Twitter Analysis Project report

Submitted by:

Manoj Prabhakar Ejjirothu – 16203317

Pradeep Chaitanya Tirumalsetty –16208316

Ragunandan Rao Malangully – 16207927

Submitted on:

12/2/2015

Introduction:

This project was complete in two parts. The first part was developed in IBM Bluemix using the in- built features of IBM Bleumix. This phase dealt with collecting tweets using an application that collected tweets based on a topic. These tweets were then pushed to DashDB to persist them for future analysis. This Dash DB store was passed as a data store to the Apache spark instance in the IBM Bluemix. Spark SQL queries were run on the data set to perform analytics and the resultant data was saved in the Object storage in IBM Bluemix. This data was visualized using the Jupyter notebook present in IBM Bluemix. In the second part we have collected queries using a Java application. The tweets were saved as a JSON file. Then we used Scala to interact with the local spark server to run queries based on user input. This Scala program was accessed through a Java rest service. This service was accessed from the client side to get real time data analysis in JSON format. The resultant data was then displayed in the form of various visualizations using Plotly.js. Angular JS was used to query the Java service which in turn interacted with the spark processing process to perform analytics and return the data in JSON format. For the image analysis scenario we had created a service that interacted with the IBM Alchemy API to perform analysis on provided image.

Technology used:

1. Java REST API framework for the backend services

2. Ionic framework for the front end application

3. IBM Bluemix Watson services for image analysis

4. Spark framework for running the Spark SQL queries.

5. Plotly.js - Charting library for visualization.

Data Analytics service Apache Spark-S7

We have used the above service in the Bluemix for the visualization of first 4 queries mentioned in the later part of this report.

SCALA NOTEBOOK: Connected the dashDB database to the Scala service and ran the queries on the table ALLSTARS Table and stored the results in a file with *.Parquet* file format.

IPYTHON NOTEBOOK: Using IPYTHON notebook we have accessed the *.Parquet* file from the above section and plot the graphs for all the results from the queries.

References:

We have collected the tweets from a tool twitter loader-1111.

https://hub.jazz.net/project/torsstei/Twitter-Loader/overview

GitHUB URL :

https://github.com/ragunandanrao/PB-Project-Fall-2015

Tweet File DropBox Link

https://www.dropbox.com/s/tsgwwt21ykawg2z/TweetSet.json?dl=0

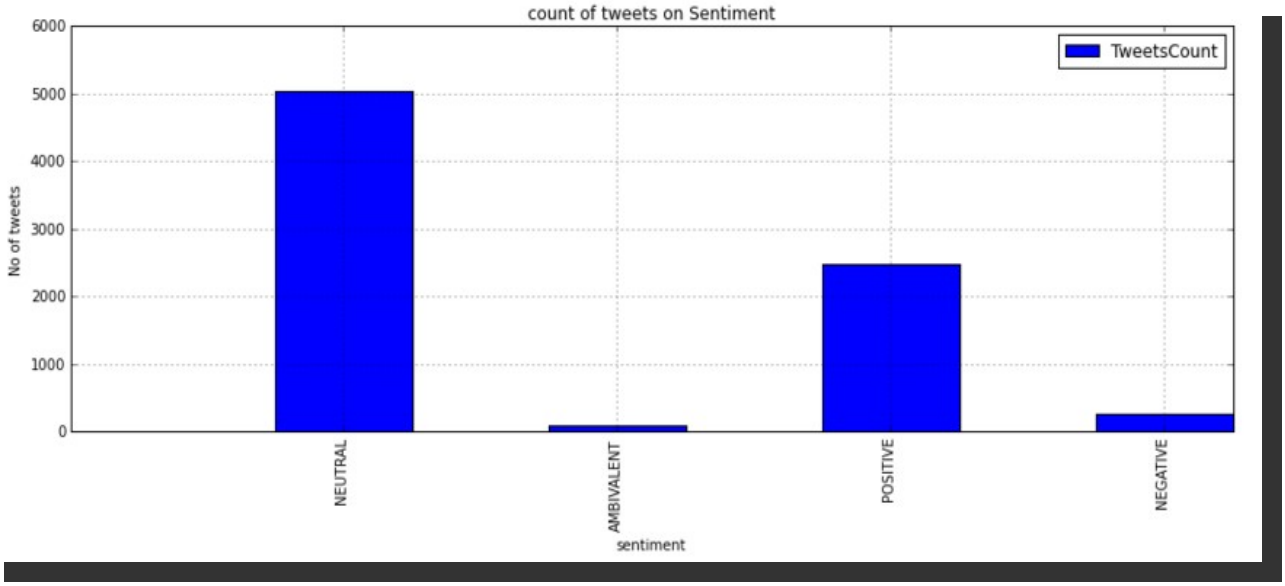
Bluemix URL

https://cdsx.ng.bluemix.net/data/notebooks/f1fce3e9-8d5e-4397-84a6-e9747ac18c15?tenant=f87263a8-ab7b-479f-8c3a-a1878c30cc8d

Queries:

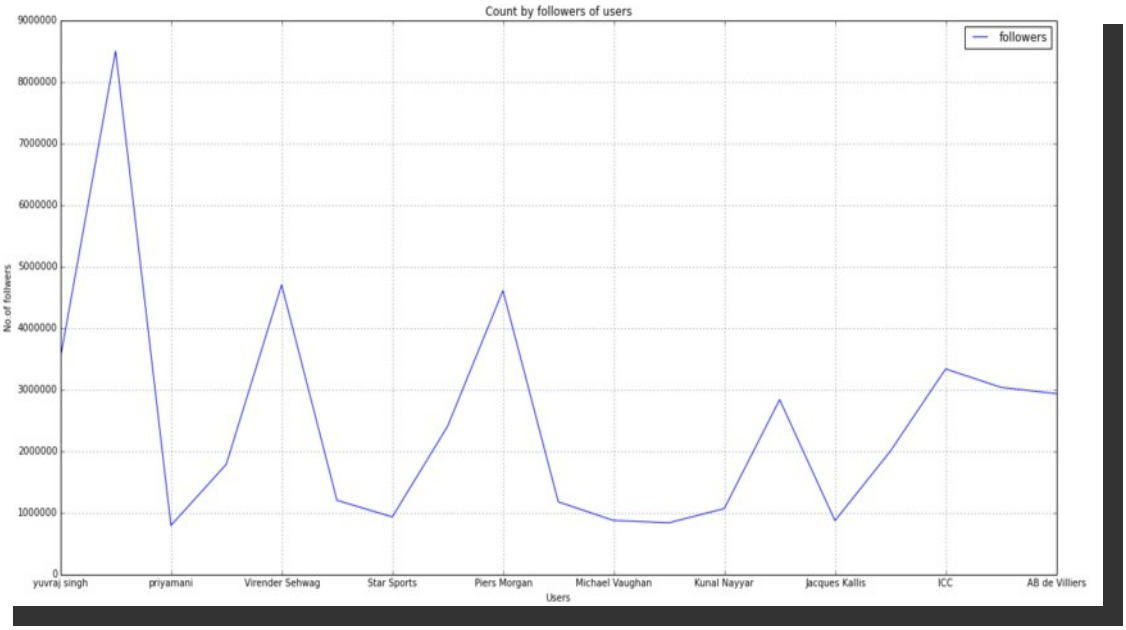
QUERY1: *select smaSentiment, count (*) as TweetsCount from tweets group by smaSentiment*

The above query analyses all the tweets and categorizes all the tweets into 4 categories POSITIVE, NEGATIVE, AMBIVALENT and NEUTRAL



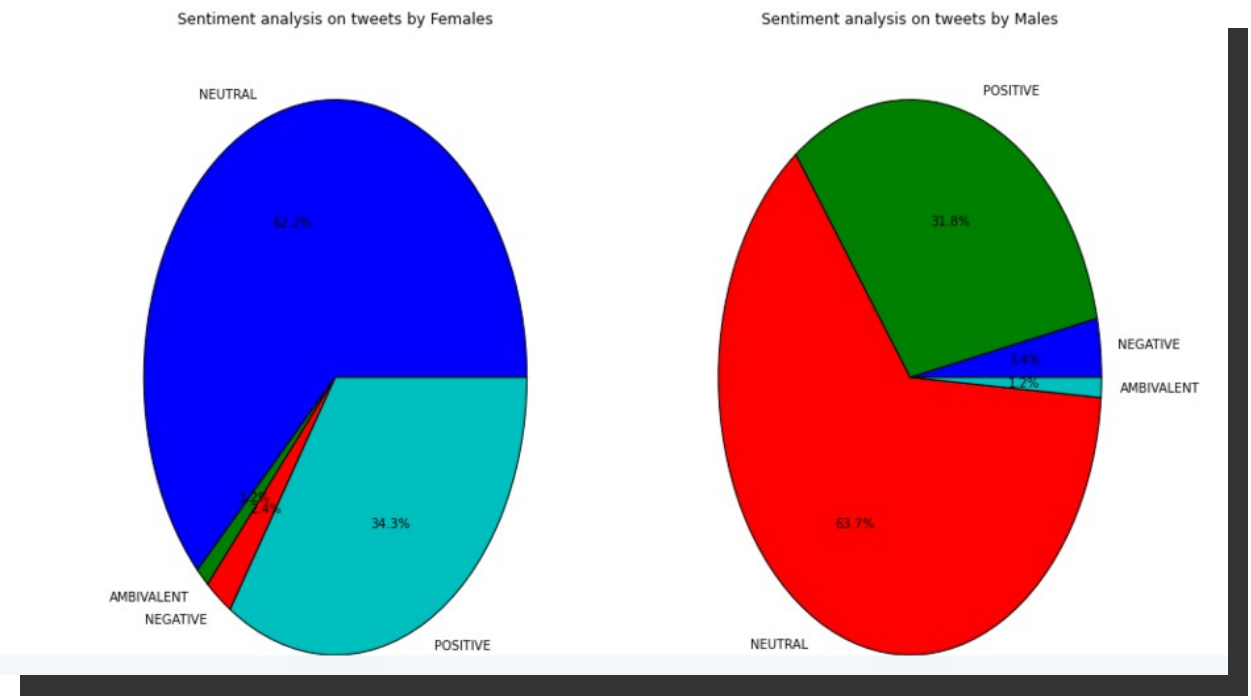
QUERY2: *select userDisplayName, max (userFollowersCount) as followers from tweets GROUP by userDisplayName order by followers desc limit 20*

The above query lists the maximum count of followers for a user among the collected tweets and take the first 20 randomly selected users



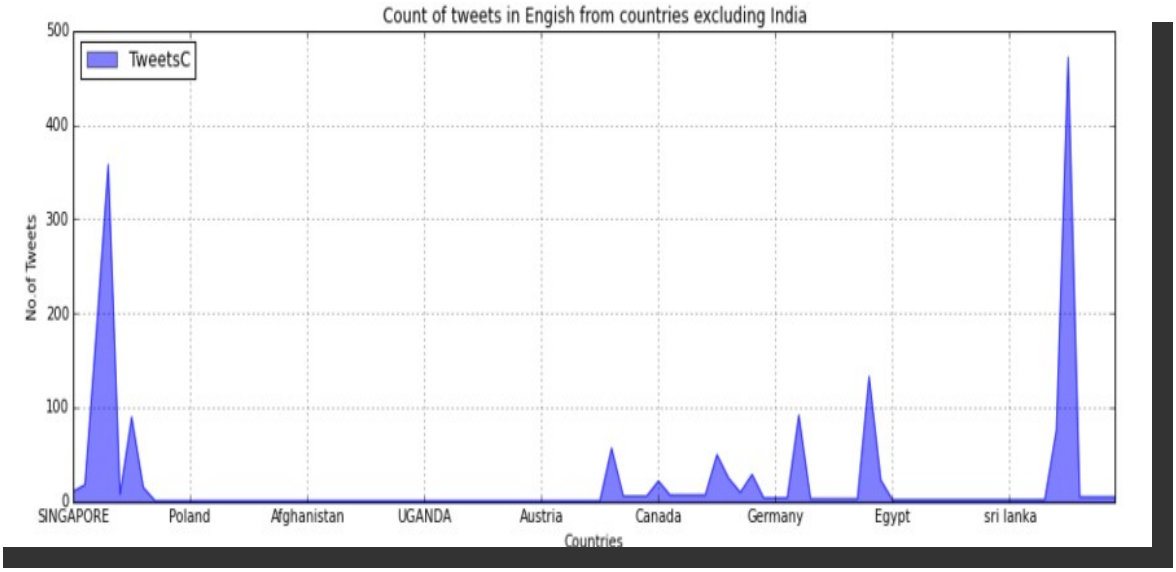
QUERY3: *select smaSentiment, smaAuthorGender, count (*) as count1 from tweets where smaAuthorGender in ('male','female') group by smaSentiment, smaAuthorGender order by smaAuthorGender*

The above query counts the total number of tweets based on the sentiment and again groups them based on the gender attribute of the tweet.



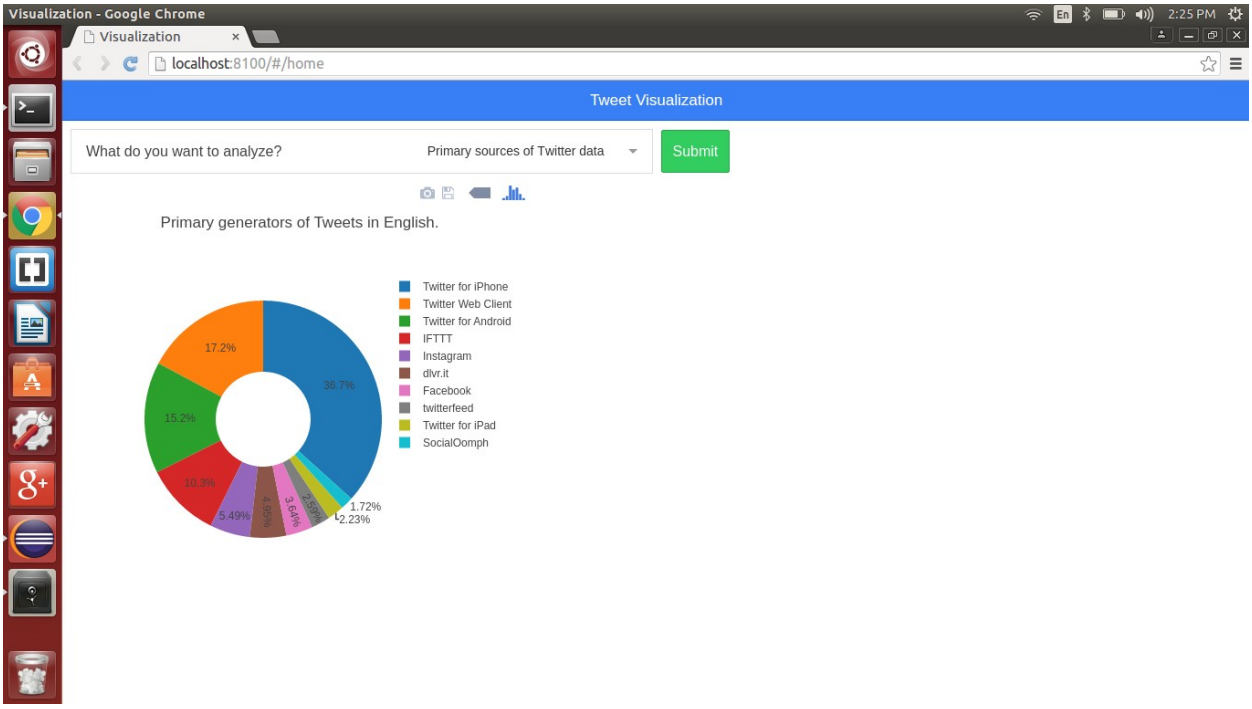
QUERY4: *select distinct smaAuthorCountry, count (*) as TweetsC from tweets where userLanguage in ('[en]') and smaAuthorCountry not in ('India','INDIA','india', null) group by smaAuthorCountry*

The above query counts the number of tweets which are in English and groups them based on the country attribute of the tweet.



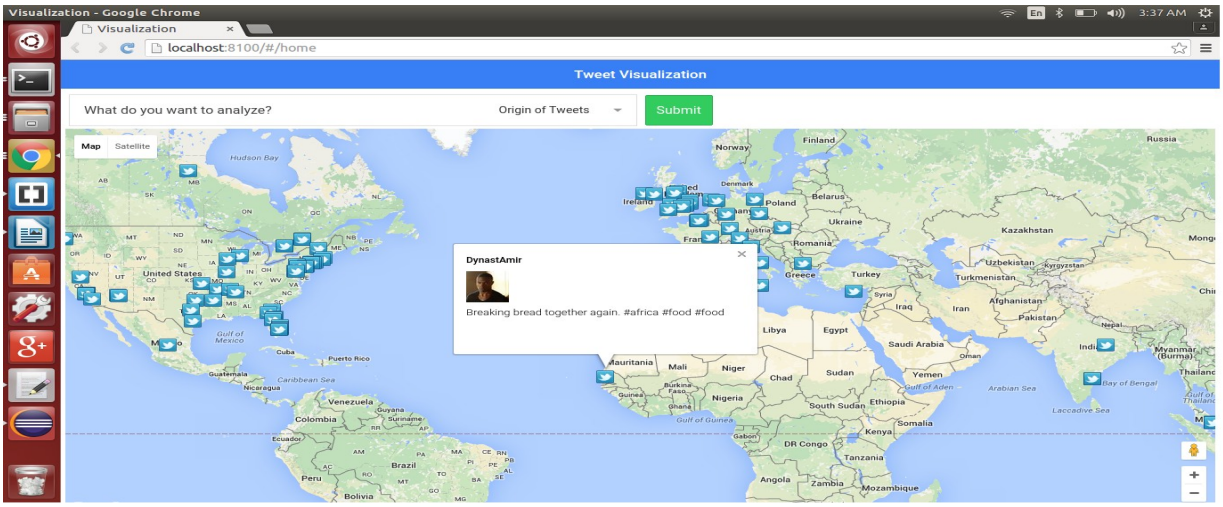
QUERY 5*: SELECT source, COUNT(*) AS total_count FROM Tweets WHERE source IS NOT NULL and user.lang = 'en' and (retweeted_status IS NULL or retweeted_status='') GROUP BY source ORDER BY total_count DESC LIMIT 10.*

Analyzing the devices used mostly to generate Twitter data. This query does a survey on what are the most common sources that generate Twitter data.



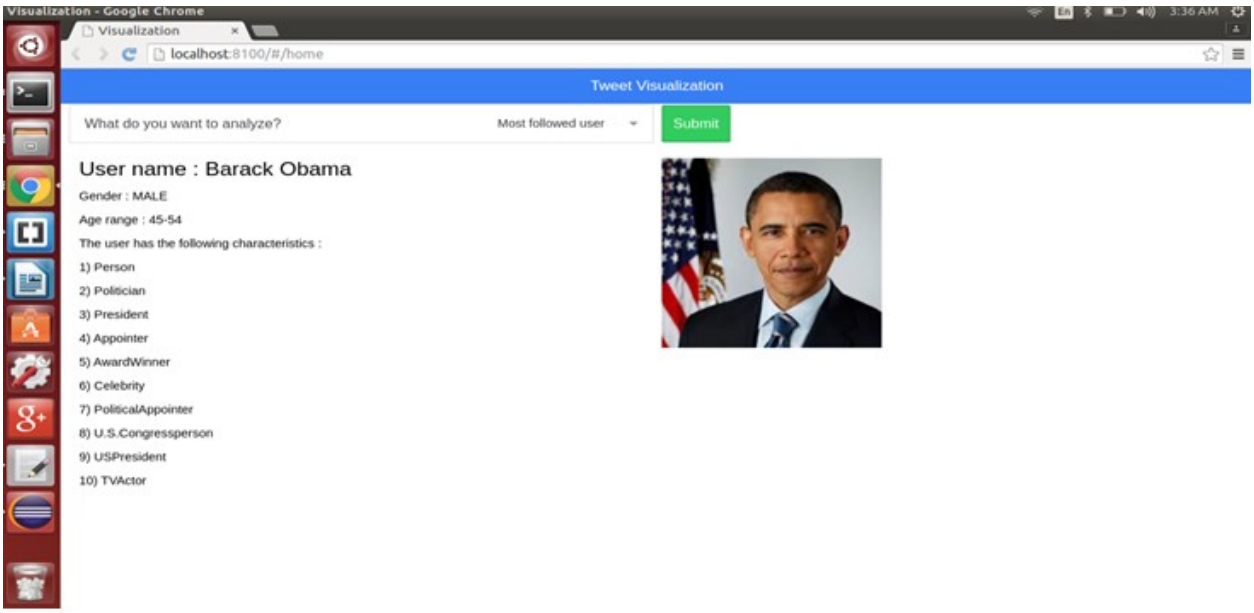
QUERY 6: *SELECT user.screen_name as name,text,user.profile_image_url as userimage, coordinates from Tweets where coordinates IS NOT NULL and user.screen_name IS NOT NULL.*

Analyzing the origin of set of Tweets. This query intended to analyze the geo coordinates of the user who tweeted a particular tweet. The location of the user was marked on map.



QUERY 7*: select user.screen_name as username, user.profile_image_url as userimage, user.followers_count as followers from Tweets order by user.followers_count desc limit 1.*

Analyzing the most followed user. The purpose of this query was to do an image analysis on the profile picture of the most followed user in the data set. Some of the information of the user was displayed.



QUERY 8: *SELECT user.time_zone as timezone, SUBSTR(created_at, 0, 10) as postedtime, COUNT(*) AS total_count FROM Tweets WHERE user.time_zone IS NOT NULL AND SUBSTR(created_at, 0, 10) in ('Sat Nov 28') GROUP BY user.time_zone,SUBSTR(created_at, 0, 10) ORDER BY total_count DESC LIMIT 5*

Analyzing the most active time zones for a particular day. This query dealt with categorizing the most active time zones that generated the highest number of tweets for a mentioned date.