDate": "202010210000"
    },
    "tweetify": true,
    "max_results": 25000
}
HTTP Error code: 429: {"error":{"message":"Exceeded rate limit","sent":"2020-10-29T21:08:50+00:00","transactionId":"0081290c00d1fcc3"}}
Request payload: {'query': 'AAPL', 'maxResults': 100, 'toDate': '202010280000', ' fromDate': '202010210000', 'next': 'eyJtYXhJZCI6MTMyMDg3ODAwNzIxMzUwMjQ2OX0='}
Rate limit hit... Will retry...
Will retry in 4 seconds...
HTTP Error code: 429: {"error":{"message":"Exceeded rate limit","sent":"2020-10-29T21:08:54+00:00","transactionId":"00a30a4000337b34"}}
Request payload: {'query': 'AAPL', 'maxResults': 100, 'toDate': '202010280000', ' fromDate': '202010210000', 'next': 'eyJtYXhJZCI6MTMyMDg3ODAwNzIxMzUwMjQ2OX0='}
Rate limit hit... Will retry...
Will retry in 16 seconds...
HTTP Error code: 429: {"error":{"message":"Exceeded rate limit","sent":"2020-10-29T21:09:10+00:00","transactionId":"004f30bd000c6d40"}}
```

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.

```
jaisekhar@hadoopjs.northcentralus.cloudapp.azure.com:22 - Bitvise xterm [disconnected]
Will retry in 4 seconds...
HTTP Error code: 429: {"error":{"message":"Exceeded rate limit","sent":"2020-10-29T21:14:40+00:00","transactionId":"0034a4410023995c"}}
Request payload: {'query': 'AAPL', 'maxResults': 100, 'toDate': '202010280000', 'fromDate': '202010210000', 'next': 'eyJtYXhJZCI6MTMxOTA3MDI1MjA3Mjg4MjE3N30='}
Rate limit hit... Will retry...
Will retry in 16 seconds...
HTTP Error code: 429: {"error":{"message":"Exceeded rate limit","sent":"2020-10-29T21:14:56+00:00","transactionId":"00d68a9200e28cccd"}}
Request payload: {'query': 'AAPL', 'maxResults': 100, 'toDate': '202010280000', 'fromDate': '202010210000', 'next': 'eyJtYXhJZCI6MTMxOTA3MDI1MjA3Mjg4MjE3N30='}
Rate limit hit... Will retry...
Will retry in 36 seconds...
HTTP Error code: 429: {"error":{"message":"Exceeded rate limit","sent":"2020-10-29T21:15:47+00:00","transactionId":"004e0ca10031ca4b"}}
Request payload: {'query': 'AAPL', 'maxResults': 100, 'toDate': '202010280000', 'fromDate': '202010210000', 'next': 'eyJtYXhJZCI6MTMxODczMDIyMTI0MDAzMzI4MX0='}
Rate limit hit... Will retry...
Will retry in 4 seconds...
HTTP Error code: 429: {"error":{"message":"Exceeded rate limit","sent":"2020-10-29T21:15:51+00:00","transactionId":"0035621e003bc5d7"}}
Request payload: {'query': 'AAPL', 'maxResults': 100, 'toDate': '202010280000', 'fromDate': '202010210000', 'next': 'eyJtYXhJZCI6MTMxODczMDIyMTI0MDAzMzI4MX0='}
Rate limit hit... Will retry...
Will retry in 16 seconds...
HTTP Error code: 429: {"error":{"message":"Exceeded rate limit","sent":"2020-10-29T21:16:07+00:00","transactionId":"00b4d98600dcfd332"}}
Request payload: {'query': 'AAPL', 'maxResults': 100, 'toDate': '202010280000', 'fromDate': '202010210000', 'next': 'eyJtYXhJZCI6MTMxODczMDIyMTI0MDAzMzI4MX0='}
Rate limit hit... Will retry...
Will retry in 36 seconds...
done
jaisekhar@projectH:~/tapi$ wc -l tweets1.json
83122 tweets1.json
jaisekhar@projectH:~/tapi$
```

- **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.

```
jaisekhar@hadoopjs.northcentralus.cloudapp.azure.com:22 - Bitvise xterm [disconnected]
Last login: Thu Oct 29 23:35:40 2020 from 136.37.18.201
jaisekhar@projectH:~$ hdfs dfs -ls /bdp
Found 2 items
-rw-r--r-- 1 jaisekhar supergroup      85492 2020-10-29 23:58 /bdp/json-serde-1.3.8-jar-with-dependencies.jar
-rw-r--r-- 1 jaisekhar supergroup  457332044 2020-10-30 00:13 /bdp/tweets1.json
jaisekhar@projectH:~$ hdfs dfs -mkdir /bdp/twitter
jaisekhar@projectH:~$ hdfs dfs -cp /bdp/tweets1.json /bdp/twitter
jaisekhar@projectH:~$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/jaisekhar/hive/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/jaisekhar/hadoop-2.7.3/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
```

- Hive Operations: (initialization and Loading Data)

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

```
jaisekhar@hadoopjs.northcentralus.cloudapp.azure.com:22 - Bitvise xterm [disconnected]
Initialization script completed
schemaTool completed
jaisekhar@projectH:~/hive$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/jaisekhar/hive/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/jaisekhar/hadoop-2.7.3/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/jaisekhar/hive/lib/hive-common-2.1.0.jar!/hive-log4j2.properties Async: true
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.
hive> set hive.support.sql11.reserved.keywords=false;
```

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.

```
jaisekhar@projectH:~/hive$ cd ..
jaisekhar@projectH:~$ hdfs dfs -ls /
Found 4 items
drwxr-xr-x  - jaisekhar supergroup      0 2019-09-17 17:25 /hbase
drwxr-xr-x  - jaisekhar supergroup      0 2019-09-19 20:08 /home
drwxr-xr-x  - jaisekhar supergroup      0 2019-09-16 17:35 /jstest
drwx-----  - jaisekhar supergroup      0 2019-09-10 20:20 /tmp
jaisekhar@projectH:~$ hdfs dfs -ls -mkdir /bdp
-ls: Illegal option -mkdir
Usage: hadoop fs [generic options] -ls [-d] [-h] [-R] [<path> ...]
jaisekhar@projectH:~$ hdfs dfs -mkdir /bdp
jaisekhar@projectH:~$ hdfs dfs -copyFromLocal json-serde-1.3.8-jar-with-dependencies.jar /bdp
jaisekhar@projectH:~$ hive
```

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.

```
jaisekhar@hadoopjs.northcentralus.cloudapp.azure.com:22 - Bitvise xterm [disconnected]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/jaisekhar/hive/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/jaisekhar/hadoop-2.7.3/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/jaisekhar/hive/lib/hive-common-2.1.0.jar!/hive-log4j2.properties Async: true
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.
hive> set hive.support.sql11.reserved.keywords=false;
hive> CREATE EXTERNAL TABLE tweets_raw (
    >     id BIGINT,
    >     created_at STRING,
    >     source STRING,
    >     favorited BOOLEAN,
    >     retweet_count INT,
    >    retweeted_status STRUCT<
    >         text:STRING,
    >         user:STRUCT<screen_name:STRING,name:STRING>,
    >     entities STRUCT<
    >         urls:ARRAY<STRUCT<expanded_url:STRING>>,
    >         user_mentions:ARRAY<STRUCT<screen_name:STRING,name:STRING>>,
    >         hashtags:ARRAY<STRUCT<text:STRING>>>,
    >     text STRING,
    >     user STRUCT<
    >         screen_name:STRING,
    >         name:STRING,
    >         friends_count:INT,
    >         followers_count:INT,
    >         statuses_count:INT,
    >         verified:BOOLEAN,
    >         utc_offset:STRING, -- was INT but nulls are strings
    >         time_zone:STRING>,
    >     in_reply_to_screen_name STRING,
    >     year int,
    >     month int,
    >     day int,
    >     hour int
    > )
    > ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
    > LOCATION '/bdp/twitter'
    > ;
```

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

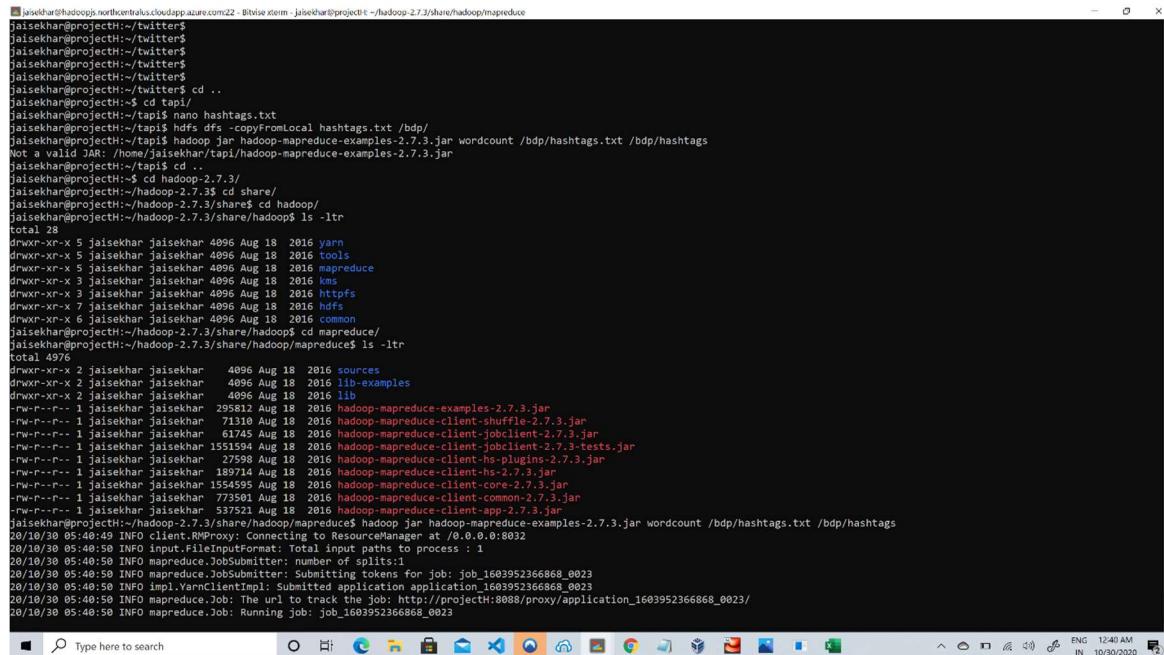
The python code is executed and saved the output in hashtags.txt using out operator.

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 output in /bdp/hashtags directory in hdfs.

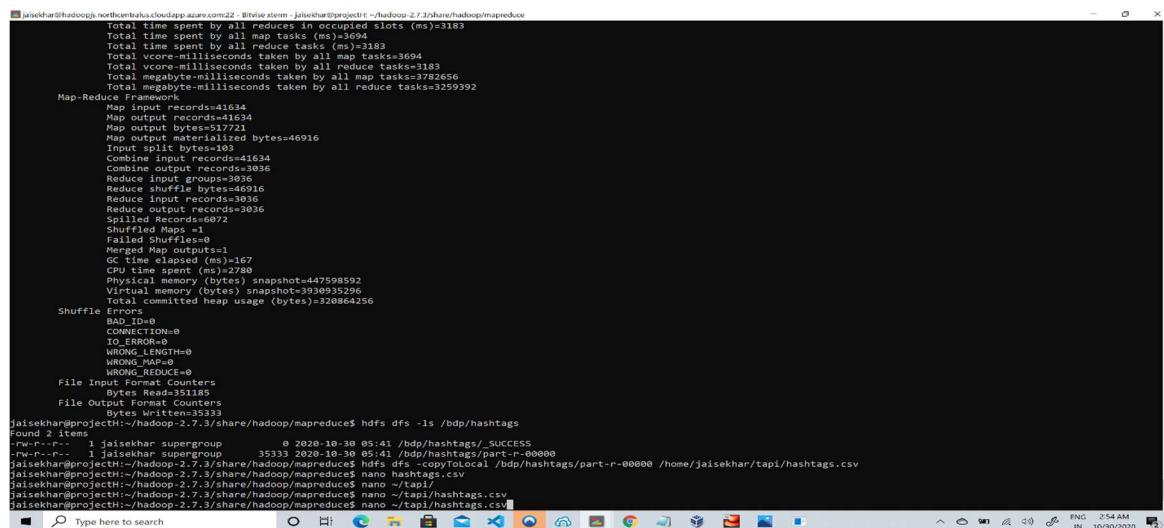
```
hadoop jar hadoop-mapreduce-examples-2.7.3.jar wordcount /bdp/hashtags.txt /bdp/hashtags
```



```
jaisekhar@projectH:/tapi$ hadoop jar hadoop-mapreduce-examples-2.7.3.jar wordcount /bdp/hashtags.txt /bdp/hashtags
20/10/30 05:40:50 INFO mapreduce.Job: Running job: job_160395236686_0023
20/10/30 05:40:50 INFO mapreduce.Job:  map 0% reduce 0%
20/10/30 05:40:50 INFO mapreduce.Job: Job complete: job_160395236686_0023
20/10/30 05:40:50 INFO mapreduce.Job: Output Compressed: false
20/10/30 05:40:50 INFO mapreduce.Job: RecordWriter: org.apache.hadoop.mapred.TextRecordWriter
20/10/30 05:40:50 INFO mapreduce.Job: Map output collector class: org.apache.hadoop.mapred.MapOutputCollector
20/10/30 05:40:50 INFO mapreduce.Job: Number of splits: 1
20/10/30 05:40:50 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_160395236686_0023
20/10/30 05:40:50 INFO mapreduce.JobSubmitter: Submitted tokens for job: job_160395236686_0023
20/10/30 05:40:50 INFO mapreduce.Job: The url to track the job: http://projectH:8088/proxy/application_160395236686_0023/
20/10/30 05:40:50 INFO mapreduce.Job: Running job: job_160395236686_0023
```

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
```



```
jaisekhar@projectH:/tapi$ hdfs dfs -ls /bdp/hashtags
Found 2 items
-rw-r--r-- 1 jaisekhar supergroup 35333 2020-10-30 05:41 /bdp/hashtags/part-r-00000
-rw-r--r-- 1 jaisekhar supergroup 35333 2020-10-30 05:41 /bdp/hashtags/part-r-00001
jaisekhar@projectH:/tapi$ hdfs dfs -copyToLocal /bdp/hashtags/part-r-00000 /home/jaisekhar/tapi/hashtags.csv
jaisekhar@projectH:/tapi$ hdfs dfs -copyToLocal /bdp/hashtags/part-r-00001 /home/jaisekhar/tapi/hashtags.csv
jaisekhar@projectH:/tapi$ nano /tapi/hashtags.csv
jaisekhar@projectH:/tapi$ nano -r /tapi/hashtags.csv
jaisekhar@projectH:/tapi$ nano -r /tapi/hashtags.csv
```

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

```
jaisekhar@hadoop5:~$ hadoop jar /home/jaisekhar/tapi/wordcount.jar wordcount /tmp/tapi/hashtags /tmp/tapi/hashtags.out
150RB 12
1V 1
1ab 1
2020election 1
20Kto10KChallenge 1
20Kto25K 2
20Kto10kChallenge 2
20代投資家と繋がりたい 1
25sep8aaje25Minute 2
25万円以下で売却 1
2BTRADERS 1
trillion 1
00RB 4
01k 1
4chan 1
5726A 1
5G 360
5iPhone 3
5g 5
5gNetwork 2
5gfuture 1
65club 7
9MileHighInTech 1
737max 1
99superShoppingDay 1
Dat9 8
A 1
A14 1
A14BIONIC 1
AAP 1
AAPI 2
AAPL 1418
AAPLF 2
AAPLishestockmarket 1
AAPLyousuck 1
AAPL万歳 1
AAPL 3
ABC 1
ABD 50
ACBconfirmation 2
ACT 9
ACTFamily 1
ADBE 2
[ Read 3036 lines ]
M-G Get Help M-W Write Out M-A Where Is M-C Cut Text M-U Undo M-A Mark Text M-M To Bracket M-V Previous M-B Back
M-X Exit M-R Read File M-N Replace M-U Uncut Text M-E Redo M-W WhereIs Next M-V Next M-F Forward
Type here to search
```

• 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.

```
jaisekhar@hadoop5:~$ hive -e 'create table split_words as select id as id,split(text,' ') as words from tweets_raw;
WARNING: Hive-on-HDFS 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_20201030034630_1d4f2ea6-07d9-a6a9-078d2c45380e
Total tasks : 1
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1603952366868_0006, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0006/
Kill Command = /home/jaisekhar/hadoop/bin/hadoop job -kill job_1603952366868_0006
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2020-10-30 03:46:40,543 Stage-1 map = 0%, reduce = 0%
2020-10-30 03:47:03,296 Stage-1 map = 36%, reduce = 0%, Cumulative CPU 27.72 sec
2020-10-30 03:47:03,514 Stage-1 map = 75%, reduce = 0%, Cumulative CPU 32.68 sec
2020-10-30 03:47:03,751 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 37.79 sec
MapReduce Total cumulative CPU time: 37 seconds 730 msec
Ended Job = job_1603952366868_0006
Stage-4 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Launching Job 3 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1603952366868_0007, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0007/
Kill Command = /home/jaisekhar/hadoop/bin/hadoop job -kill job_1603952366868_0007
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2020-10-30 03:47:29,759 Stage-3 map = 0%, reduce = 0%
2020-10-30 03:47:37,421 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 5.04 sec
MapReduce Total cumulative CPU time: 5 seconds 40 msec
Ended Job = job_1603952366868_0007
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/split_words
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Cumulative CPU: 37.73 sec  HDFS Read: 457361250 HDFS Write: 11972221 SUCCESS
Stage-Stage-3: Map: 1  Cumulative CPU: 5.04 sec  HDFS Read: 1974663 HDFS Write: 9918572 SUCCESS
Total MapReduce CPU Time Spent: 42 seconds 770 msec
OK
Time taken: 69.486 seconds
Type here to search
```

The split words in split_words table will look like below

```
jaisekhar@hadoop01:~$ hadoop fs -ls /user/hive/warehouse/split_words
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1603952366868_0006, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0006/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0006
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 0
2020-10-30 03:46:03,543 Stage-1 map = 0%, reduce = 0%
2020-10-30 03:47:03,296 Stage-1 map = 36%, reduce = 0%, Cumulative CPU 27.72 sec
2020-10-30 03:47:06,514 Stage-1 map = 75%, reduce = 0%, Cumulative CPU 32.68 sec
2020-10-30 03:47:15,311 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 37.73 sec
MapReduce Total cumulative CPU time: 37 seconds 730 msec
Ended Job = job_1603952366868_0006
Stage-4 is filtered out by condition resolver.
Stage-3 is selected by condition resolver.
Stage-5 is filtered out by condition resolver.
Launching Job 3 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1603952366868_0007, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0007/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0007
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2020-10-30 03:47:29,759 Stage-3 map = 0%, reduce = 0%
2020-10-30 03:47:31,421 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 5.04 sec
MapReduce total cumulative CPU time: 5 seconds 40 msec
Ended Job = job_1603952366868_0007
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/split_words
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Cumulative CPU: 37.73 sec HDFS Read: 457361250 HDFS Write: 11972221 SUCCESS
Stage-Stage-3: Map: 1 Cumulative CPU: 5.04 sec HDFS Read: 11974063 HDFS Write: 9918572 SUCCESS
Total MapReduce CPU Time Spent: 42 seconds 770 msec
OK
Time taken: 69.486 seconds
hive> describe split_words;
OK
id          bigint
words        array<string>
Time taken: 0.074 seconds, Fetched: 2 row(s)
hive> select *from split_words limit 5;
OK
1313630114157010945  ["@EyoeftheStormZ", "I", "got", "some", "120c", "for", "10/23", "today", "I", "think", "people", "are", "hoping", "aapl", "goes", "", "up", "and", "it", "did", "look", "strong", "before", "the", "dump", "today"]
1313629620298747905  ["RT", "@StockHighAlertz:", "$OPTI", "up", "+18.75%", "OPTEC", "International", "Inc.", "(OPTI)", "to", "Announce", "Q1", "Interim", "Financial", "Statement", "Pre-Market", "Wednesday"]
1313629506297421284  ["$AAPL", "lots", "of", "option", "action", "https://t.co/S1kbk8pASS"]
1313629440488869888  ["RT", "@stock_goodies:", "$TPIN", "@thomasSci", "the", "third", "largest", "scientific", "distributor", "in", "the", "nation", "about", "to", "launch", "sales", "of", "QuikLAB", "and", "SANIQuik", "$"]
1313629305621155840  ["Apple", "iPhone", "launch", "event", "coming", "October", "13:", "What", "to", "expect", "$AAPL", "https://t.co/SL2nrg05Tu"]
Time taken: 0.101 seconds, Fetched: 5 row(s)
hive>
```

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.

```
jaisekhar@hadoop01:~$ hadoop fs -ls /user/hive/warehouse/tweet_word
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1603952366868_0008, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0008/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0008
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2020-10-30 03:49:47,469 Stage-1 map = 0%, reduce = 0%
2020-10-30 03:49:54,984 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 3.94 sec
MapReduce Total cumulative CPU time: 3 seconds 940 msec
Ended Job = job_1603952366868_0008
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/_hive-staging_hive_2020-10-30_03-49-48_342_4677166674228850463-1-ext-10001
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/tweet_word
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Cumulative CPU: 3.94 sec HDFS Read: 9923285 HDFS Write: 29791380 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 940 msec
OK
Time taken: 15.885 seconds
hive> describe tweet_word;
OK
id          bigint
word        string
Time taken: 0.048 seconds, Fetched: 2 row(s)
hive> select *from tweet_word limit 5;
OK
1313630114157010945  @EyoeftheStormZ
1313630114157010945  I
1313630114157010945  got
1313630114157010945  some
1313630114157010945  120c
Time taken: 0.107 seconds, Fetched: 5 row(s)
hive>
```

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 shown below.

```
jaisekhar@hadoop3.northcentralus.cloudapp.azure.com:22 - Bltwise xterm - jaisekhar@project:t ~/hive
Ended Job = job_1603952366868_0008
Stage-4 is selected by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/.hive-staging_hive_2020-10-30_03-49-40_342_4677166674228850463-1/-ext-10001
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/tweet_word
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Cumulative CPU: 3.94 sec HDFS Read: 9923285 HDFS Write: 29791380 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 940 msec
OK
Time taken: 15.885 seconds
hive> describe tweet_word;
OK
id          bigint
word        string
Time taken: 0.048 seconds, Fetched: 2 row(s)
hive> select *from tweet_word limit 5;
OK
1313630114157810945 @EyeoftheStormZ
1313630114157810945 I
1313630114157810945 got
1313630114157810945 some
1313630114157810945 128c
Time taken: 0.107 seconds, Fetched: 5 row(s)
hive> create table dictionary(word string,rating int) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
OK
Time taken: 0.062 seconds
hive> LOAD DATA INPATH '/AFINN.txt' into TABLE dictionary;
FAILED: SemanticException Line 1:17 Invalid path ''/AFINN.txt'': No files matching path hdfs://localhost:9000/AFINN.txt
hive> LOAD DATA LOCAL INPATH '/AFINN.txt' into TABLE dictionary;
FAILED: SemanticException Line 1:23 Invalid path ''/AFINN.txt'': No files matching path file:/AFINN.txt
hive> LOAD DATA LOCAL INPATH '~/tapi/AFINN.txt' into TABLE dictionary;
FAILED: SemanticException Line 1:23 Invalid path '~/tapi/AFINN.txt'': No files matching path file:/home/jaisekhar/hive/~/tapi/AFINN.txt
hive> LOAD DATA LOCAL INPATH '../tapi/AFINN.txt' into TABLE dictionary;
Loading data to table default.dictionary
OK
Time taken: 0.308 seconds
hive> select *from di
directory    disable    distinct    distribute    div(
hive> select *from dictionary limit 5;
OK
abandon -2
abandoned -2
abandons -2
abducted -2
abduction -2
Time taken: 0.084 seconds, Fetched: 5 row(s)
hive>
```

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

```
jaisekhar@hadoop3.northcentralus.cloudapp.azure.com:22 - Bltwise xterm - jaisekhar@project:t ~/hive
FAILED: SemanticException Line 1:23 Invalid path ''~/tapi/AFINN.txt'': No files matching path file:/home/jaisekhar/hive/~/tapi/AFINN.txt
hive> LOAD DATA LOCAL INPATH '../tapi/AFINN.txt' into TABLE dictionary;
Loading data to table default.dictionary
OK
Time taken: 0.308 seconds
hive> select *from di
directory    disable    distinct    distribute    div(
hive> select *from dictionary limit 5;
OK
abandon -2
abandoned -2
abandons -2
abducted -2
abduction -2
Time taken: 0.084 seconds, Fetched: 5 row(s)
hive> create table word_join as select tweet_word.id,tweet_word.word,dictionary.rating from tweet_word LEFT OUTER JOIN dictionary ON(tweet_word.word =dictionary.word);
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_20201030035549_36795561-d71-4285-8fd1-2dbbf3cf3348
Total jobs = 1
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/jaisekhar/hive/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/jaisekhar/hadoop-2.7.3/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
2020-10-30 03:55:14 Starting to process local tasks to process map join; maximum memory = 477626368
2020-10-30 03:55:16 Dump the reduce-table for tag: 1 with group count: 2477 into file: file:/tmp/jaisekhar/1219b18d-06d9-4391-8622-9e6e48671738/hive_2020-10-30_03-55-49_214_727643782734096
4210_1-local_10004/HashTable-Stage-4/MapJoin-mapfile01--hashtable
2020-10-30 03:55:17 Uploaded 1 File to: file:/tmp/jaisekhar/1219b18d-06d9-4391-8622-9e6e48671738/hive_2020-10-30_03-55-49_214_727643782734096210-1-local-10004/HashTable-Stage-4/MapJoin-mapfile01--hashtable (69200 bytes)
2020-10-30 03:55:17 End of local task; Time Taken: 1.193 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1603952366868_0009, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0009/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0009
Hadoop job information for Stage-4: number of mappers: 1; number of reducers: 0
2020-10-30 03:56:04.874 Stage-4 map = 0%, reduce = 0%
2020-10-30 03:56:15.656 Stage-4 map = 100%, reduce = 0%, Cumulative CPU 8.41 sec
MapReduce Total cumulative CPU time: 8 seconds 410 msec
Ended Job = job_1603952366868_0009
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/word_join
MapReduce Jobs Launched:
Stage-Stage-4: Map: 1 Cumulative CPU: 8.41 sec HDFS Read: 29797484 HDFS Write: 33104055 SUCCESS
Total MapReduce CPU Time Spent: 8 seconds 410 msec
OK
Time taken: 27.641 seconds
hive>
```

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

```
jaisekhar@hadoop3:~$ hive
hive> select id,AVG(rating) as rating from word_join GROUP BY word_join.id order by rating DESC 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_20201030035831_b1683b39-dfce-45a3-9d6c-39e41e47cc55
Total jobs = 2
Launching Job 1 out of 2
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_0010, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0010/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0010
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-10-30 03:58:39,826 Stage-1 map = 0%, reduce = 0%
2020-10-30 03:58:49,726 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 9.09 sec
2020-10-30 03:58:58,235 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 14.14 sec
MapReduce Total cumulative CPU time: 14 seconds 140 msec
Ended Job = job_1603952366868_0010
Launching Job 2 out of 2
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_0011, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0011/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0011
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2020-10-30 03:59:12,128 Stage-2 map = 0%, reduce = 0%
2020-10-30 03:59:16,652 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 4.17 sec
2020-10-30 03:59:26,045 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 6.46 sec
MapReduce Total cumulative CPU time: 6 seconds 460 msec
Ended Job = job_1603952366868_0011
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 14.14 sec HDFS Read: 33112130 HDFS Write: 2333576 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 6.46 sec HDFS Read: 2339005 HDFS Write: 447 SUCCESS
Total MapReduce CPU Time Spent: 20 seconds 600 msec
OK
1320066591153938438 5.0
1318279840190005248 5.0
1318000560465903616 5.0
1315042659937984520 5.0
1315279419041345537 5.0
1320106069327302664 5.0
1312466814400565248 5.0
13115451109585241600 4.0
1311544751712137217 4.0
131164499275538691 4.0
Time taken: 55.286 seconds, Fetched: 10 row(s)
hive> create table testjoin as select id,AVG(rating) as polarity from word_join GROUP BY word_join.id;
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_20201030040157_l5624f5e-f8ce-47fe-a731-b4ffe5253835
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_0012, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0012/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0012
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-10-30 04:02:04,689 Stage-1 map = 0%, reduce = 0%
2020-10-30 04:02:15,310 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 9.12 sec
2020-10-30 04:02:23,908 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 13.73 sec
MapReduce Total cumulative CPU time: 13 seconds 730 msec
Ended Job = job_1603952366868_0012
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/testjoin
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 13.73 sec HDFS Read: 33112656 HDFS Write: 1944460 SUCCESS
Total MapReduce CPU Time Spent: 13 seconds 730 msec
OK
Time taken: 28.087 seconds
hive>
```

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

```
jaisekhar@hadoop3:~$ hive
hive> select id,AVG(rating) as rating from word_join GROUP BY word_join.id order by rating DESC 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_20201030035831_b1683b39-dfce-45a3-9d6c-39e41e47cc55
Total jobs = 2
Launching Job 1 out of 2
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_0010, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0010/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0010
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-10-30 03:59:12,128 Stage-1 map = 0%, reduce = 0%
2020-10-30 03:59:19,652 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 4.17 sec
2020-10-30 03:59:26,045 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 6.46 sec
MapReduce Total cumulative CPU time: 6 seconds 460 msec
Ended Job = job_1603952366868_0010
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 14.14 sec HDFS Read: 33112130 HDFS Write: 2333576 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 6.46 sec HDFS Read: 2339005 HDFS Write: 447 SUCCESS
Total MapReduce CPU Time Spent: 20 seconds 600 msec
OK
1320066591153938438 5.0
1318279840190005248 5.0
1318000560465903616 5.0
1315042659937984520 5.0
1315279419041345537 5.0
1320106069327302664 5.0
1312466814400565248 5.0
13115451109585241600 4.0
1311544751712137217 4.0
131164499275538691 4.0
Time taken: 55.286 seconds, Fetched: 10 row(s)
hive> create table testjoin as select id,AVG(rating) as polarity from word_join GROUP BY word_join.id;
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_20201030040157_l5624f5e-f8ce-47fe-a731-b4ffe5253835
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_0012, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0012/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0012
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-10-30 04:02:04,689 Stage-1 map = 0%, reduce = 0%
2020-10-30 04:02:15,310 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 9.12 sec
2020-10-30 04:02:23,908 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 13.73 sec
MapReduce Total cumulative CPU time: 13 seconds 730 msec
Ended Job = job_1603952366868_0012
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/testjoin
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 13.73 sec HDFS Read: 33112656 HDFS Write: 1944460 SUCCESS
Total MapReduce CPU Time Spent: 13 seconds 730 msec
OK
Time taken: 28.087 seconds
hive>
```

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

```

hive> create table tweets as select tweets_raw.* ,nvl(testjoin.polarity,0) as polarity from tweets_raw LEFT OUTER JOIN testjoin ON(tweets_raw.id =testjoin.id);
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_2020103004201_610f07f8-be5f-4c87-b4b2-9a364ba5d51
Total jobs = 1
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/jaisekhar/hive/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/jaisekhar/hadoop-common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
2020-10-30 04:42:08 Starting to launch local task to process map join; maximum memory = 477626368
2020-10-30 04:42:10 Dump the side-table for tag: 1 with group count: 83122 into file: file:/tmp/jaisekhar/1219b18d-06d9-4391-8622-9e6e48671738/hive_2020-10-30_04-42-01_928_13949070656869
72736-1-local-10004/HashTable-Stage-4/MapJoin-mapfile31--.hashtable
2020-10-30 04:42:10 Uploaded 1 file to: file:/tmp/jaisekhar/1219b18d-06d9-4391-8622-9e6e48671738/hive_2020-10-30_04-42-01_928_1394907065686972736-1-local-10004/HashTable-Stage-4/MapJoin-mapfile31--.hashtable (2405610 bytes)
2020-10-30 04:42:10 End of local task; Time Taken: 1.906 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1603952366868_0018, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0018/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0018
Hadoop Job Information for Stage-4: number of mappers: 2; number of reducers: 0
2020-10-30 04:42:10 Stage-4 map = 0%, reduce = 0%
2020-10-30 04:42:41 884 Stage-4 map = 36%, reduce = 0%, Cumulative CPU 27.81 sec
2020-10-30 04:42:45 920 Stage-4 map = 75%, reduce = 0%, Cumulative CPU 31.14 sec
2020-10-30 04:42:49 216 Stage-4 map = 100%, reduce = 0%, Cumulative CPU 34.91 sec
MapReduce Total cumulative CPU time: 34 seconds 910 msec
Ended Job = job_1603952366868_0018
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/tweets
MapReduce Jobs Launched:
Stage-Stage-4: Map: 2 Cumulative CPU: 34.91 sec HDFS Read: 457377858 HDFS Write: 36222244 SUCCESS
Total MapReduce CPU Time Spent: 34 seconds 910 msec
OK
Time taken: 48.472 seconds
hive> select * from tweets limit 10;
OK
1313630114157018945 Tue Oct 06 23:59:59 +0000 2020 <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a> false 0 NULL {"urls":[],"user_mentions":[{"screen_name":"EyeoftheStormZ","name":"cr8ig"},{}], "hashtags":[]}&#xA0; @EyeoftheStormZ I got some 120c for 10/23 today, I think people are "hoping" aapl goes up and it did look strong before the dump today {"screen_name":"house_money","name":"House Money","friends_count":583,"followers_count":121,"statuses_count":1397,"verified":false,"utc_offset":null,"time_zone":null}
} EyeoftheStormZ NULL NULL NULL NULL 0.5
1313629520298747905 Tue Oct 06 23:58:02 +0000 2020 <a href="http://twitter.com/#!/download/ipad" rel="nofollow">Twitter for iPad</a> false 0 {"text":">$OPTI up +18.75% OPTEC International, Inc. (OPTI) to Announce Q1 Interim Financial Statement Pre-Market Wednesday 0.. https://t.co/czegEdBhwX","user":{"screen_name":"StockHighAlertz","name":"Stocks On HIGH ALERTZ!"}}, {"urls":[],"user_mentions":[{"screen_name":"StockHighAlertz","name":"Stocks On HIGH ALERTZ!"}], "hashtags":[]}&#xA0; RT @StockHighAlertz: $OPTI up +18.75% OPTEC International, Inc. (OPTI) to Announce Q1 Interim Financial Statement Pre-Market Wednesday Octo... {"screen_name":"Investoon","name":"Broke Investor","friends_count":1988,"followers_count":404,"statuses_count":1597,"verified":false,"utc_offset":null,"time_zone":null}
NULL NULL NULL NULL NULL 0.0
1313629506297421824 Tue Oct 06 23:57:35 +0000 2020 <a href="https://mobile.twitter.com" rel="nofollow">Twitter Web App</a> false 0 NULL {"urls":[],"user_mentions":[]}, "hashtags":[]

Type here to search

```

Below, we can 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-d6c4-49ea-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=<nnumber>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<nnumber>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<nnumber>
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 @ $1.95 with some profits from earlier. Holding overnight going to risk it for a gap.. https://t.co/imjIWbod0D -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/p2pu1TSPqm 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/oV7yQnubE -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/oZuunz58XZ 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/1bfuI95mtF -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/bhglsSwe40 -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/TsjxMsnnNs 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/tknf28QrL 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;
```

```
jaisekhar@hadoop-northcentralus.cloudapp.azure.com:22 Bitwise xterm -jaisekhar@project1: ~/hive
SUBCOMMITTEE RELEASES 449 PAGE REPORT DETAILING MARKET POWER ABUSE BY $AAPL, $AMZN, $GOOGL AND $FB, "user": {"screen_name": "DeItaone", "name": "Walter Bloomberg"}, {"urls": [], "user_mentions": [{"screen_name": "DeItaone", "name": "Walter Bloomberg"}, {"hashtags": []}], "RT @DeItaOne: U.S. HOUSE ANTITRUST SUBCOMMITTEE RELEASES 449 PAGE REPORT DETAILING MARKET POWER ABUSE BY $AAPL, $AMZN, $GOOGL AND $FB", "user": {"screen_name": "MichaelJobe", "name": "Michael Jobe"}, "friends_count": 69, "followers_count": 388, "statuses_count": 8776, "verified": false, "utc_offset": null, "time_zone": null}
NULL NULL NULL NULL NULL 0.0
1313629074179276808 Tue Oct 23:55:51 +0000 2020 <a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a> false 0 {"text": "HOUSE DEMOCRATS ACCU
SE FACEBOOK, AMAZON, ALPHABET, APPLE OF HAVING 'MONOPOLY POWER' AND RECOMMEND BIG CHANGES", "user": null} NULL NULL
Time taken: 0.073 seconds, Fetched: 10 row(s)
hive> select polarity from tweets limit 10;
OK
0.5
0.0
0.0
0.0
0.0
0.0
-2.0
0.0
NULL
NULL
Time taken: 0.078 seconds, Fetched: 10 row(s)
hive> select 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 engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID : jaisekhar_2020103004400_5bb3ce57-3dac-4b66-9642-be46a750c166
Total jobs = 1
Launching Job 1 out of 1
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_0019, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0019/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0019
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-10-30 04:44:08,126 Stage-1 map = 0%, reduce = 0%
2020-10-30 04:44:15,630 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 3.67 sec
2020-10-30 04:44:22,013 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.92 sec
MapReduce Total cumulative CPU time: 5 seconds 920 msec
Ended Job = job_1603952366868_0019
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 5.92 sec HDFS Read: 36235254 HDFS Write: 106 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 920 msec
OK
238807
Time taken: 23.847 seconds, Fetched: 1 row(s)
hive> 
```

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.

```
jaisekhar@hadoop-northcentralus.cloudapp.azure.com:22 Bitwise xterm -jaisekhar@project1: ~/hive
hive> select distinct user.screen_name as name , user.followers_count as count
> from tweets
> where size(entities.hashtags) > 0
> order by count desc
> limit 5;
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_20201030054226_12485fee-aed1-45c4-ab67-778aba9b3e97
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_0024, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0024/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0024
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-10-30 05:42:24,491 Stage-1 map = 0%, reduce = 0%
2020-10-30 05:42:41,951 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 4.92 sec
2020-10-30 05:42:49,511 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 7.86 sec
MapReduce Total cumulative CPU time: 7 seconds 860 msec
Ended Job = job_1603952366868_0024
Launching Job 2 out of 2
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_0025, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0025/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0025
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2020-10-30 05:43:10,598 Stage-2 map = 0%, reduce = 0%
2020-10-30 05:43:17,052 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 1.44 sec
2020-10-30 05:43:17,052 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 3.93 sec
MapReduce Total cumulative CPU time: 3 seconds 930 msec
Ended Job = job_1603952366868_0025
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 7.86 sec HDFS Read: 36235774 HDFS Write: 87747 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 3.93 sec HDFS Read: 93131 HDFS Write: 232 SUCCESS
Total MapReduce CPU Time Spent: 11 seconds 798 msec
OK
Apple 5462984
TheStreet 736435
Nasdaq 608042
hive> 
```

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.

```
jaisekhar@hadoop-northcentralus.cloudapp.azure.com:22 ~$hive xterm -jaisekhar@project01t ~/hive
hive> SELECT user.name, user.screen_name, count(1) as cc
   > FROM tweets
   > WHERE text like "%AAPL%" and user.name is not null
   > GROUP BY user.name,user.screen_name
   > ORDER BY cc desc limit 5;
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_20201030054655_54084153-d322-45d4-a2ef-ba68b25a3df5
Total jobs = 2
Launching Job 1 out of 2
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_0028, Tracking URL = http://project01t:8088/proxy/application_1603952366868_0028/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0028
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-10-30 05:47:17,480 Stage-1 map = 0%, reduce = 0%
2020-10-30 05:47:17,480 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 4.56 sec
2020-10-30 05:47:17,480 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 7.72 sec
MapReduce Total cumulative CPU time: 7 seconds 729 msec
Ended Job = job_1603952366868_0028
Launching Job 2 out of 2
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_0029, Tracking URL = http://project01t:8088/proxy/application_1603952366868_0029/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0029
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2020-10-30 05:47:37,636 Stage-2 map = 0%, reduce = 0%
2020-10-30 05:47:37,636 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 2.43 sec
2020-10-30 05:47:44,061 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 4.77 sec
MapReduce Total cumulative CPU time: 4 seconds 770 msec
Ended Job = job_1603952366868_0029
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 7.72 sec HDFS Read: 36236574 HDFS Write: 520124 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 4.77 sec HDFS Read: 525896 HDFS Write: 384 SUCCESS
Total MapReduce CPU Time Spent: 12 seconds 490 msec
OK
TIN-Tech Bloggers          TINTechBloggers    477
FinBuzz PortfolioBuzz      442
Investor News newsfilterio  435

```

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.

```
jaisekhar@hadoop-northcentralus.cloudapp.azure.com:22 ~$hive xterm -jaisekhar@project01t ~/hive
2020-10-30 05:56:05,185 Stage-1 map = 0%, reduce = 0%
2020-10-30 05:56:13,811 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 4.8 sec
2020-10-30 05:56:21,222 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 7.07 sec
MapReduce Total cumulative CPU time: 7 seconds 70 msec
Ended Job = job_1603952366868_0030
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 7.07 sec HDFS Read: 36236027 HDFS Write: 101 SUCCESS
Total MapReduce CPU Time Spent: 7 seconds 70 msec
OK
7
Time taken: 23.902 seconds. Fetched: 1 row(s)
hive> select count(*) from tweets where text like '%iphone%' or text like '%iphone%' and polarity > 0.0;
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_20201030055719_f1d887d9-fe6f-428f-b901-578898c97a34
Total jobs = 1
Launching Job 1 out of 1
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_0031, Tracking URL = http://project01t:8088/proxy/application_1603952366868_0031/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0031
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-10-30 05:57:27,002 Stage-1 map = 0%, reduce = 0%
2020-10-30 05:57:34,389 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 4.16 sec
2020-10-30 05:57:41,839 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 7.0 sec
MapReduce Total cumulative CPU time: 7 seconds 0 msec
Ended Job = job_1603952366868_0031
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 7.0 sec HDFS Read: 36236745 HDFS Write: 103 SUCCESS
Total MapReduce CPU Time Spent: 7 seconds 0 msec
OK
234
Time taken: 23.217 seconds. Fetched: 1 row(s)
hive> select count(*) from tweets where text like '%event%' or '%Event%' and polarity < 0.0;
FAILED: ClassCastException org.apache.hadoop.hive.serde2.objectinspector.primitive.WritableConstantStringObjectInspector cannot be cast to org.apache.hadoop.hive.serde2.objectinspector.primitive.BooleanObjectInspector
hive> select count(*) from tweets where text like '%event%' or text like '%Event%' and polarity < 0.0;
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_20201030060055_a108454d-ee0c-4083-a01b-252584c0960a
Total jobs = 1
Launching Job 1 out of 1
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>
```

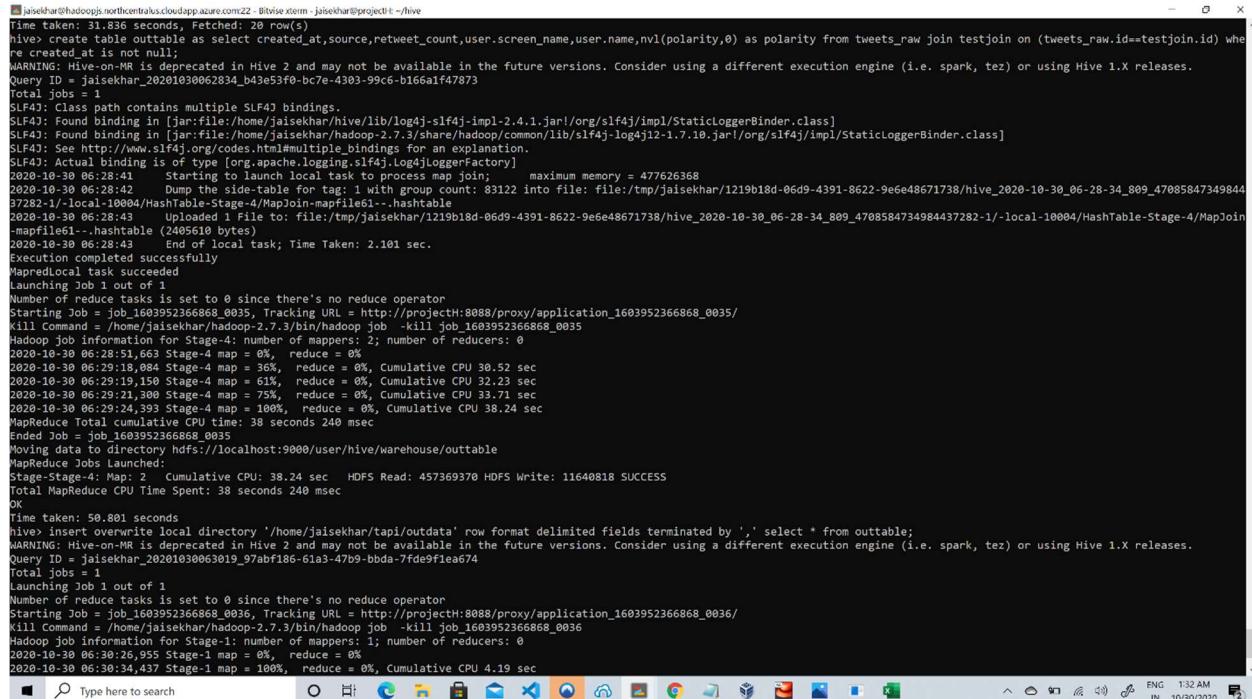
5. Here, we are getting the number of negative tweets which contains the keyword 'Event' and there are 1462 tweets in this dataset.

```
jaisekhar@hadoopjs:~$ hive -e "select count(*) from tweets where text like '%event%' or '%Event%' and polarity < 0.0;" 
Time taken: 23.217 seconds, Fetched: 1 row(s)
hive> select count(*) from tweets where text like '%event%' or text like '%Event%' and polarity < 0.0;
FAILED: ClassCastException org.apache.hadoop.hive.serde2.objectinspector.primitive.WritableConstantStringObjectInspector cannot be cast to org.apache.hadoop.hive.serde2.objectinspector.primitive.BooleanObjectInspector
hive> select count(*) from tweets where text like '%event%' or text like '%Event%' and polarity < 0.0;
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_20201030060055_a108454d-ee0c-4083-a01b-252584c0960a
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducer.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_0031, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0031/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0031
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-10-30 05:57:27,002 Stage-1 map = 0%, reduce = 0%
2020-10-30 05:57:34,389 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 4.16 sec
2020-10-30 05:57:41,839 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 7.0 sec
MapReduce Total cumulative CPU time: 7 seconds 0 msec
Ended Job = job_1603952366868_0031
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1   Cumulative CPU: 7.0 sec   HDFS Read: 36236745 HDFS Write: 103 SUCCESS
Total MapReduce CPU Time Spent: 7 seconds 0 msec
OK
234
Time taken: 23.217 seconds, Fetched: 1 row(s)
hive> select count(*) from tweets where text like '%event%' or '%Event%' and polarity < 0.0;
FAILED: ClassCastException org.apache.hadoop.hive.serde2.objectinspector.primitive.WritableConstantStringObjectInspector cannot be cast to org.apache.hadoop.hive.serde2.objectinspector.primitive.BooleanObjectInspector
hive> select count(*) from tweets where text like '%event%' or text like '%Event%' and polarity < 0.0;
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_20201030060055_a108454d-ee0c-4083-a01b-252584c0960a
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducer.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_0032, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0032/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0032
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-10-30 06:01:02,516 Stage-1 map = 0%, reduce = 0%
2020-10-30 06:01:10,965 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 5.35 sec
2020-10-30 06:01:18,430 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 7.9 sec
MapReduce Total cumulative CPU time: 7 seconds 900 msec
Ended Job = job_1603952366868_0032
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1   Cumulative CPU: 7.9 sec   HDFS Read: 36236729 HDFS Write: 104 SUCCESS
Total MapReduce CPU Time Spent: 7 seconds 900 msec
OK
1462
Time taken: 24.028 seconds, Fetched: 1 row(s)
hive>
```

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

```
jaisekhar@hadoopjs:~$ hive -e "select created_at,source,retweet_count,user.screen_name,user.name,polarity from tweets limit 10;" 
Time taken: 0.036 seconds, Fetched: 15 row(s)
hive> select created_at,source,year,day from tweets limit 10;
NULL      NULL      NULL
Tue Oct 06 23:59:59 +0000 2020  NULL      NULL      NULL
Tue Oct 06 23:58:02 +0000 2020  NULL      NULL      NULL
Tue Oct 06 23:57:35 +0000 2020  NULL      NULL      NULL
Tue Oct 06 23:57:19 +0000 2020  NULL      NULL      NULL
Tue Oct 06 23:56:47 +0000 2020  NULL      NULL      NULL
Tue Oct 06 23:56:30 +0000 2020  NULL      NULL      NULL
Tue Oct 06 23:56:18 +0000 2020  NULL      NULL      NULL
Tue Oct 06 23:56:02 +0000 2020  NULL      NULL      NULL
Tue Oct 06 23:55:51 +0000 2020  NULL      NULL      NULL
NULL      NULL      NULL
Time taken: 0.081 seconds, Fetched: 10 row(s)
hive> insert overwrite local directory '/home/jaisekhar/tapi/savedata' row format delimited fields terminated by ',' select created_at,source,retweeted_count,user.screen_name,user.name,polarity from tweets;
FAILED: SemanticException [Error 10004]: Line 1:136 Invalid table alias or column reference 'retweeted_count': (possible column names are: id, created_at, source, favorited, retweet_count, retweeted_status, entities, text, user, in_reply_to_screen_name, year, month, day, hour, polarity)
hive> insert overwrite local directory '/home/jaisekhar/tapi/savedata' row format delimited fields terminated by ',' select created_at,source,retweet_count,user.screen_name,user.name,polarity 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 engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = jaisekhar_2020103004930_22e0a15c-4826-acba-1cede010cf0
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1603952366868_0020, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0020/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0020
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2020-10-30 04:49:38,086 Stage-1 map = 0%, reduce = 0%
2020-10-30 04:49:45,464 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 4.12 sec
MapReduce Total cumulative CPU time: 4 seconds 120 msec
Ended Job = job_1603952366868_0020
Moving data to local directory '/home/jaisekhar/tapi/savedata'
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 4.12 sec   HDFS Read: 36229022 HDFS Write: 16707511 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 120 msec
OK
Time taken: 17.678 seconds
hive> select created_at,source,retweet_count,user.screen_name,user.name,polarity from tweets limit 10;
OK
Tue Oct 06 23:59:59 +0000 2020 <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>    0      house_money_      House Money      0.5
Tue Oct 06 23:58:02 +0000 2020 <a href="http://twitter.com/#/download/ipad" rel="nofollow">Twitter for iPad</a>    0      InvesToor      Broke Investor      0.0
Tue Oct 06 23:57:35 +0000 2020 <a href="https://mobile.twitter.com" rel="nofollow">Twitter Web App</a>  0      sunchartist      Sunchartist      0.0
Tue Oct 06 23:57:19 +0000 2020 <a href="https://mobile.twitter.com" rel="nofollow">Twitter Web App</a>  0      PennyPresident      Penny President Yang      0.0
Tue Oct 06 23:56:47 +0000 2020 <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>    0      notstockadvice3      Alex      0.0
Tue Oct 06 23:56:30 +0000 2020 <a href="https://mobile.twitter.com" rel="nofollow">Twitter Web App</a>  0      Moorewealthal      Susan Moore      0.0
Tue Oct 06 23:56:18 +0000 2020 <a href="https://mobile.twitter.com" rel="nofollow">Twitter Web App</a>  0      Puucktalk      dailypicks      -2.0
Tue Oct 06 23:56:02 +0000 2020 <a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>    0      MichaelJobe      Michael Jobe      0.0
Time taken: 0.036 seconds
hive>
```

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



```
jaisekhar@hadoop:~$ hive -e "CREATE TABLE outtable AS SELECT created_at,source,retweet_count,user.screen_name,user.name,nvl(polarity,0) AS polarity FROM tweets_raw JOIN testjoin ON (tweets_raw.id==testjoin.id) WHERE created_at IS NOT NULL;" -- Query ID = jaisekhar_20201030062834_b43e53f0-bc7e-4303-99c6-b166af47873
Total jobs = 1
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/jaisekhar/hive/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/jaisekhar/hadoop-2.7.3/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
2020-10-30 06:28:41 Starting to launch local task to process map join; maximum memory = 477626368
2020-10-30 06:28:42 Dump the map-side-table for tag: 1 with group count: 83122 into file: file:/tmp/jaisekhar/1219b18d-06d9-4391-8622-9e6e48671738/hive_2020-10-30_06-28-34_809_47085847349844
37282-1-local-10004/HashTable-Stage-4/MapJoin-mapfile61--.hashtable
2020-10-30 06:28:43 Uploaded 1 file to: file:/tmp/jaisekhar/1219b18d-06d9-4391-8622-9e6e48671738/hive_2020-10-30_06-28-34_809_4708584734984437282-1-local-10004/HashTable-Stage-4/MapJoin
-mapfile61--.hashtable (2405610 bytes)
2020-10-30 06:28:43 End of local task; Time Taken: 2.101 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1603952366868_0035, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0035/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0035
Hadoop job information for Stage-4: number of mappers: 2; number of reducers: 0
2020-10-30 06:28:51,663 Stage-4 map = 0%, reduce = 0%
2020-10-30 06:29:18,084 Stage-4 map = 30%, reduce = 0%, Cumulative CPU 30.52 sec
2020-10-30 06:29:19,150 Stage-4 map = 61%, reduce = 0%, Cumulative CPU 32.23 sec
2020-10-30 06:29:21,300 Stage-4 map = 75%, reduce = 0%, Cumulative CPU 33.71 sec
2020-10-30 06:29:24,393 Stage-4 map = 100%, reduce = 0%, Cumulative CPU 38.24 sec
MapReduce Total cumulative CPU time: 38 seconds 240 msec
Ended Job = job_1603952366868_0035
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/outtable
MapReduce Jobs Launched:
Stage-Stage-4: Map: 2 Cumulative CPU: 38.24 sec HDFS Read: 457369370 HDFS Write: 11640818 SUCCESS
Total MapReduce CPU Time Spent: 38 seconds 240 msec
OK
Time taken: 50.801 seconds
hive> insert overwrite local directory '/home/jaisekhar/tapi/outdata' row format delimited fields terminated by ',' select * from outtable;
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_20201030063019_97abf186-61a3-47b9-bbda-7fde9f1ea674
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1603952366868_0036, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0036/
Kill Command = /home/jaisekhar/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0036
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2020-10-30 06:30:26,955 Stage-1 map = 0%, reduce = 0%
2020-10-30 06:30:34,437 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 4.19 sec
MapReduce Total cumulative CPU time: 38 seconds 240 msec
Ended Job = job_1603952366868_0036
OK
Time taken: 50.801 seconds
hive>
```

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 engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = jaisekhar_20201030073931_8dc522d-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.
```

```
jaisekarh@hadoopj:~$ bin/hive -e "SELECT * FROM tweets_raw GROUP BY user.name ORDER BY count DESC LIMIT 10;" -hiveconf mapreduce.job.reduces=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/jaisekarh/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 07:39:35,650 Stage-2 map = 100%, 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 766 msec
Ended Job = job_1603952366868_0037
Stage-5 is selected by condition resolver.
Stage-1 is filtered out by condition resolver.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/jaisekarh/hive/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/jaisekarh/hadoop-2.7.3/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
2020-10-30 07:40:02 Starting to launch local task to process map join; maximum memory = 477626368
2020-10-30 07:40:02 Dump the side-table for tag: 1 with group count: 1 into file: file:/tmp/jaisekarh/1219b18d-06d9-4391-8622-9e6e48671738/hive_2020-10-30_07-39-31_293_8745835599822287755
5-1-local-10005/HashTable-Stage-3/MapJoin-mapfile71-.hashtable
2020-10-30 07:40:03 Uploaded 1 File to: file:/tmp/jaisekarh/1219b18d-06d9-4391-8622-9e6e48671738/hive_2020-10-30_07-39-31_293_8745835599822287755-1-local-10005/HashTable-Stage-3/MapJoin-mapfile71-.hashtable (288 bytes)
2020-10-30 07:40:03 End of local task; Time Taken: 0.891 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 3 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1603952366868_0038, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0038/
Kill Command = /home/jaisekarh/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0038
Hadoop job information for Stage-3: number of mappers: 2; number of reducers: 0
2020-10-30 07:40:11,676 Stage-3 map = 0%, reduce = 0%
2020-10-30 07:40:31,328 Stage-3 map = 61%, reduce = 0%, Cumulative CPU 22.63 sec
2020-10-30 07:40:34,548 Stage-3 map = 75%, reduce = 0%, Cumulative CPU 28.05 sec
2020-10-30 07:40:36,734 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 29.85 sec
MapReduce Total cumulative CPU time: 29 seconds 856 msec
Ended Job = job_1603952366868_0038
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 6.76 sec HDFS Read: 36235131 HDFS Write: 116 SUCCESS
Stage-Stage-3: Map: 2 Reduce: 0 Cumulative CPU: 29.85 sec HDFS Read: 457367714 HDFS Write: 208 SUCCESS
Total MapReduce CPU Time Spent: 36 seconds 610 msec
OK
Earnings Whispers      775
Time taken: 67.545 seconds, Fetched: 1 row(s)
hive>
```

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

```
jaisekarh@hadoopj:~$ bin/hive -e "SELECT user.name, count(*) AS count FROM tweets_raw GROUP BY user.name ORDER BY count DESC LIMIT 10;" -hiveconf mapreduce.job.reduces=1
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 = jaisekarh_20201030075001_6df17d34-ba0b-4dc6-97c2-7c630aa97c
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 2
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_0041, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0041/
Kill Command = /home/jaisekarh/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0041
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 2
2020-10-30 07:50:01,021 Stage-1 map = 0%, reduce = 0%
2020-10-30 07:50:27,579 Stage-1 map = 48%, reduce = 0%, Cumulative CPU 22.66 sec
2020-10-30 07:50:28,622 Stage-1 map = 67%, reduce = 0%, Cumulative CPU 23.58 sec
2020-10-30 07:50:30,725 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 25.73 sec
2020-10-30 07:50:39,389 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 32.46 sec
MapReduce Total cumulative CPU time: 32 seconds 460 msec
Ended Job = job_1603952366868_0041
Launching Job 2 out of 2
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_0042, Tracking URL = http://projectH:8088/proxy/application_1603952366868_0042/
Kill Command = /home/jaisekarh/hadoop-2.7.3/bin/hadoop job -kill job_1603952366868_0042
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2020-10-30 07:50:53,241 Stage-2 map = 0%, reduce = 0%
2020-10-30 07:50:59,635 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 2.31 sec
2020-10-30 07:51:06,016 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 4.66 sec
MapReduce Total cumulative CPU time: 4 seconds 660 msec
Ended Job = job_1603952366868_0042
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2 Reduce: 2 Cumulative CPU: 32.46 sec HDFS Read: 457373148 HDFS Write: 753692 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 4.66 sec HDFS Read: 759318 HDFS Write: 403 SUCCESS
Total MapReduce CPU Time Spent: 37 seconds 120 msec
OK
The Right Stock Alerts 2055
The Tribe of Benjamin 1706
TIN-Tech Bloggers 785
AppleRetweetBot 731
Sam rothstein 579
```


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.

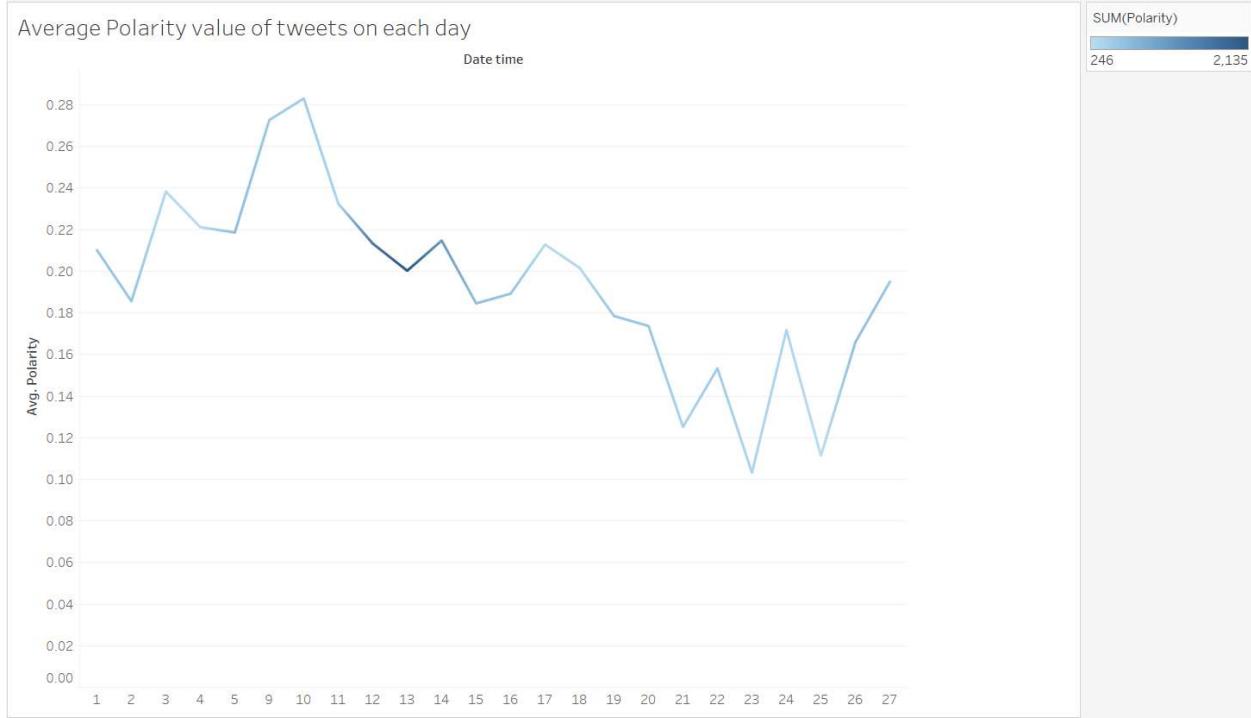
```
cqlsh:jai> select count(*) as negative from ctweets where polarity < 0 allow filtering;
-
negative
-----
4896
(1 rows)

Warnings :
Aggregation query used without partition key
```

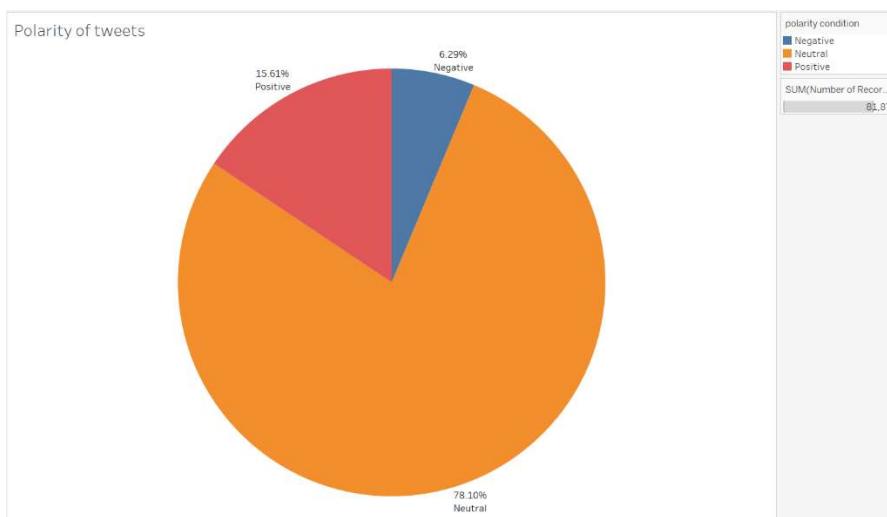
Visualization

We used Tableau for the visualilzation

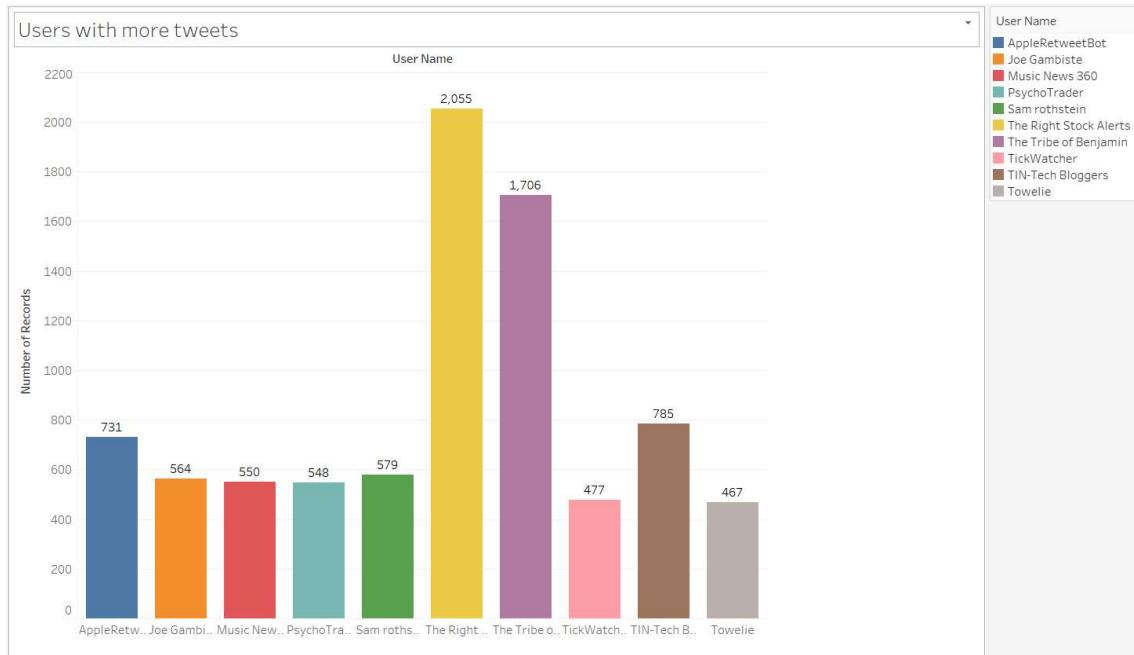
12. The Average polarity of tweets based on keyword **AAPL** on each day of October month. From October 5th to 10th there is a gradual increase in the polarity value. The highest average polarity value is recorded on October 10th. From 10th of the month polarity value gradually decreases.



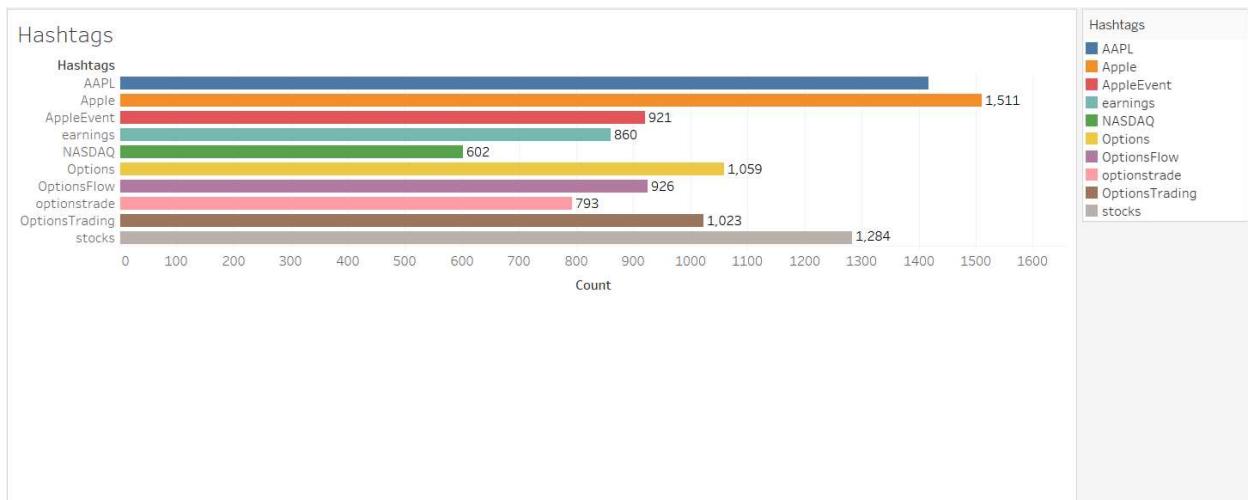
13. Polarity value of each tweet is calculated by using sentiment analysis on each tweet. We get values in the rage between -5 to +5. If the polarity value lies between -5 and 0 it is Negative. If the polarity value lies between 0 and 5 it is Positive. If the polarity value is 0 and 1 it is Neutral. 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 our 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 chosen 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>