**CSEE5590-0001/490-0003: Big Data Programming Project Proposal**

**Project Proposal:**

Spark ETL on Twitter Data, Visualization.

**Project Proposal Description:**

**Team Members:**

Bollepalli Sai Prasad Raju-5

Gomes Paul – 6

Jagadheesh Maroju -12

Poluri Praveen-14

**Goals and Objectives:**

**Motivation:**

In this data-driven world, handling data has become vital in the decision-making process in many industries such as Telecom, Banking, Financial and Health sector servicing industries. Managing the sheer volumes of data and getting insights from it would be the main factors. One of the amazing frameworks that can handle big data in real-time and perform different analysis, using Apache Spark.

**Objectives:**

Our Project's main idea is to do the ETL process using Spark Batch Processing and Integrating Spark with Web UI. The source of our system is Twitter data and we would be using Spark batch Processing to collect the data. Then we would be performing the transformations on the set of RDD’s and load the data into the Hive system which is similar to basic ETL process.

**Features:**

The project’s features includes collecting real-time tweets from twitter streaming API, performing ETL (Pre-Processing the data, Extracting necessary information and loading the data in to the Hive), and using TextBlob, predict the sentiment for each tweet. We are trying to feed the data into HDFS and implement Hive, SQL queries and we are using Sqoop to transfer data between SQL and HDFS.

**Significance:**

For sentiment analysis we are using a existing ML tool(TextBlob) to predict the sentiment of the tweets. Use spark to write queries on visualization using panda.

**Implementation:**

1. Collect tweets from Twitter api.

2. Import the collected data into hive from HDFS.

3. Export the data into RDBMS using sqoop.

4. Sentimental analysis on collected tweets.

5. Sentimental analysis on collected data.

6. Use spark sql to write queries and visualize(bar graphs/pie charts etc) the results using panda.

**References:**

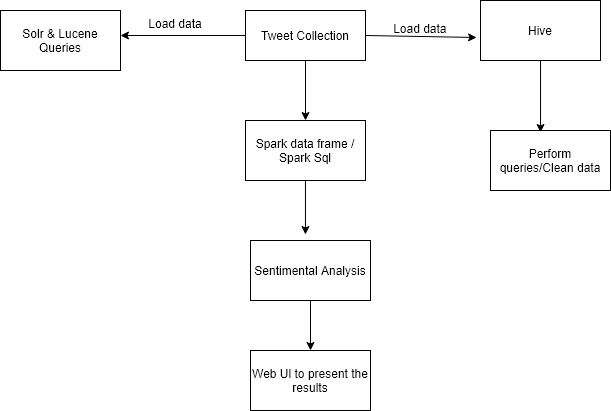
1. <https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json>
2. [https://spark.apache.org/docs/latest/](https://spark.apache.org/docs/latest/streaming-programming-guide.html)

**Big Data Programming Project Increment 1**

**Dataset:**

We are collecting twitter data using Twitter batch API using twitter developer account credentials with Republicans and Democrats as keywords. All tweets are in CSV format. We collected about 3 million tweets which is about 450 Megabytes. It has description of a tweet like user\_description, user\_location, coordinates, user\_name, user\_created, user\_followers(follower\_id, created), retweet\_count, polarity, subjectivity. We are collecting using batch API and able to download 4kb per second in which we are filtering for keywords like Democrat supporters, Republican supporters.

**Block-diagram:**



We created a Twitter Developers API account, then we got API tokens and credentials with which we got the tweets using tweetpy and twitter streaming API in python and stored tweets into sqlite db.Imported the queries into csv file using python. Collected around 4 million tweets. Imported the csv file into HDFS. Visualized the csv data file in Hue. Processed the dataset using Hue by creating database& database table in hue and implemented various queries to get specific information on tweets. For next phases we are going to Implement solr& lucene on the data, create spark dataframes on CSV file and implement various transformations and actions on dataframes. We are also going to implement Spark sql, and query the data from dataframe. we will implement sentiment analysis on the data.

**Analysis of Increment-1:**

We created a Twitter Developers API account, then we got API tokens and credentials with which we got the tweets using tweetpy and twitter streaming API in python and stored tweets into sqlite db.Imported the queries into csv file using python. Collected around 4 million tweets. Imported the csv file into HDFS. Visualized the csv data file in Hue. Processed the dataset using Hue by creating database& database table in hue and implemented various queries to get specific information on tweets.

**Implementation:**

**Collected tweets:**

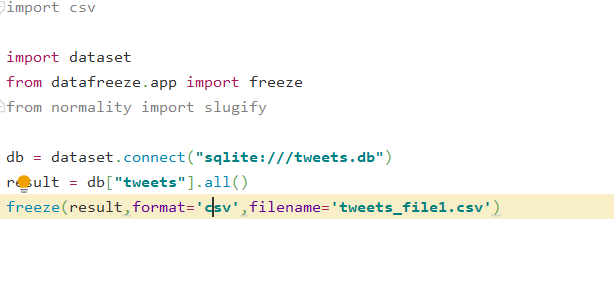
* Got developer access from twitter, then created an app to use their APIs.
* Using tweepy and twitter streaming API collected tweets in python and below is the code.



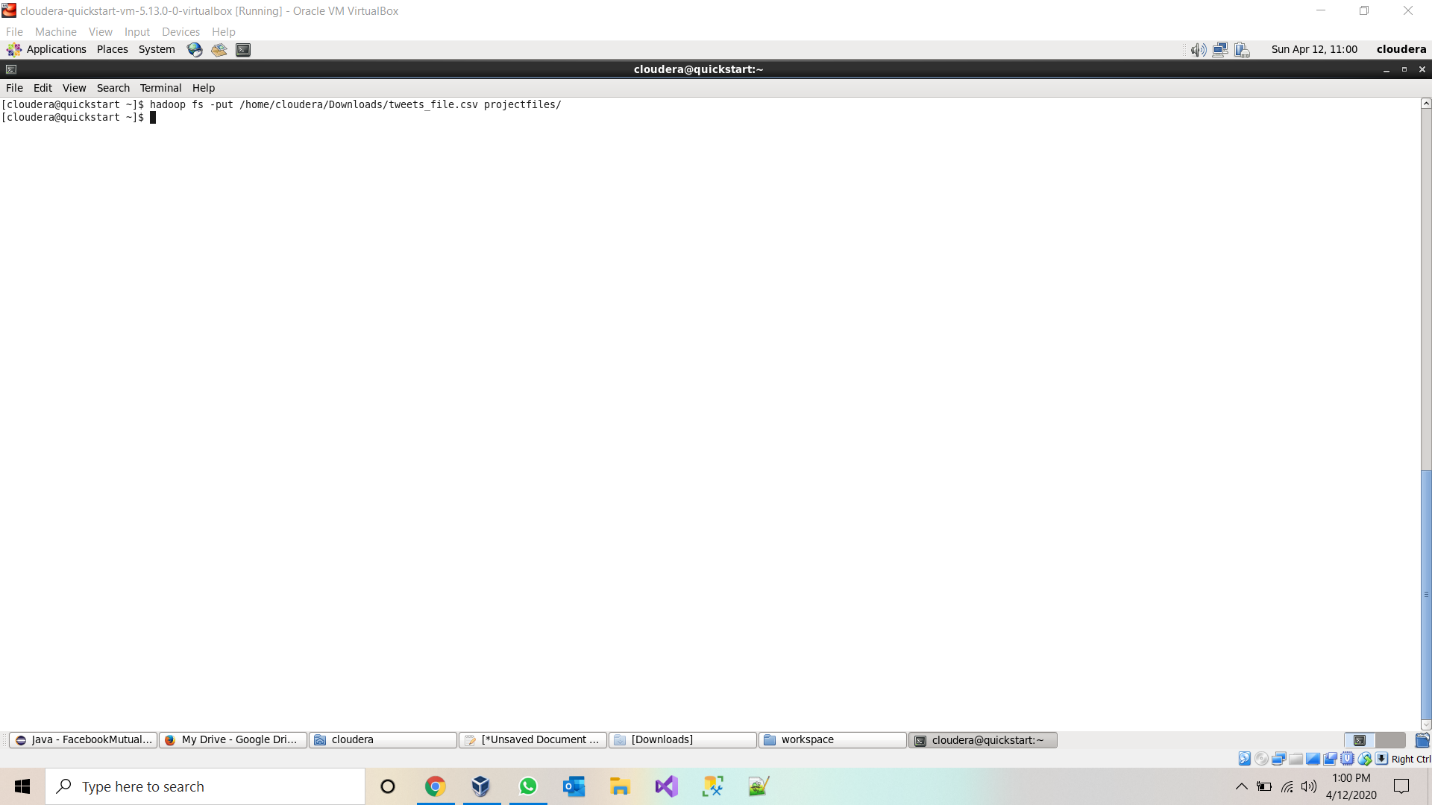
* Storing the tweets into sqlite db

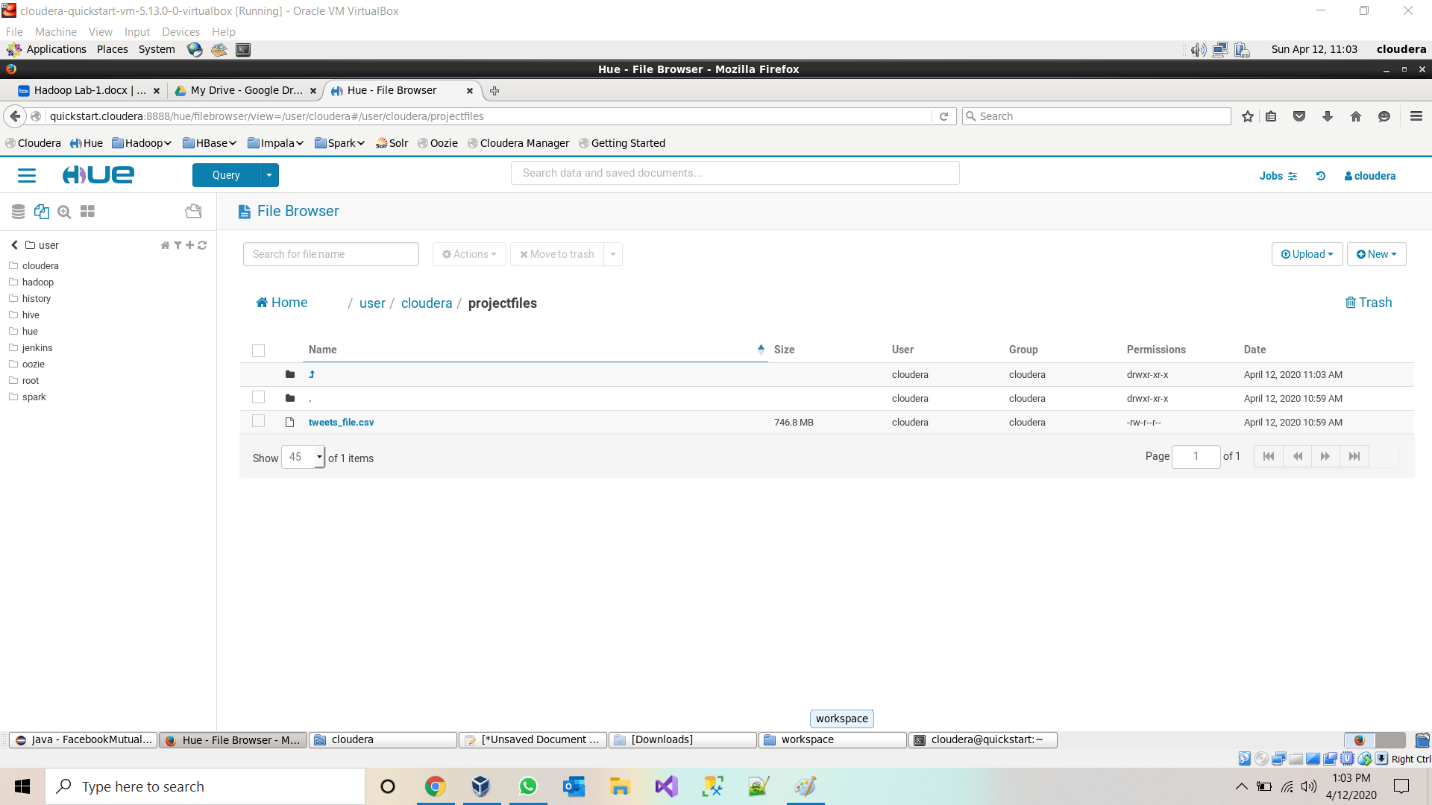


* After storing the tweets into sqlite db importing them into csv file using below code.

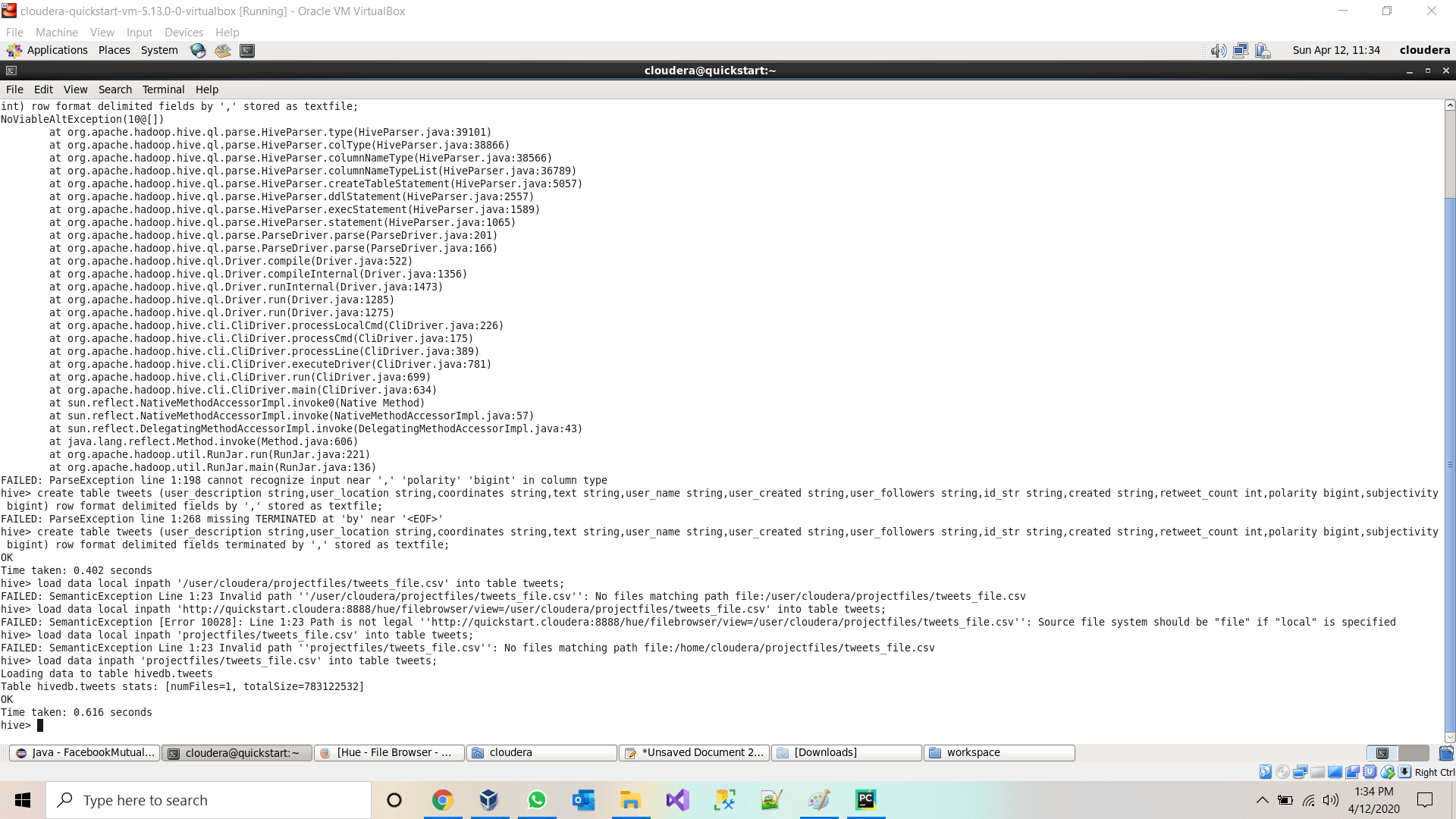


* Collected around 40 million tweets.
* Imported the file into hdfs.

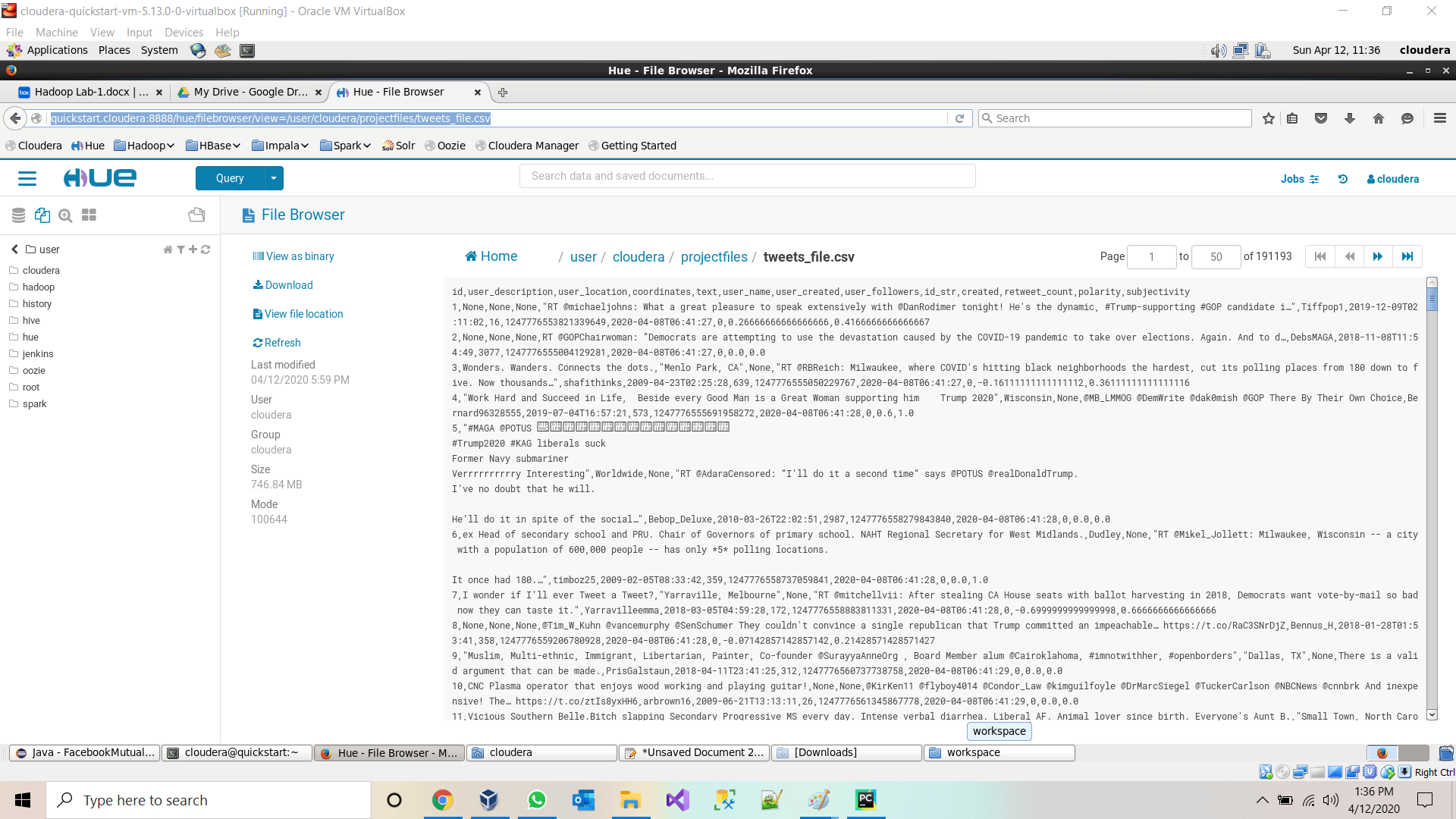




* Created a database and database table in hive and loaded the data into hive table.



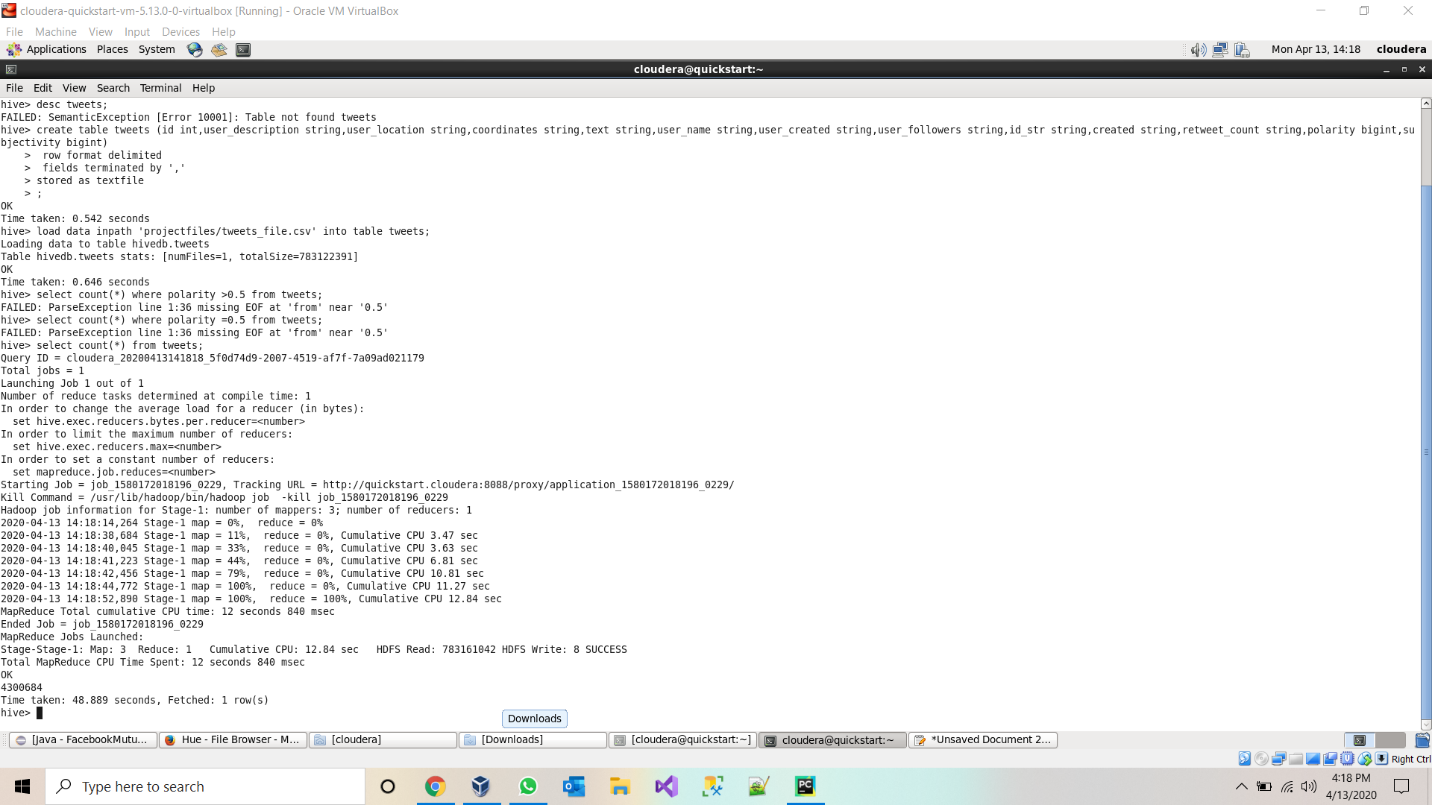
* Viewing data table in Hive.



* **Querying data from Hive Table:**

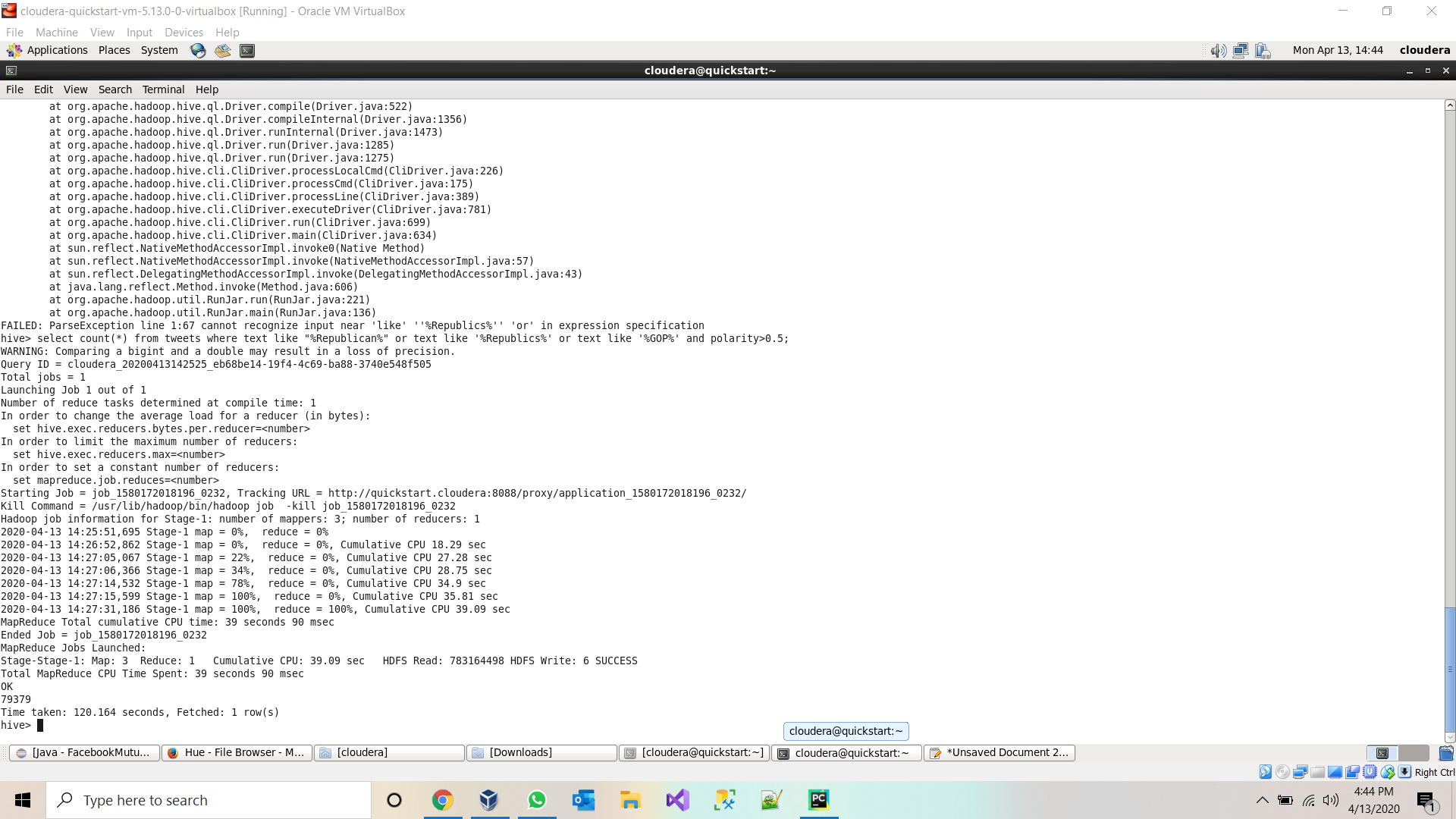
**Query 1:** 1.Basic query: total number of tweets using aggregate function

* Select count(\*) from tweets

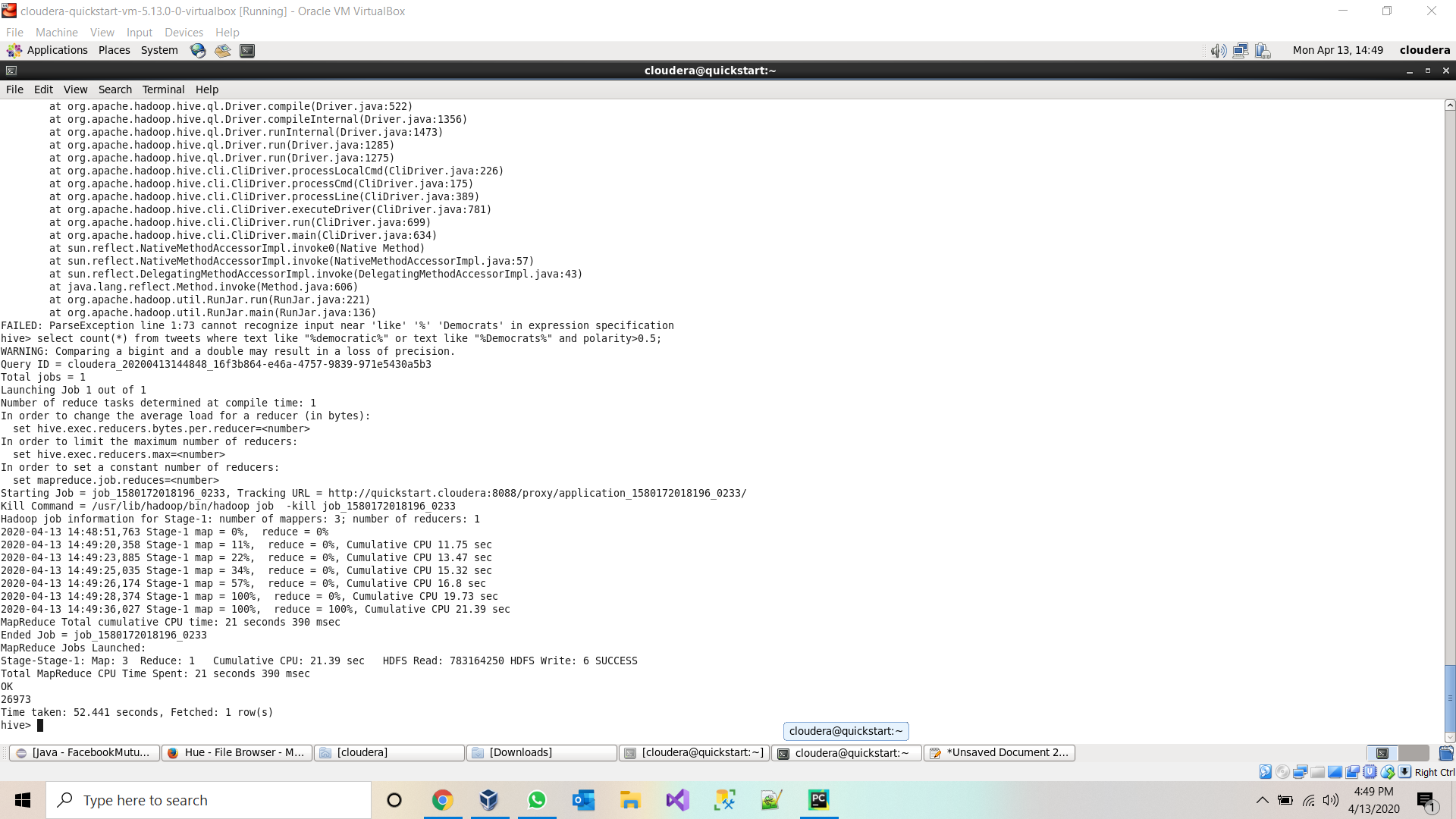


2.Number of positive tweets for republic party using polarity of sentimental analysis

select count(\*) from tweets where text like "%Republican%" or text like '%Republics%' or text like '%GOP%' and polarity>0.5;

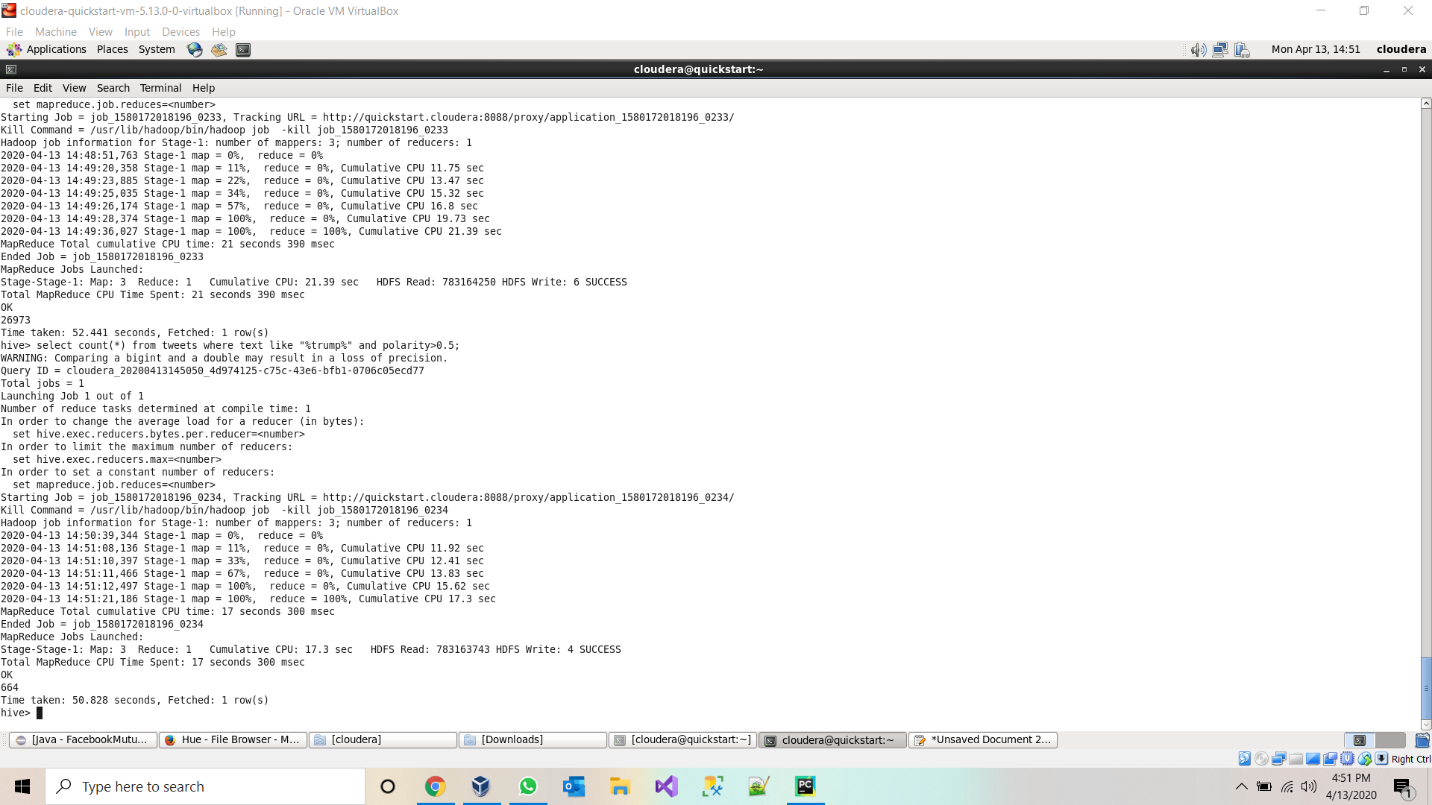


3.Count of positive tweets on democrats using polarity of sentimental analysis object. select count(\*) from tweets where text like "%democratic%" or text like "%Democrats%" and polarity>0.5;



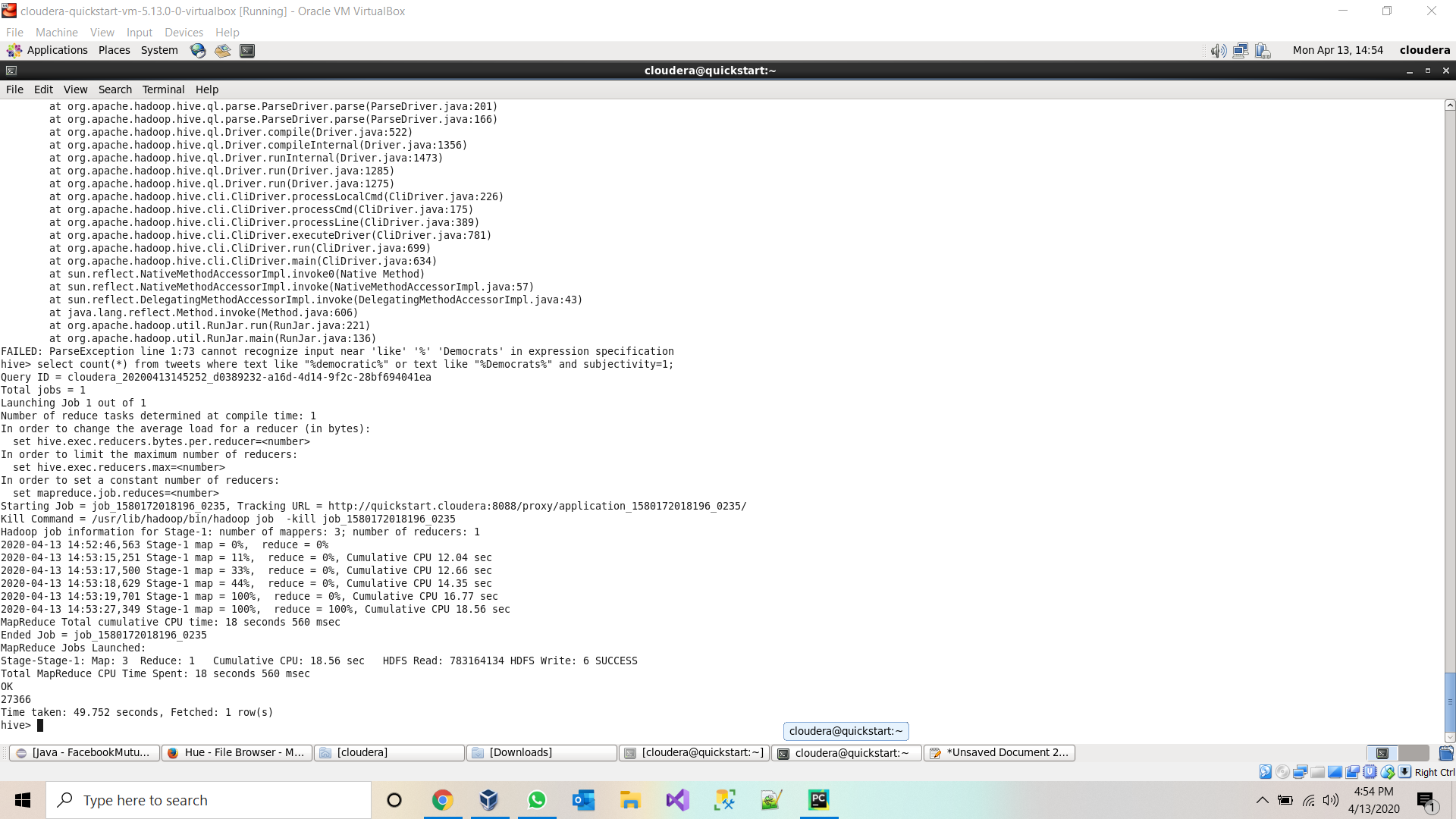
4.Tweets which contains text of US president trump and with positive.

select count(\*) from tweets where text like "%trump%" and polarity>0.5;

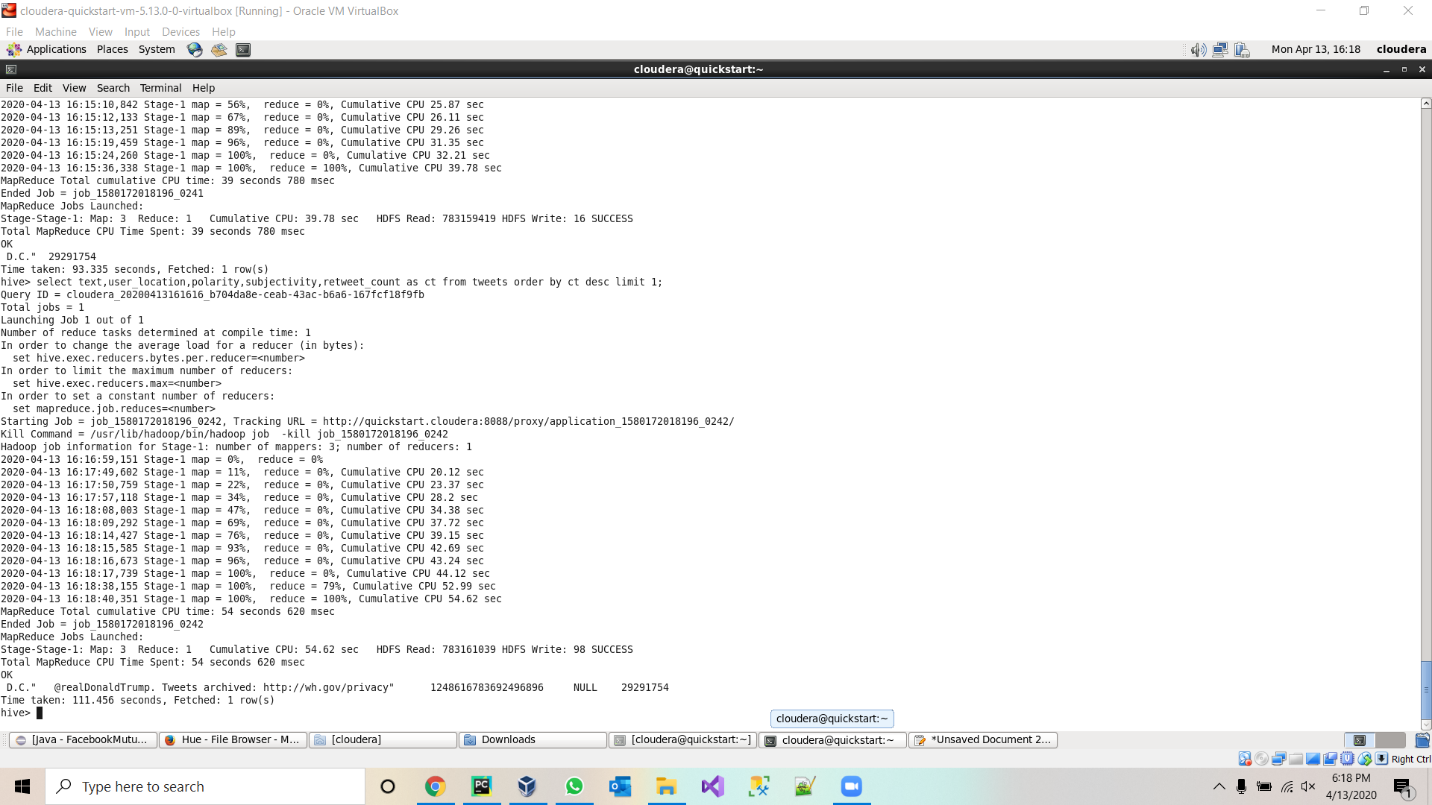


5.Count of tweets which are subjective towards democrats.

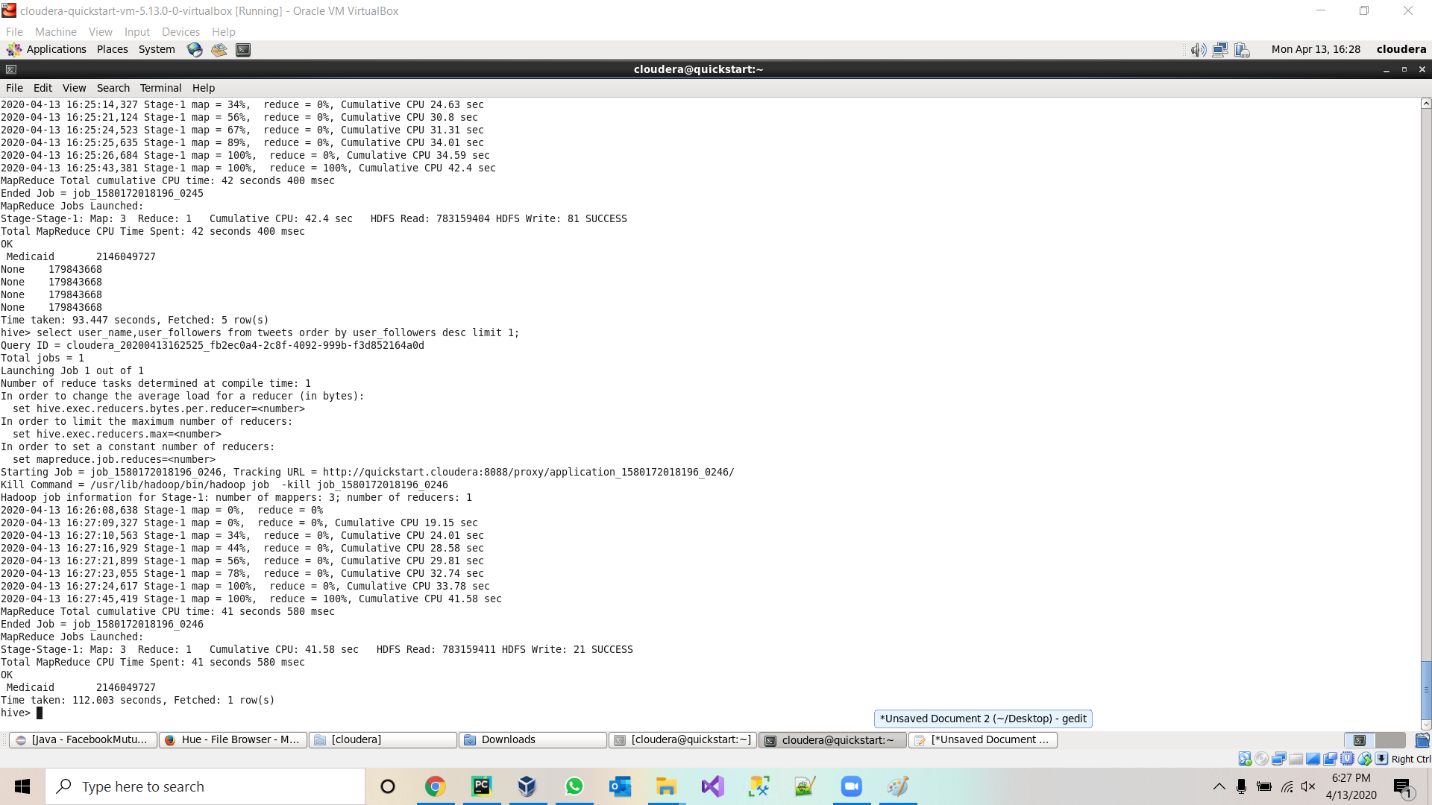
select count(\*) from tweets where text like "%democratic%" or text like "%Democrats%" and subjectivity=1;



6.Details of tweet which has highest retweet count select text,user\_location,polarity,subjectivity,retweet\_count as ct from tweets order by ct desc limit 1;

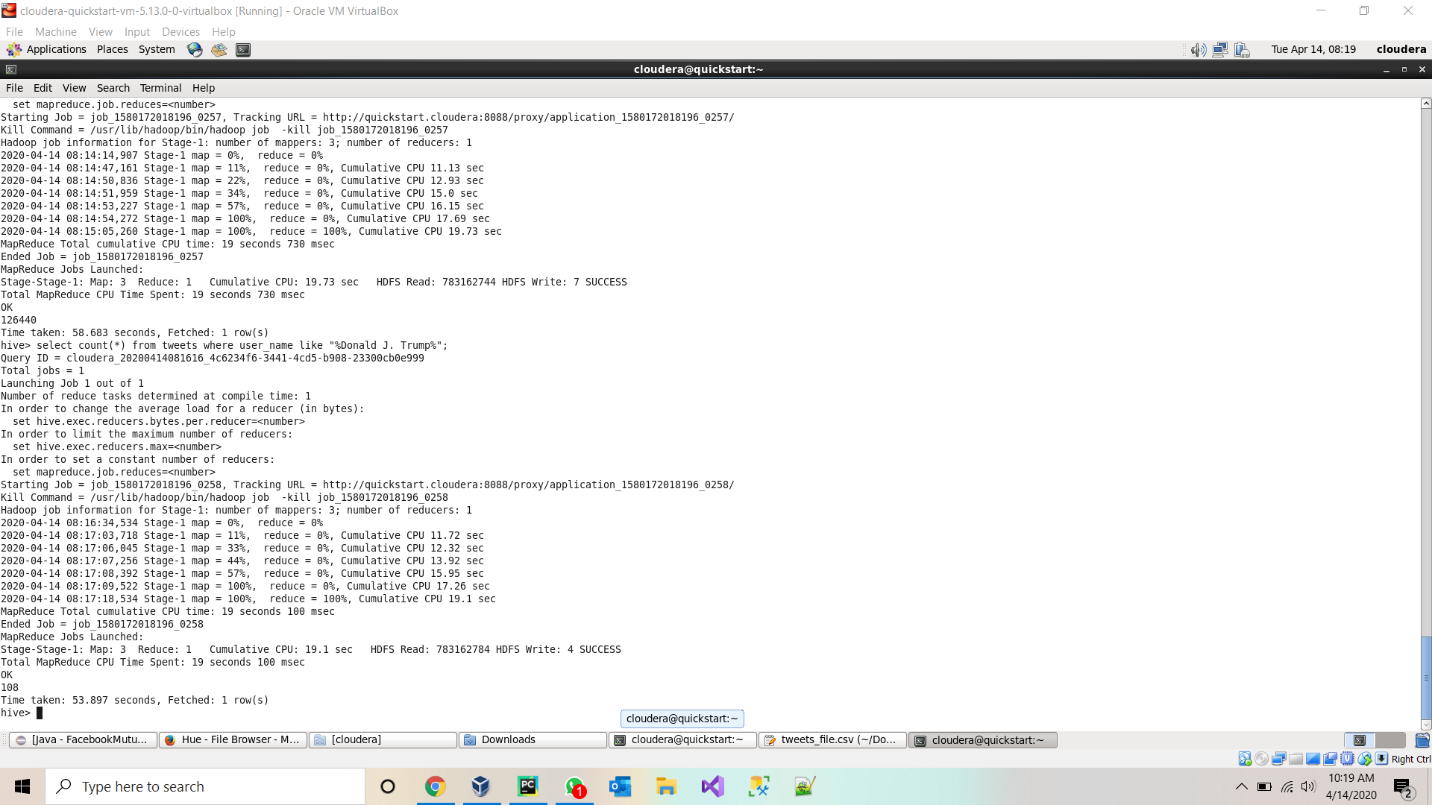


7.User which has highest follower select user\_name,user\_followers from tweets order by user\_followers desc limit 1;



8.Tweet by US president Donald Trump

Select count(\*) from tweets where user\_name like “%Donald J. Trump%”



**Project-Management:**

**Implementation-status-report:**

**Work-Completed:**

* Twitter-data batch download & documented his part-Jagadeesh Maroju
* Data Cleanup and project implementation plan & documented his part- Paul Gomes
* HDFS import and Hue Visualization and Hive import & documented his part- Hari Raju
* Hive Queries and report- Praveen Poluri

**Work to be completed:**

* Loading data into solr and implement queries on the data in solr.
* create spark dataframes on CSV file and implement various transformations and actions on dataframes.
* We are also going to implement Spark sql and query the data from dataframe.
* we will implement sentiment analysis on the data.

**References:**

* <https://towardsdatascience.com/twitter-data-collection-tutorial-using-python-3267d7cfa93e>
* <https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HDFSCommands.html>
* <https://docs.cloudera.com/HDPDocuments/HDP3/HDP-3.1.0/migrating-data/content/hive_moving_data_from_hdfs_to_hive.html>

**Increment 2:**

**Introduction:**

Collected three million tweets using twitter Streaming API, and saved the collected tweets in JSON file, Implemented Hive queries in HDFS using Hive. Built data frames on top of the JSON file, created temp table view for dataframe created and implemented Spark SQL queries on top of it. Visualized the query results & designed web application to visualize results.

**Dataset Description:**

We are gathering twitter information utilizing Twitter group API utilizing twitter developer account credentials with Republicans and Democrats as watchwords. All tweets are in JSON format. We gathered around 3 million tweets which is around 450 Megabytes. It has different data about a tweet like user\_description, user\_location, organizes, user\_name, user\_created, user\_followers(follower\_id, made), retweet\_count, extremity, subjectivity. We are gathering utilizing bunch API and ready to download 4kb every second in which we are separating for watchwords like Democrat supporters, Republican supporters.

URL: [**https://drive.google.com/open?id=18jn55iiWOOTp\_1r\_dPYG3PGkuSUxWnTp**](https://drive.google.com/open?id=18jn55iiWOOTp_1r_dPYG3PGkuSUxWnTp)

**OBJECTIVE:**

* Stream tweets from Twitter (we collected more than 300,000 tweets).
* Write at least 10 analytic queries to explore and understand the collected tweets.
* Create Visualizations on the results.
* Create a web page to view the visualizations.

**TOOLS and Technologies Used**

* Apache Spark
  + Spark Streaming to collect tweet, Spark SQL to store and execute queries.
* Matplot for visualization
* Flask for web application development
* HTML, CSS and Javascript for web application

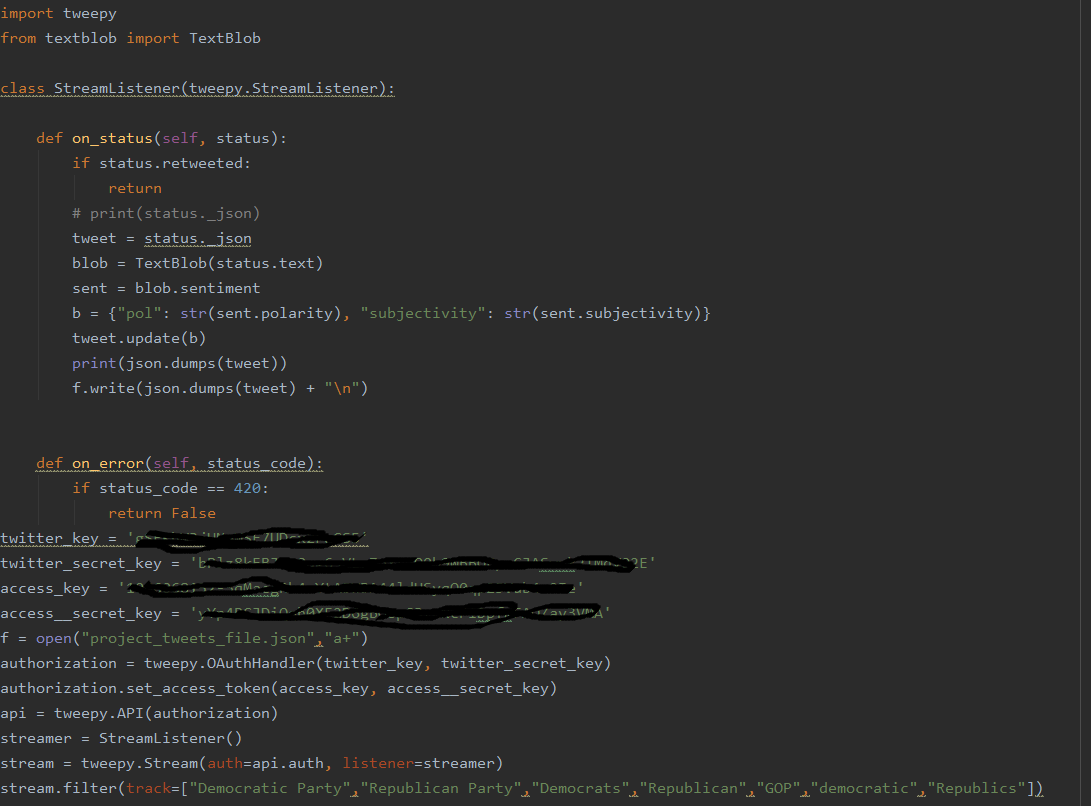
**Libraries used:**

* Matplot for visualizations
* TextBlob to get sentimental score
* Flask for web application

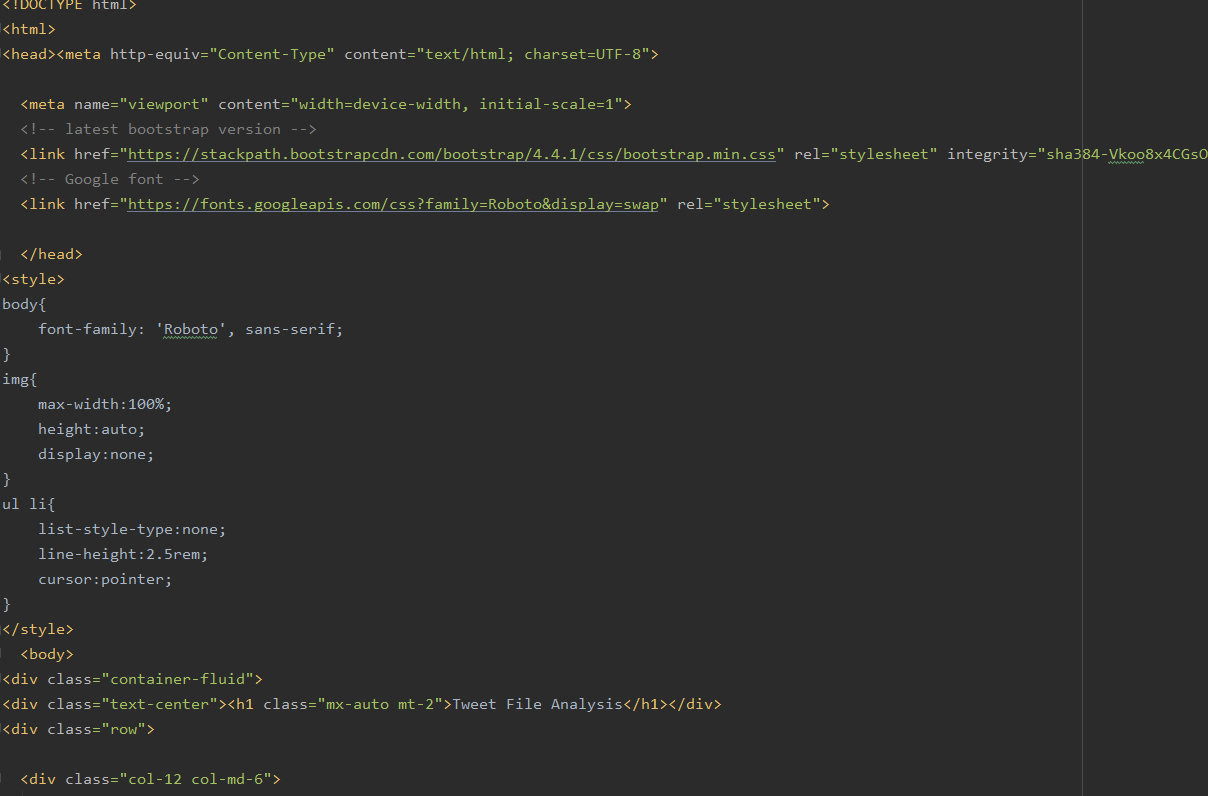
**IMPLEMENTATION**

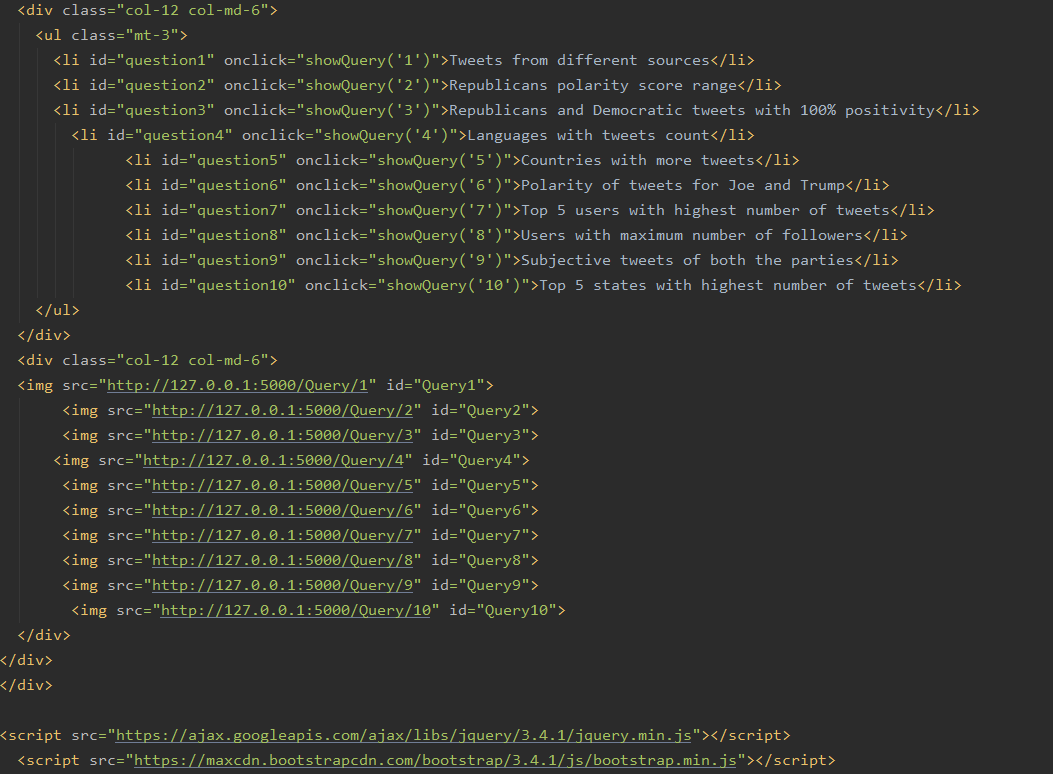
* Written a python program to stream the tweets and save them into JSON format which will have details of the tweets.
* The extracted JSON tweets are persisted into the Apache SparkSQL in the form of Views.
* Used Pycharm IDE to write queries and visualize the outputs using tables and charts.
* The web application is developed using HTML, CSS and Java script.

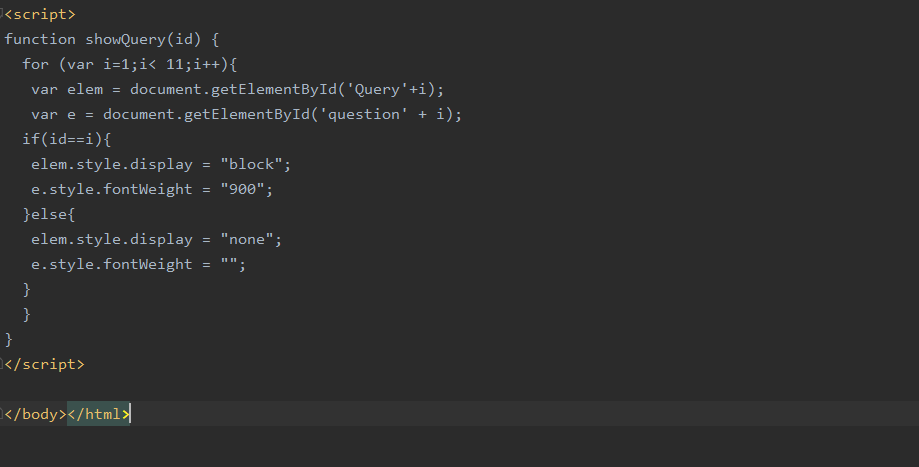
**Code for Collecting the Tweets:**



**HTML and JavaScript code for UI:**

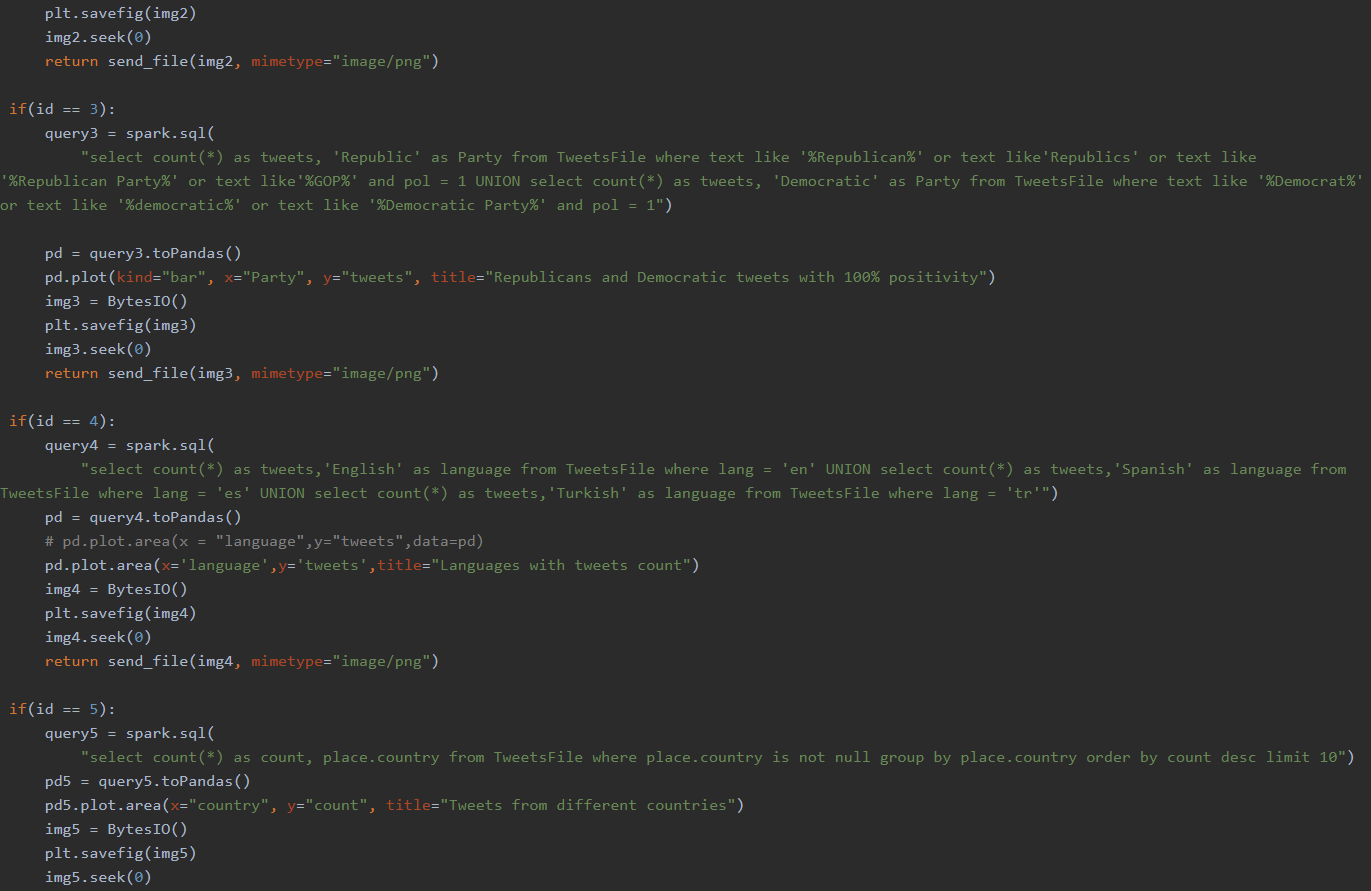


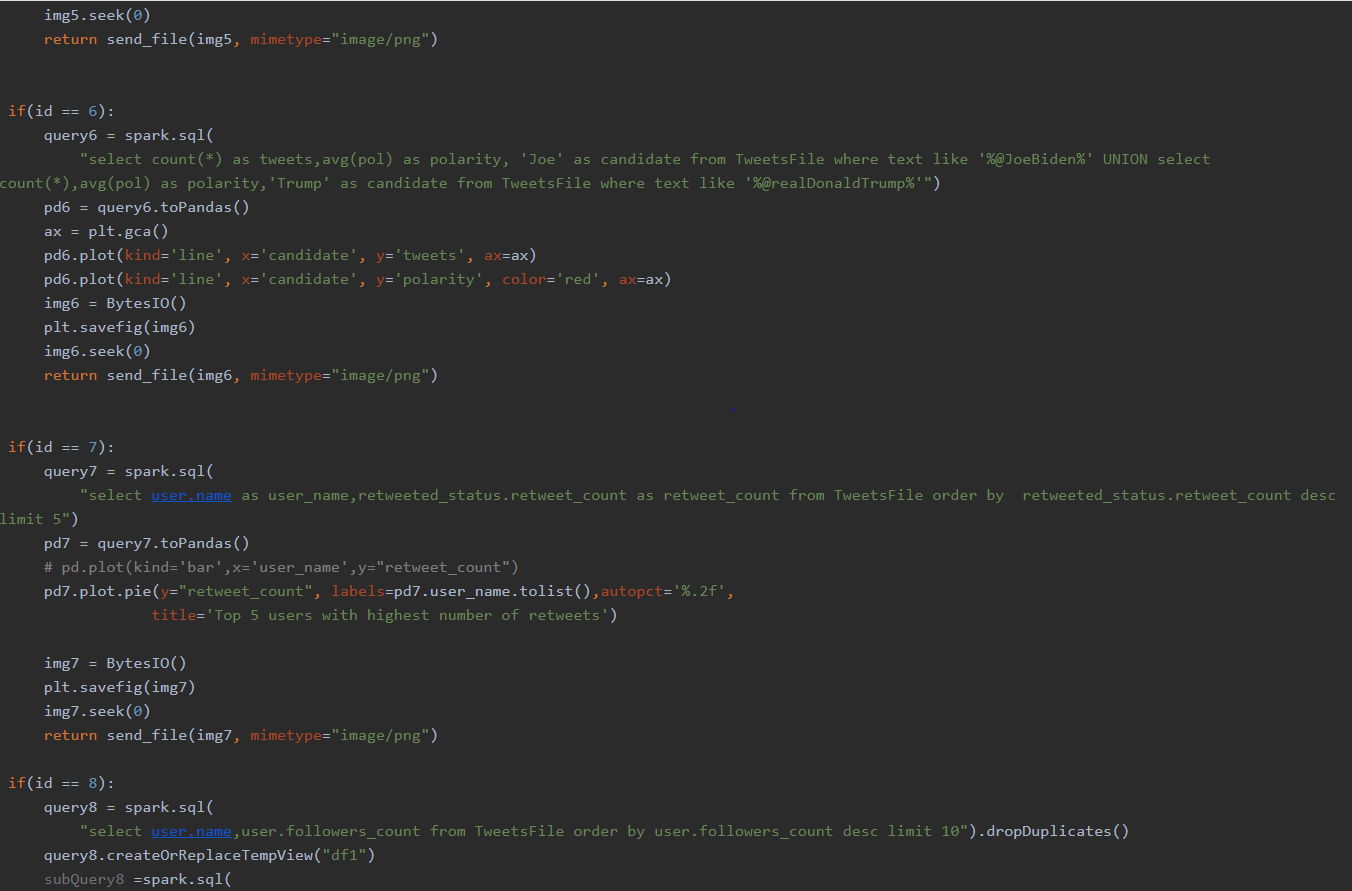


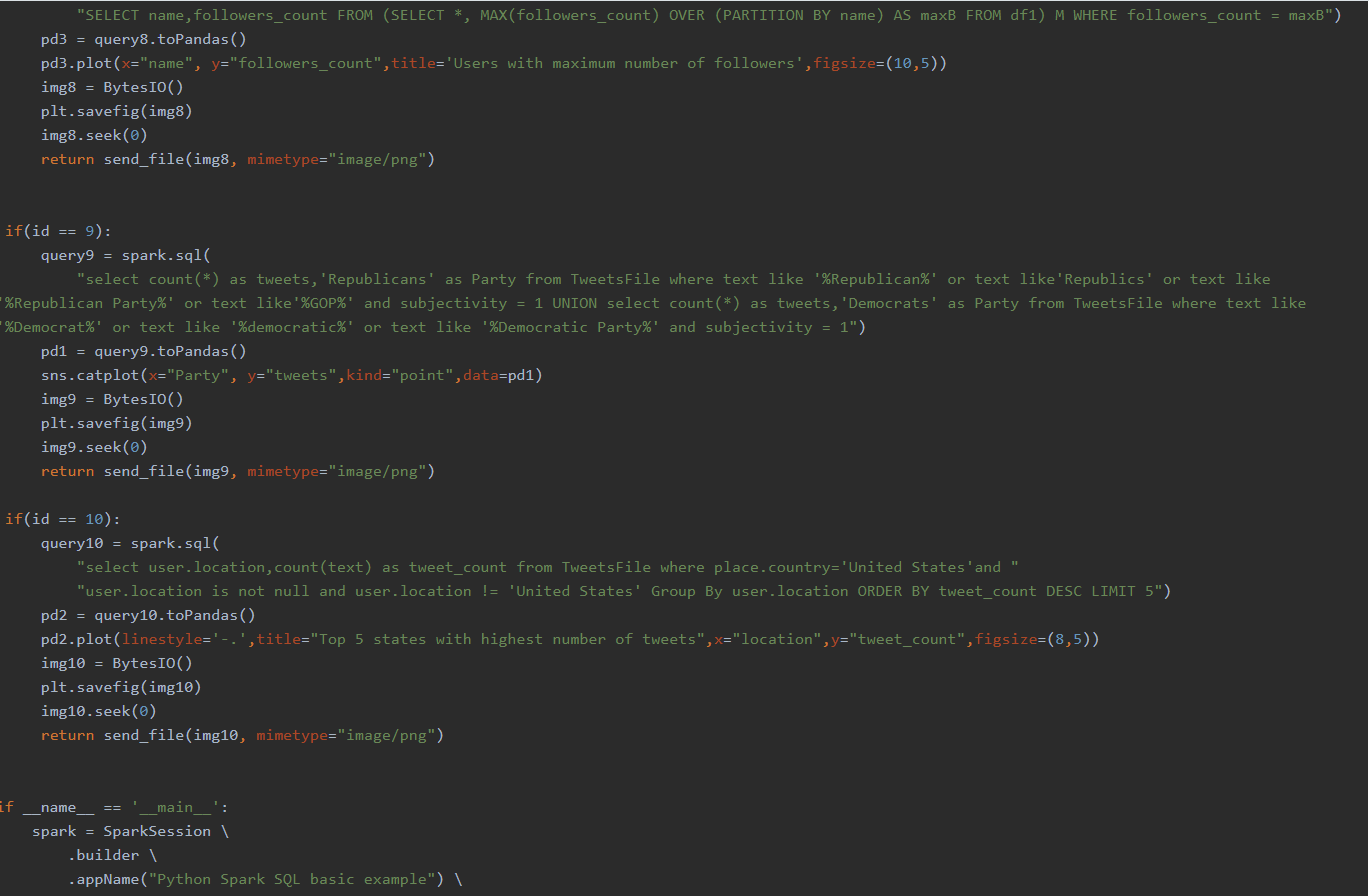


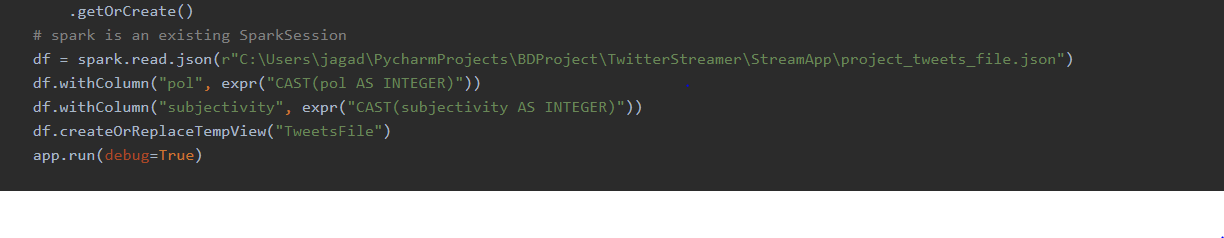
**Code For 10 Queries:**









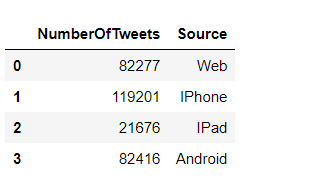


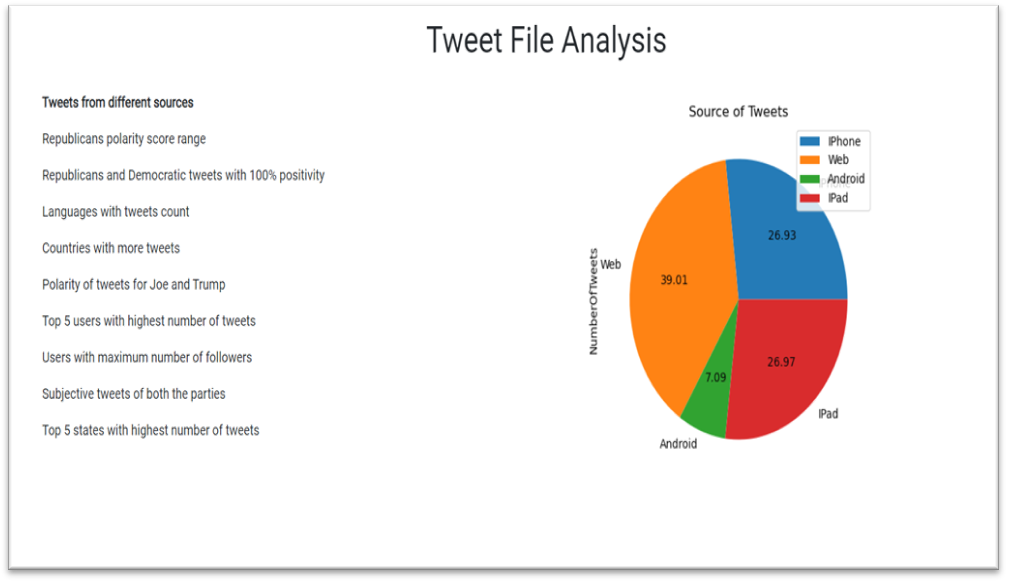
**Query Execution and Results:**

**Executed and visualized totally 10 queries.**

1). Tweets originated from Different sources using Pie Chart.

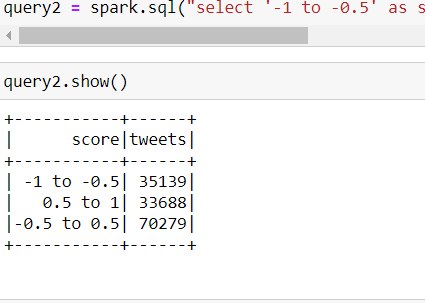
**Query result in table form:**



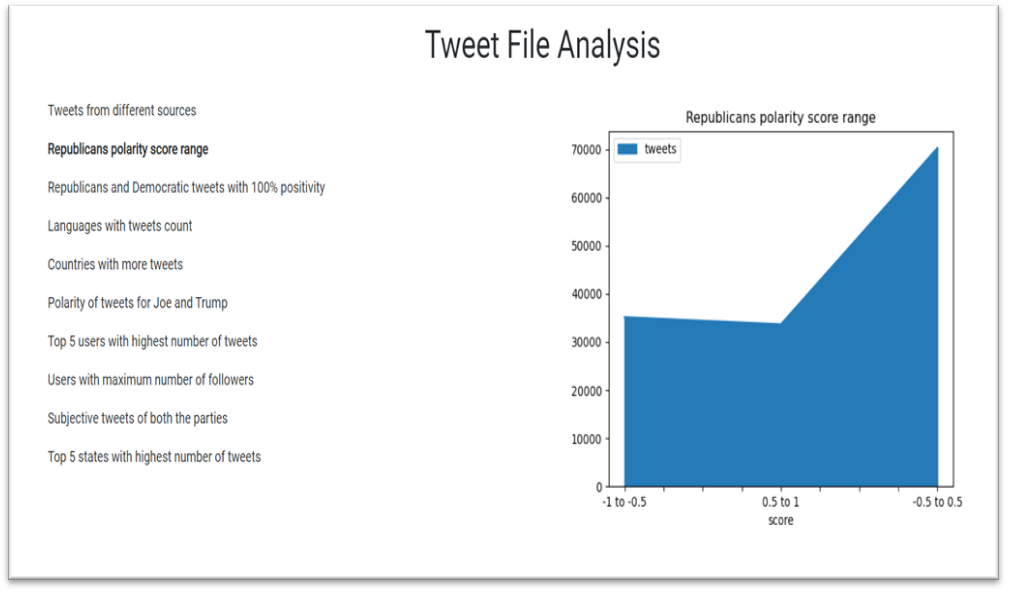


2) Query to display the Republicans with different ranges of polarity score using “Area plot”.

Result:

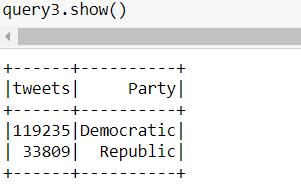


Sentimental Analysis using TextBlob:



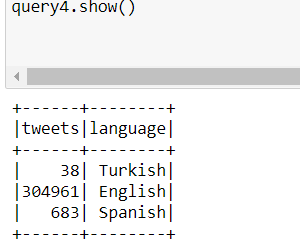
3). Tweets with 100% positivity( i.e polarity = 1) among Republicans and Democratic’s using

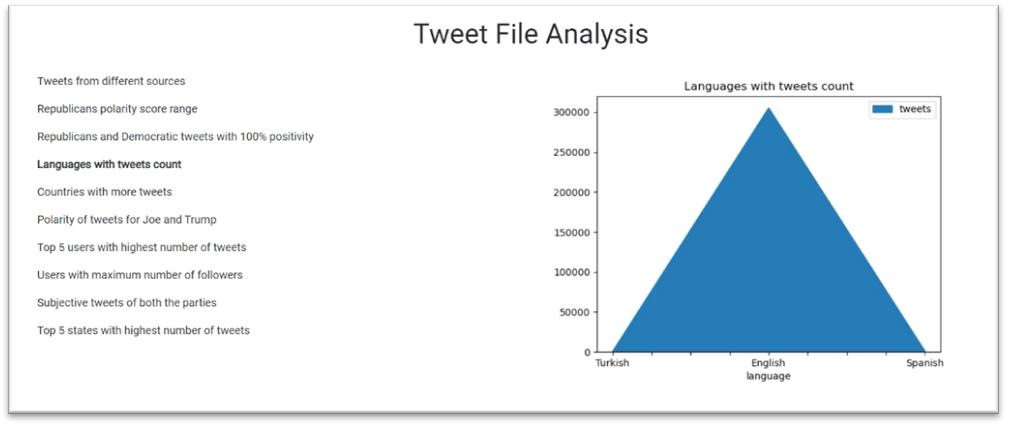
“Bar Chart”



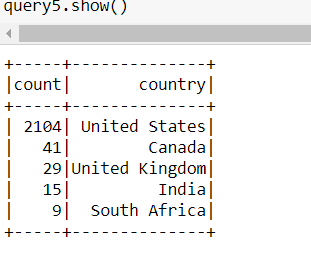


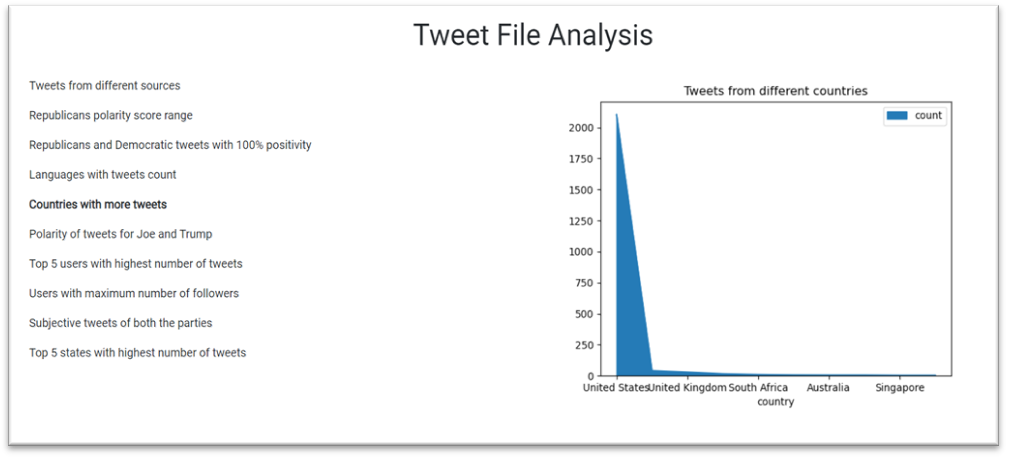
4). Tweets collected from different languages using “Area plot”



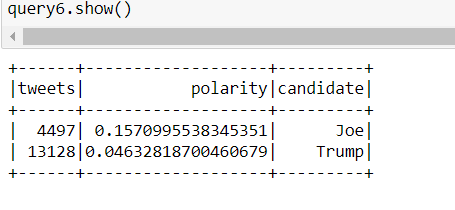


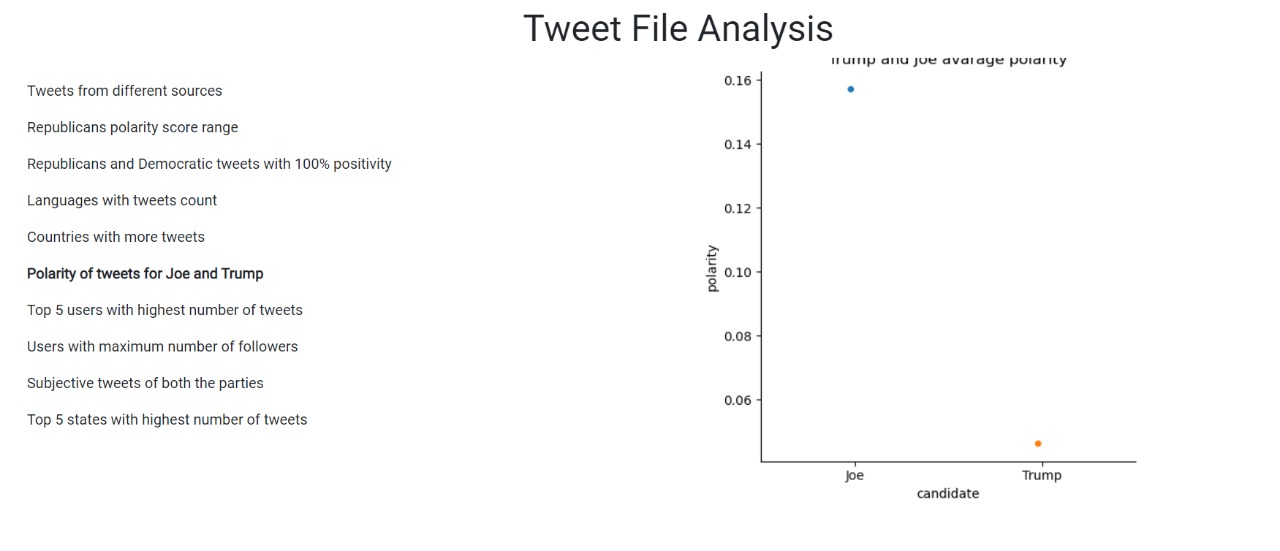
5). Top 5 Countries with Highest number of Tweets using “Area Code”



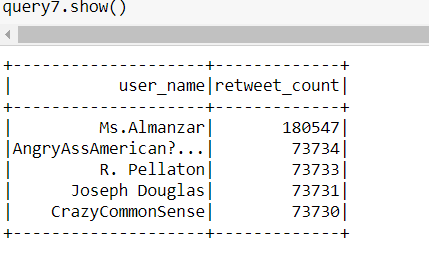


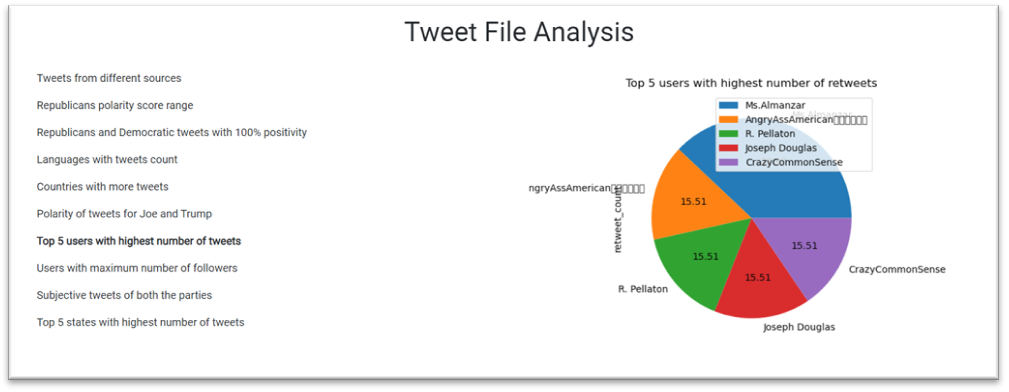
6). Comparing the Polarity of Tweets for Joe and Trump using “Line Graph”



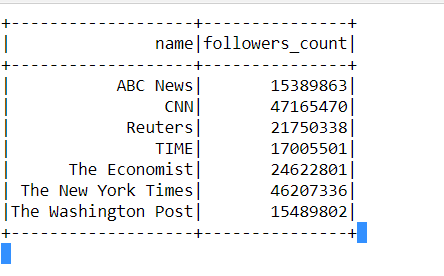


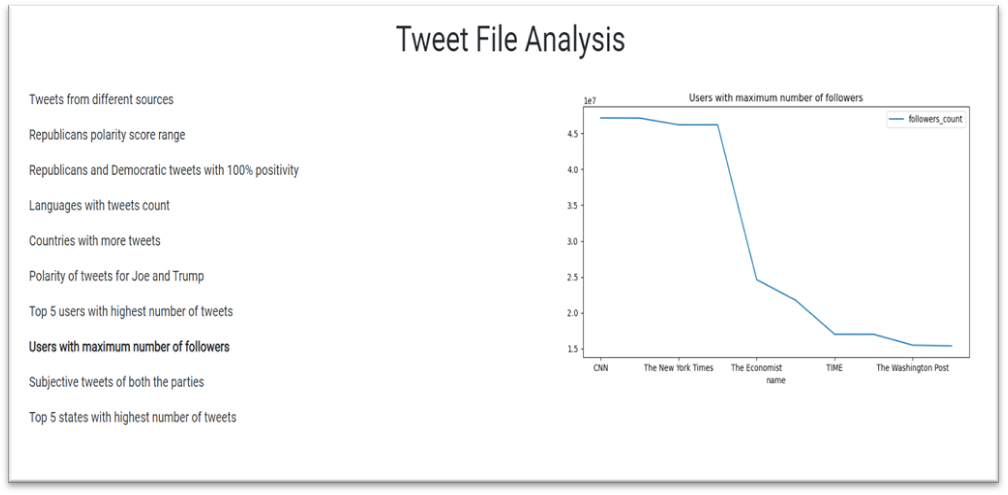
7). Top 5 Users with highest number of tweets from the collected data.



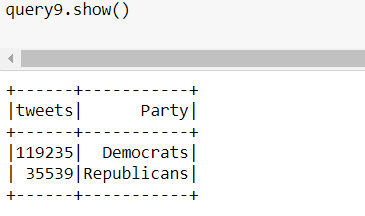


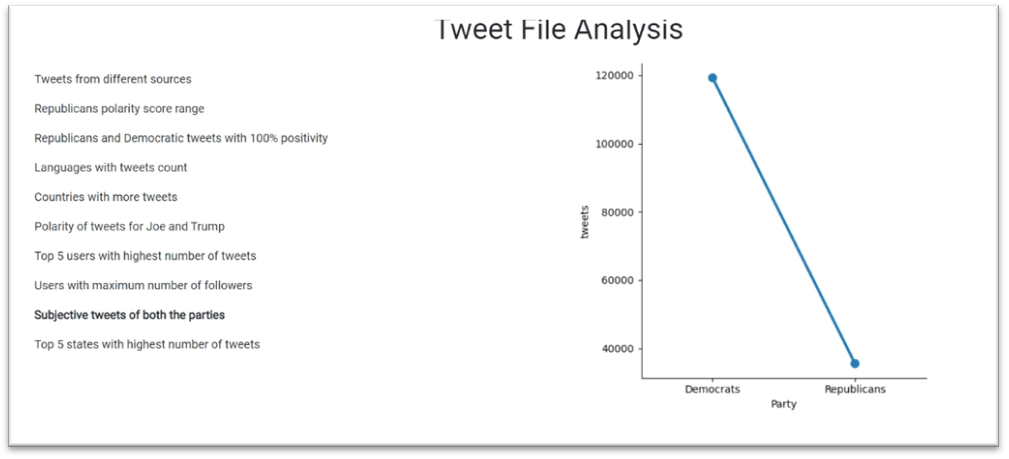
8). Top 5 users with maximum number of followers using “Line Graph”



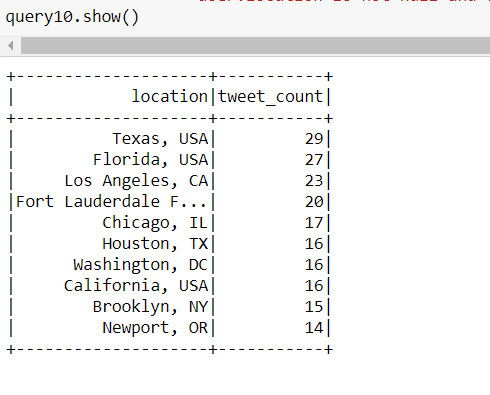


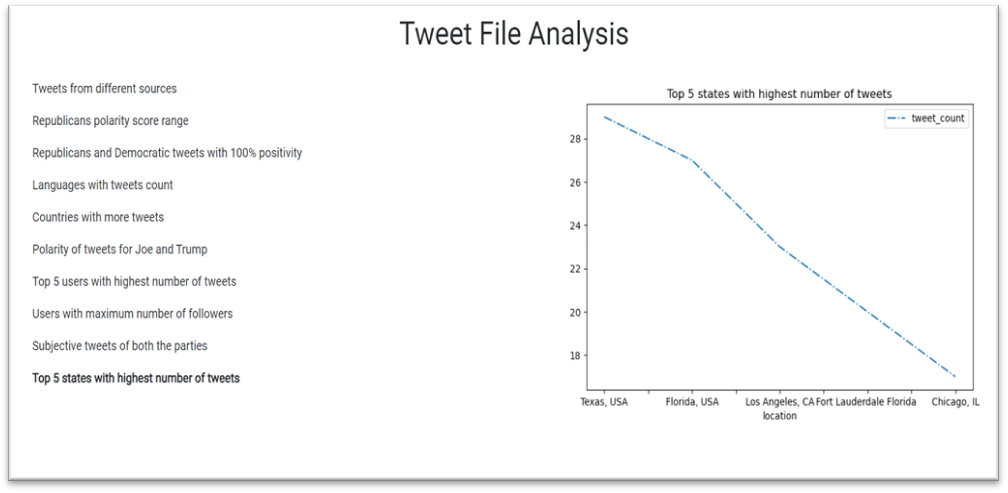
9). Subjective tweets of Democratic and Republicans parties.





10). Top 5 states in United states with highest number of tweets.





**Project-Management:**

**Implementation-status-report:**

**Work-Completed:**

* Twitter data batch download & documented his part-Jagadeesh Maroju
* Data Cleanup and project implementation plan & documented his part- Paul Gomes
* HDFS import and Hue Visualization and Hive import & documented his part- Hari Raju
* Hive Queries and report- Praveen Poluri
* create spark dataframes on CSV file and implement various transformations and actions on dataframes.
* implemented Spark sql queries and visualized the data queried.
* Implemented Sentimental analysis using textbolb and visualized the results using Flask
* Created a web application to view the visualizations.

**Work to be completed:**

* Loading data into solr and implement queries on the data in solr.

**References:**

<https://spark.apache.org/docs/latest/api/scala/index.html#org.apache.spark.package>

<https://spark.apache.org/docs/latest/api/python/index.html>

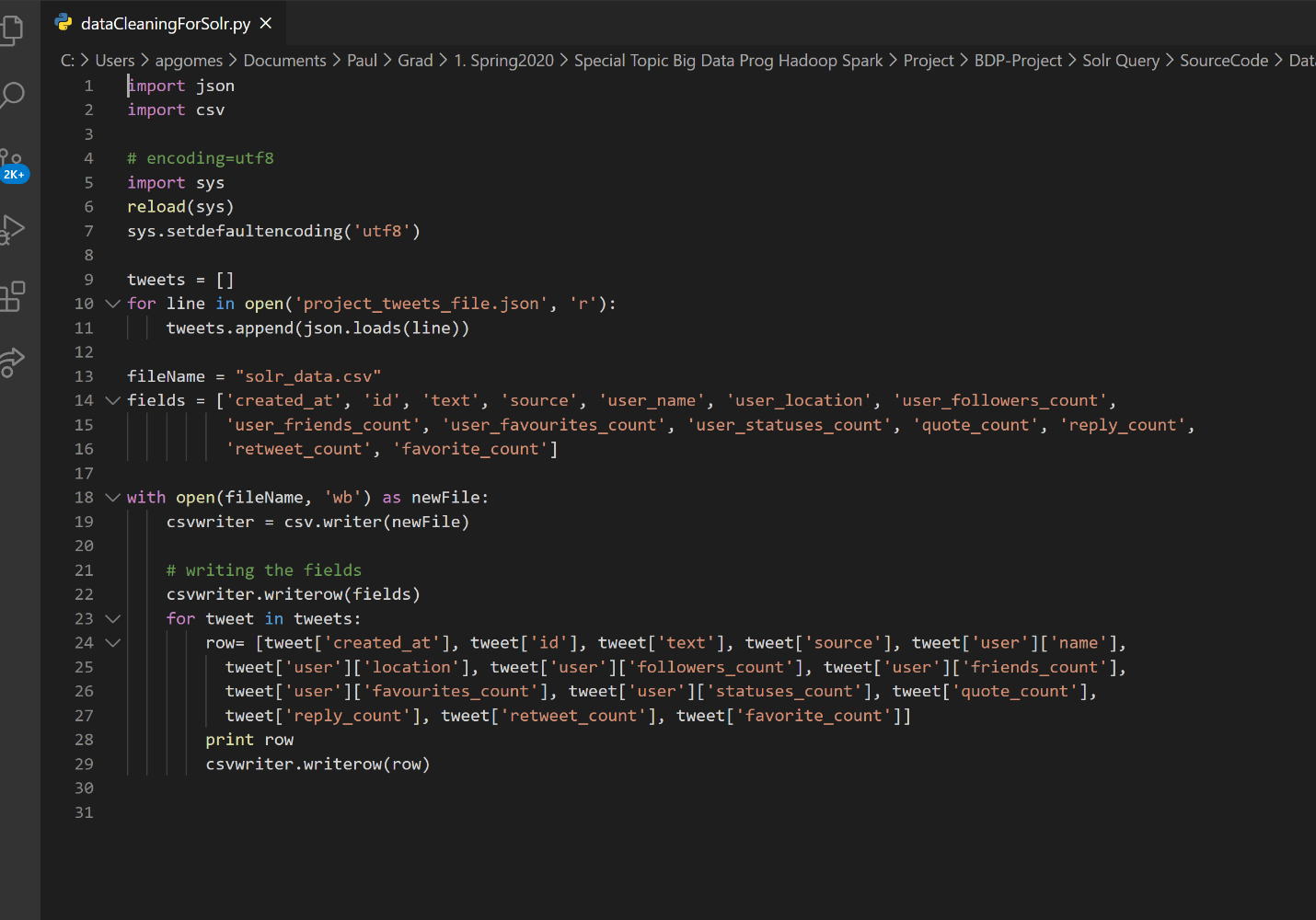
## Increment-3:

## Solr

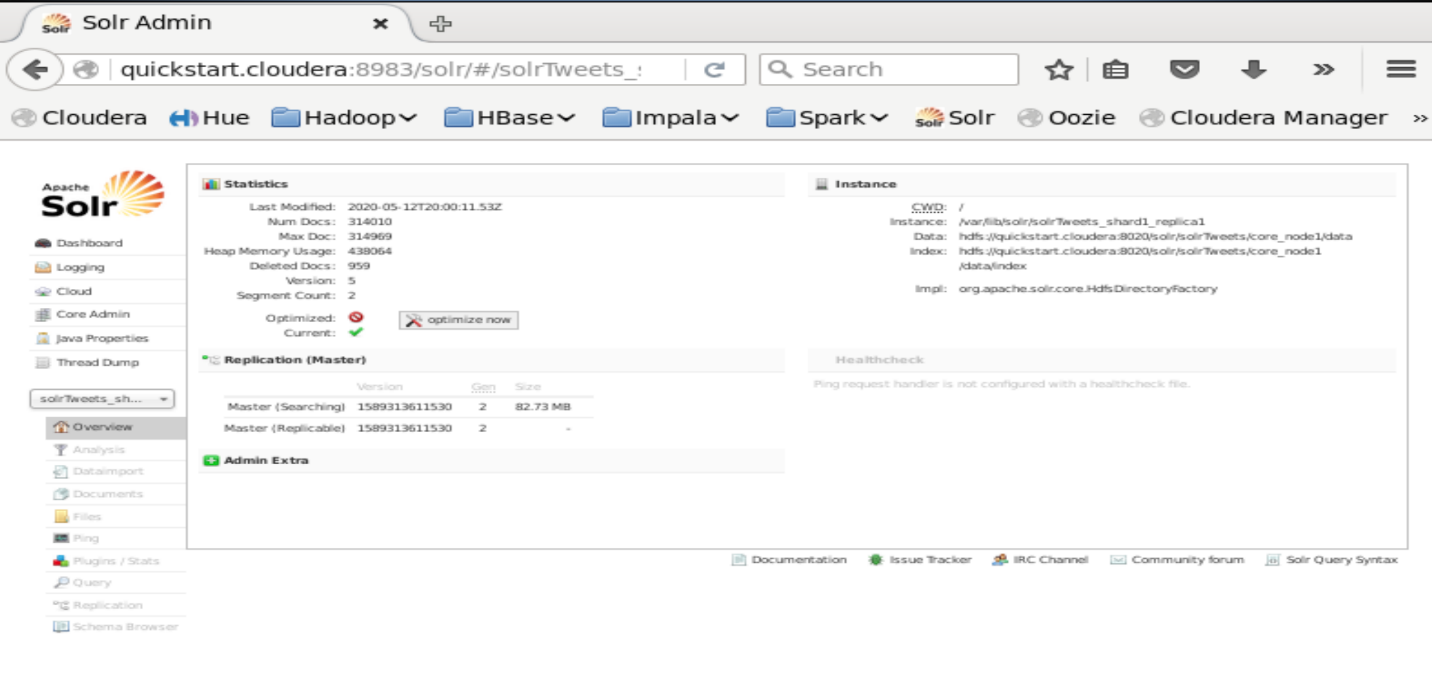
### Introduction:

In this section I imported the collected tweets data into solr and performed solr query. Collected data had so many extra fields and it was also raw data. As a result, I cleaned the data before importing it to solr.

### Data cleaning

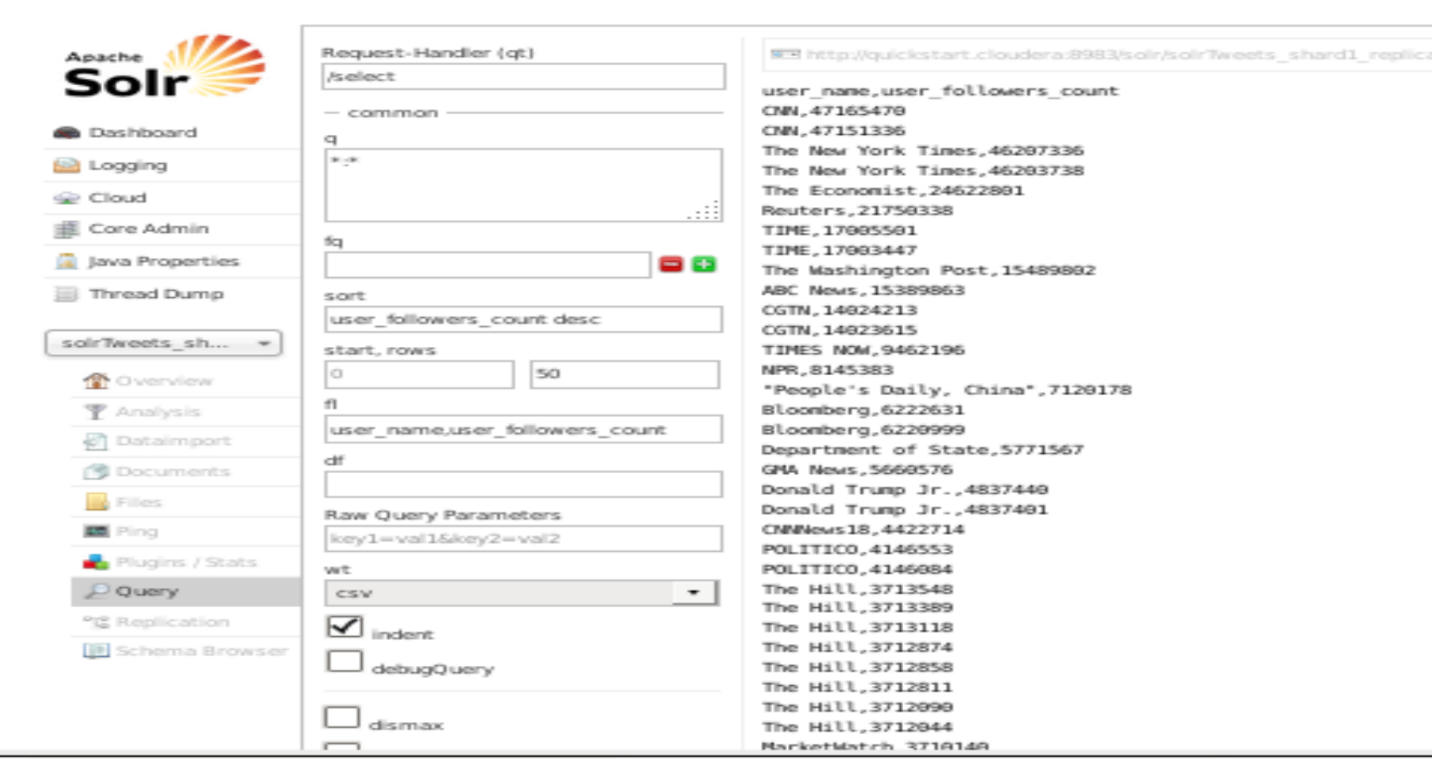
I read the json file of all the collected tweets and created a csv file with the necessary columns and data. New csv file have the following columns regarding a tweet - created\_at, id, text, source, user\_name, user\_location, user\_followers\_count, user\_friends\_count, user\_favourites\_count, user\_statuses\_count, quote\_count, reply\_count, retweet\_count, favorite\_count. 

### Solr query:

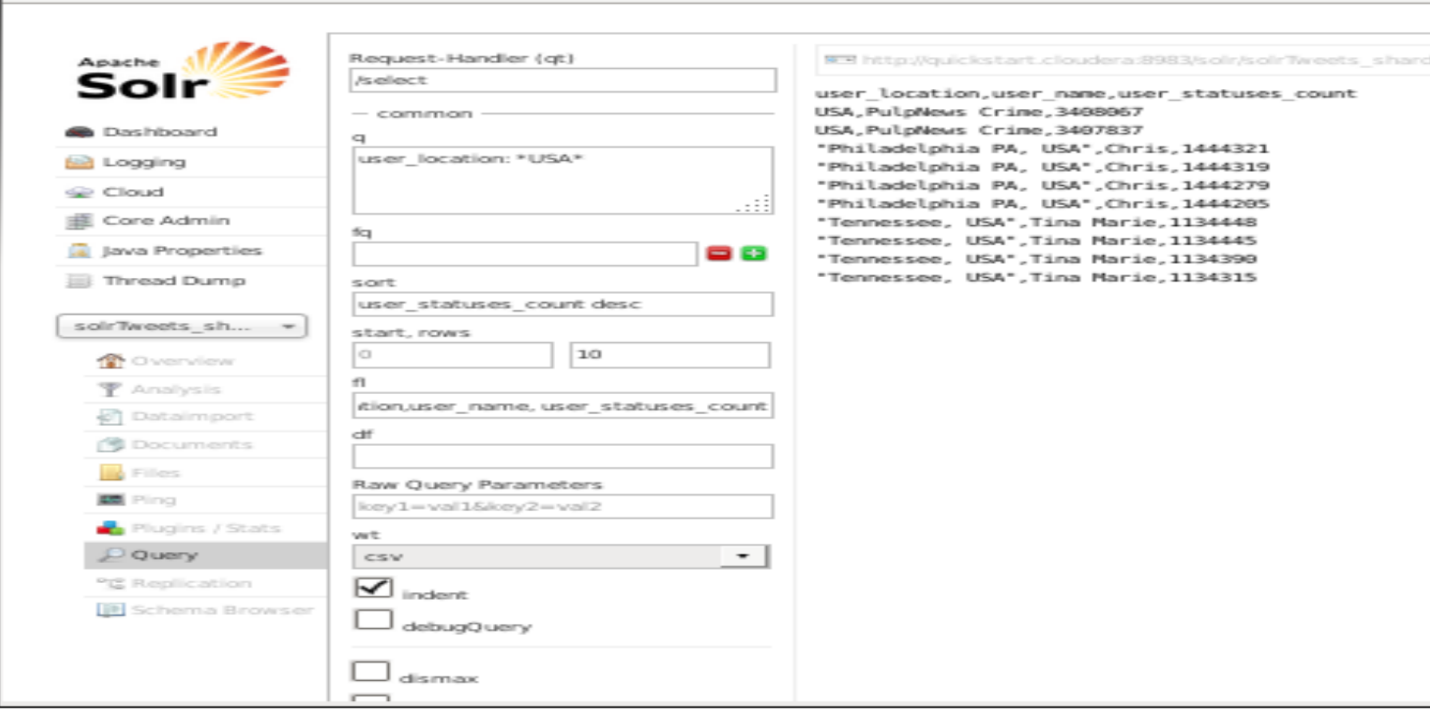
* We created schema config using solrctl instancedir --generate /tmp/solrTweets command.
* We edited the schema and created new fieldTypes for all the columns I have in the dataset.
* We uploaded the contents of instance directory to zookeeper using solrctl instancedir --create solrTweets /tmp/solrTweets
* We created a new collection using solrctl instancedir --create solrTweets
* We imported the data into solr collection using curl command. curl 'http://localhost:8983/solr/solrTweets/update?commit=true' --data-binary @tweets.csv -H 'Content-type:application/csv' 

**Queries:**

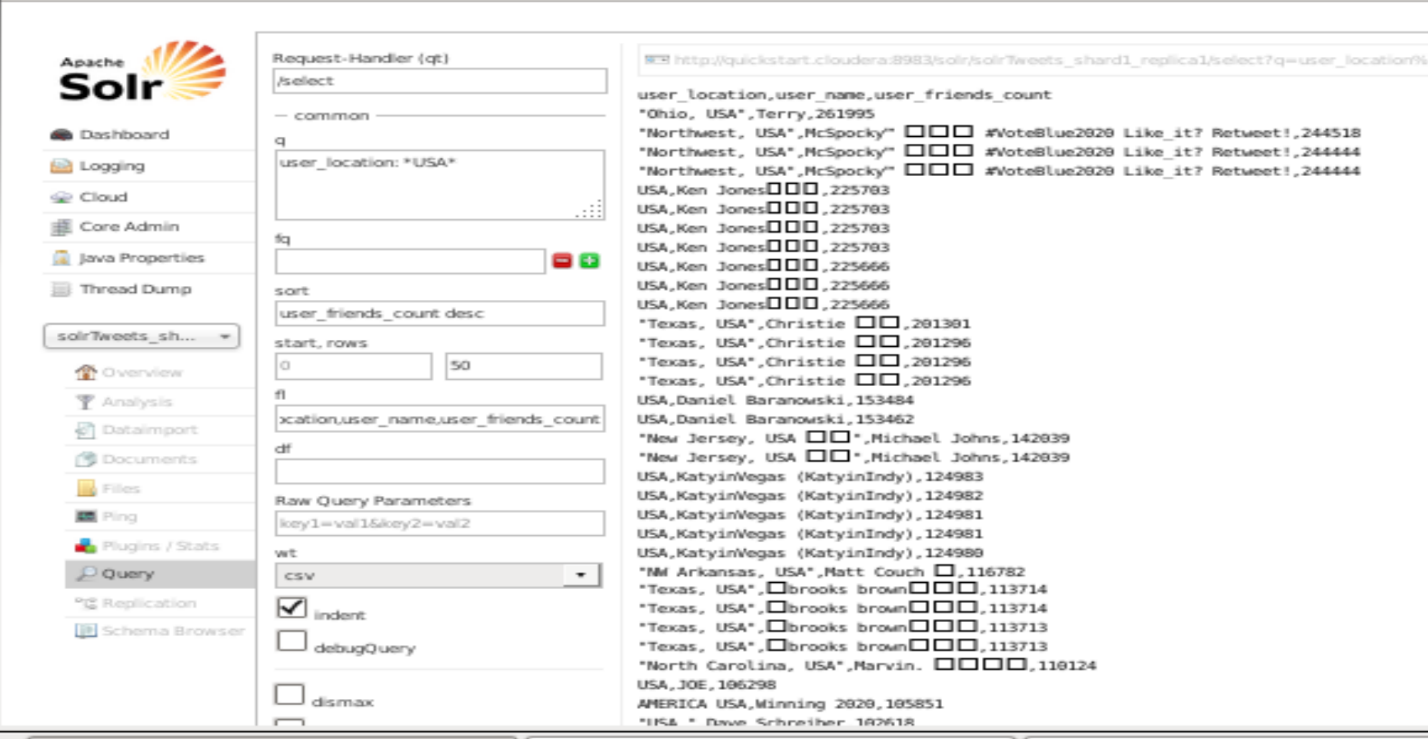
1. Top 10 users with the most followers



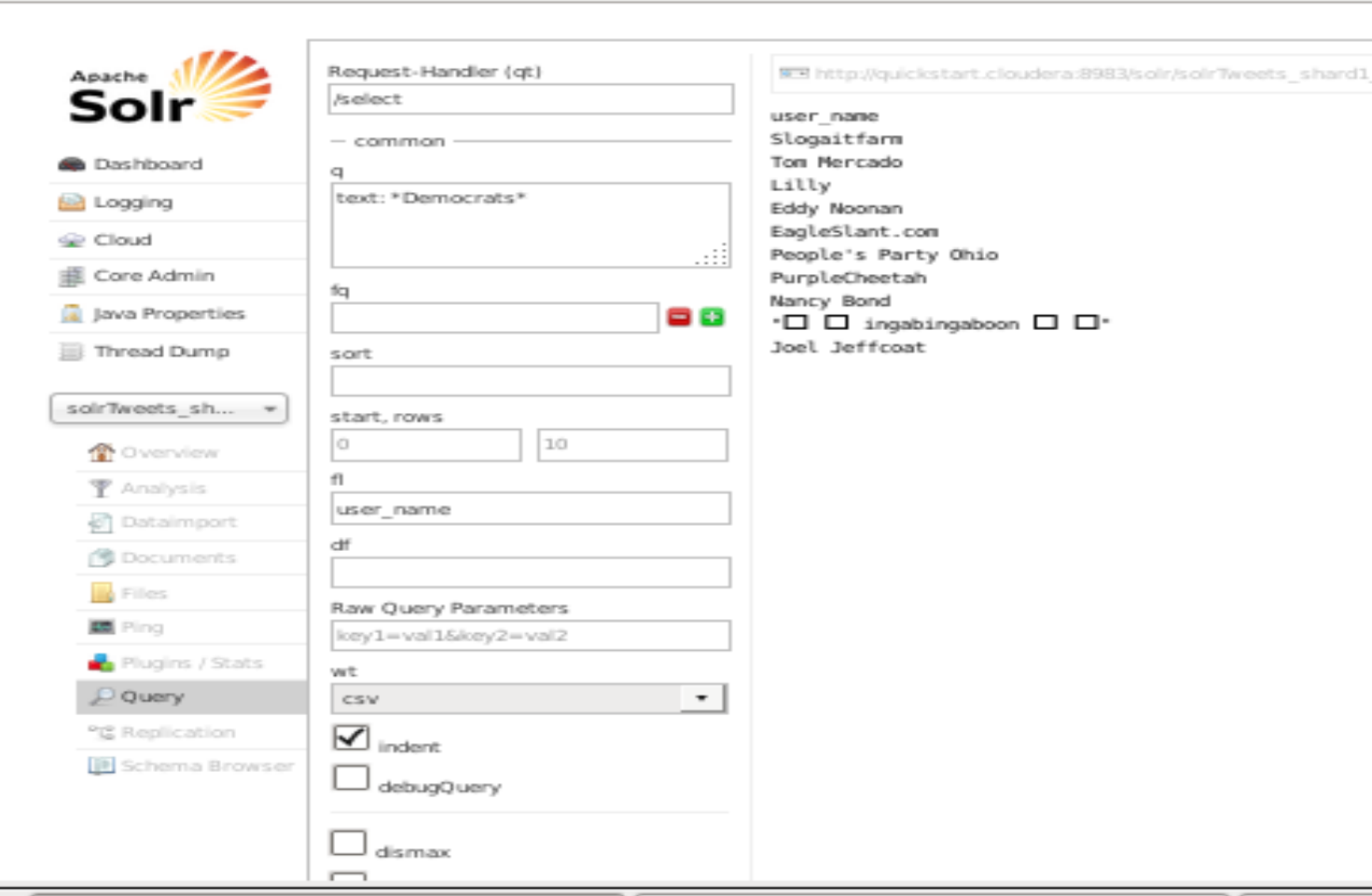
1. Top 10 users in the USA with most status count



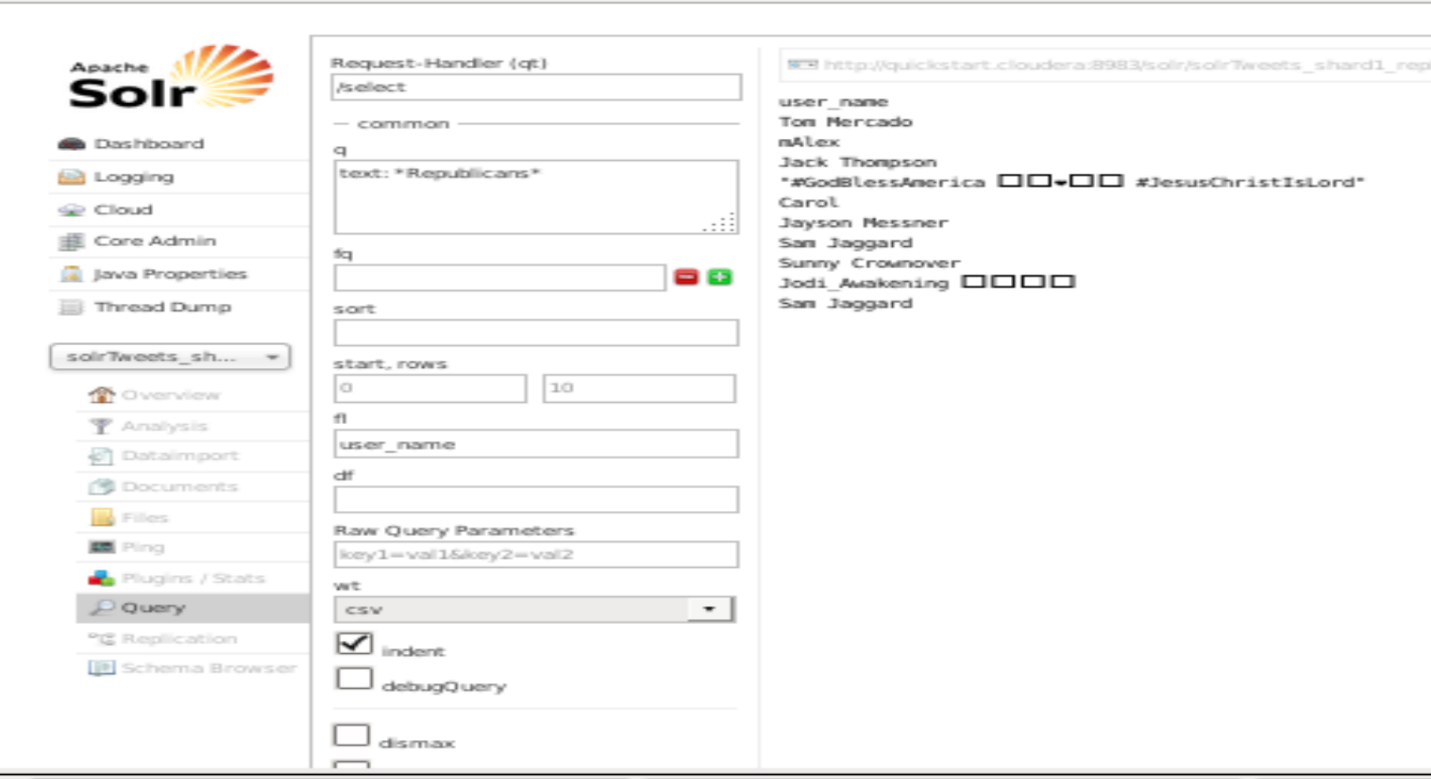
1. Top 50 users in the USA with most friends



1. Users who teweeted about democarts



1. Uesrs who tweeted about republicans



## Graphframes:

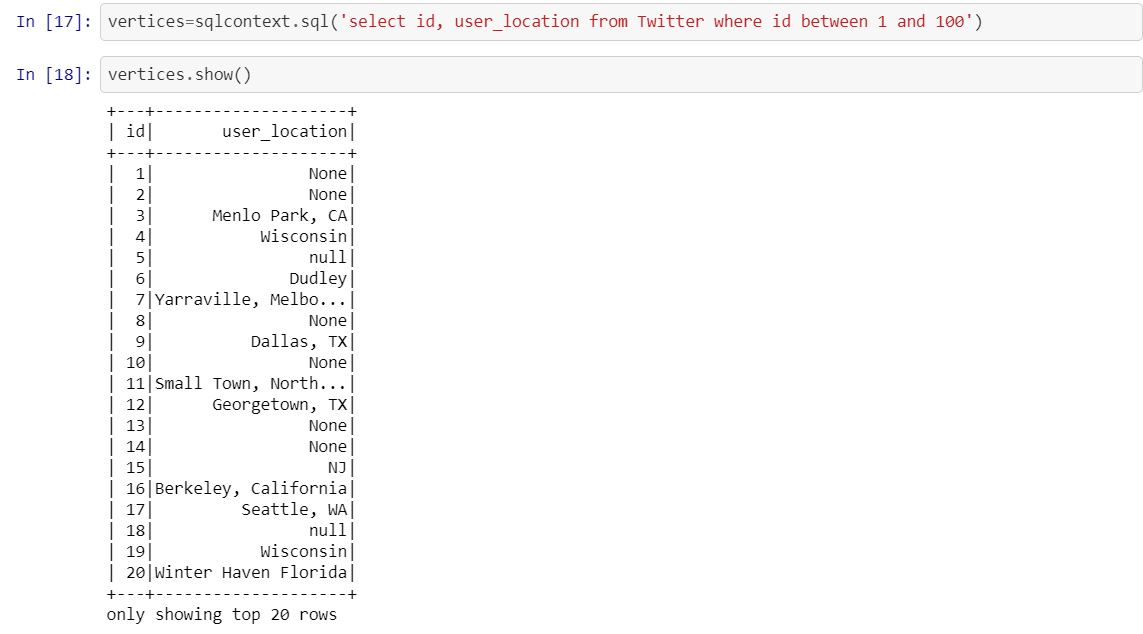
* Imported all the dependencies needed for graphframes and Pyspark.
* Created Spark, SqlContext by importing required libraries.



* Created a dataframe on the twitter data extracted using dataframe reader API and registered a temptable on the dataframe, for Schema so that we can transform, extract the data to create new dataframes using SQL on dataframes. Displayed all the attributes of data extracted using head() of dataframe API.



* Created vertices for the graphframe using id and user\_description columns from the dataframe created on the twitter\_data file.



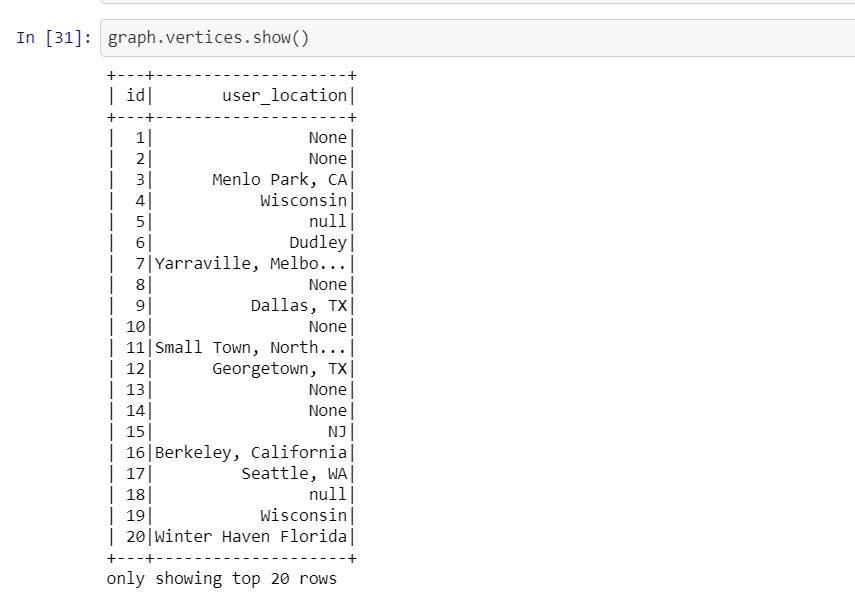
* Created Edges for the graph using username as source and user\_followers as destination from the dataframe created on the twitter\_data file.



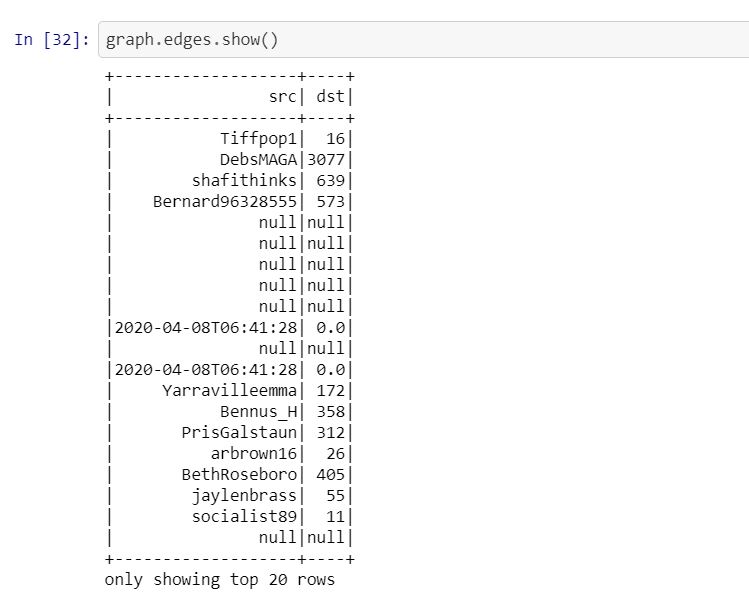
* Imported the libraries required to create a graphframe, Created a graphframe using the vertices, edges created using the GraphFrame function.



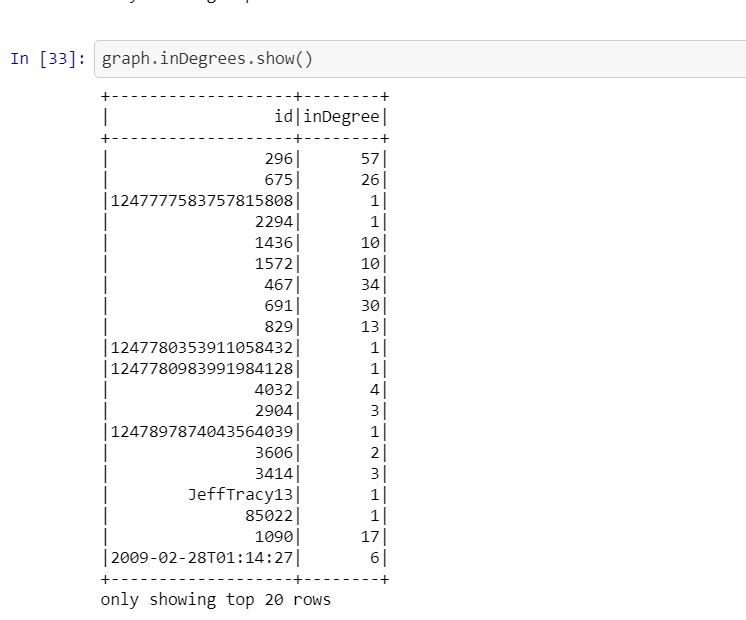
* Displayed the graph vertices created using show() of graphframes API.



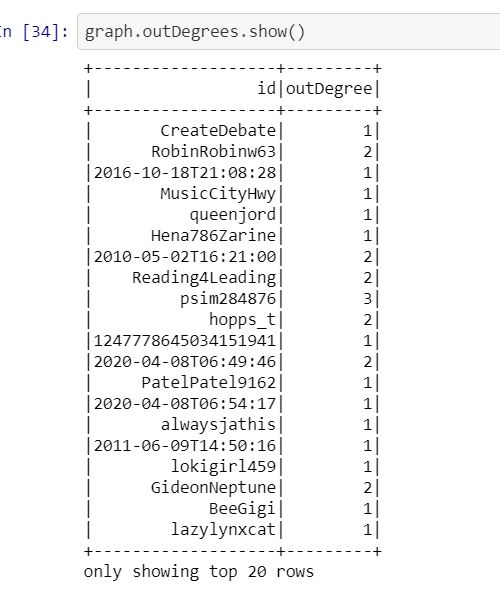
* Displayed the graph Edges using show() of Graphframe API.



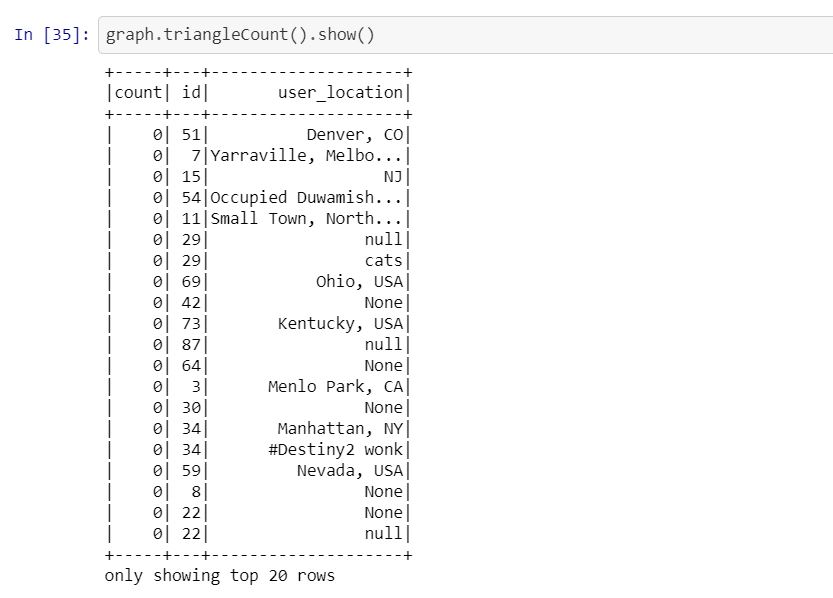
* Calculated indegree using inDegree API of graphframe and displayed using show().



* Calculated outdegree using outDegree API of graphframe and displayed using show().

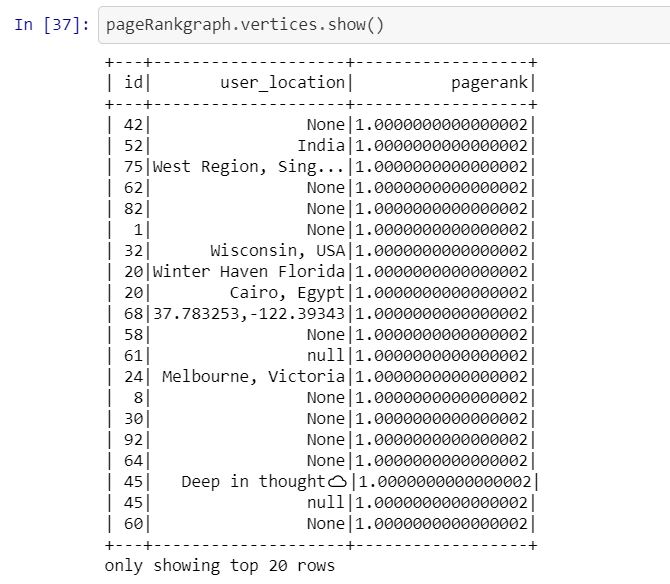


* Calculated triangle count using trianglecount() of Graphframes and showed the triangle count.



* Calculated pagerank for graphframe created and displayed it.

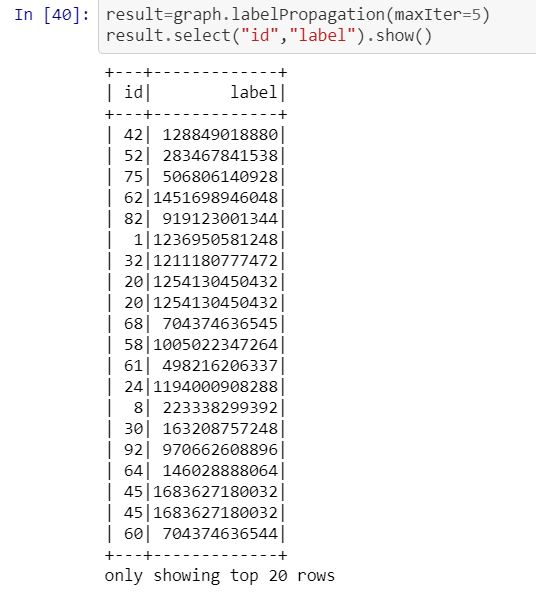




* Saved the graphframes vertices as a parquet file in a vertices folder using write.Parquet() of spark.
* similarly saved graphframe edges as a parquet file in a edges folder using write.Parquet() of spark.



* Assigned labels for the id column of the graphframe using labelpropagation() , These labels can be used for classification in Machine Learning.



## Project Management:

### Implementation status report:

### Work Completed:

* Twitter data batch download & documented his part-Jagadheesh Maroju
* Data Cleanup and project implementation plan & documented his part- Paul Gomes
* HDFS import and Hue Visualization and Hive import & documented his part- Bollepalli Sai Prasad Raju
* Hive Queries and report- Praveen Poluri
* create spark dataframes on CSV file and implement various transformations and actions on dataframes.- Jagadheesh Maroju
* implemented Spark sql queries and visualized the data queried. - Jagadheesh Maroju
* Implemented Sentimental analysis using textbolb and visualized the results using Flask- Bollepalli Sai Prasad Raju
* Created a web application to view the visualizations. - Bollepalli Sai Prasad Raju
* Loaded data into solr and implemented queries on the data in solr. - Paul Gomes
* Created Dataframes and Graphframes on the twitter data, implemented various features of Graphframes. - Praveen Poluri.

## Contributions:

* Jagadheesh Maroju -25%
* Bollepalli Sai Prasad Raju-25%
* Gomes Paul – 25%
* Poluri Praveen-25%

## Conclusion:

Collected Twitter data using hashtags #Democrats, #Republicans and loaded into hadoop and ran hive queries to get information on particular topic, exported tweets file to solar and queried using it, implemented sentimental analysis, built dataframes on the data, queried using sparksql, created graphs on the datframes and implemented various operations using Graphframes.

## References:

* <https://lucene.apache.org/solr/resources.html#tutorials>
* <https://databricks.com/blog/2016/03/03/introducing-graphframes.html>
* <https://towardsdatascience.com/graphframes-in-jupyter-a-practical-guide-9b3b346cebc5>