Project Title:

Tweets Analysis with Spark

Team:

Code Monkeys

Team members:

Harshil Patel (8), Matthew Boerner (1), Stephanie Retzke (10) and Yong Zheng (14)

Increment 1 Report - Yong

Dataset

Each member of the team is asked to collect 100K English tweets with various tracks. The analysis for each member is done on their own set of tweets (100K). Due to the size of the dataset and our previous conversation, I will not upload this dataset to GitHub.

Detail Design of Features

For collecting data from Twitter, I am using tweepy package in Python. I wrote a python script to collect 100K English tweets with various tracks. With data I collected, I then used Spark and SparkSQL to perform 10 interesting queries on it.

Here are my queries:

Query 1:

Find polarity percentage for tweets' text field

Query 2:

Find the created year distribution for users

Query 3:

Find most popular login sources

Query 4:

Generate word cloud for hashtags' popularity

Query 5:

Count null and not null for geo tag

Query 6:

Get country code distribution from geo tag

Query 7:

Get top 10 country codes distribution from place tag

Query 8:

Message truncated percentage

Query 9:

User verified distribution and analysis

Query 10:

Analysis on friends_count and followers_count tags

Analysis

Query 1:

I want to use sentential analysis on the tweet to determine the polarity of each tweet then count how many tweets are positive, negative and neutral.

Query 2:

I want to see the created year distribution for the distinct users in my data set.

Query 3:

I want to what is the most popular login sources people used for Twitter

Query 4:

I want to generate a word cloud based on hashtags' popularity

Query 5:

I want to check the count for users with and without geo info (from geo tag)

Query 6:

I want to check the country code distribution for users with not null value for geo tag. In order to get actual country code from geo code, I will need to reverse the latitude and longitude into actual address and parsed out the country code.

Query 7:

I want to check the top 10 country codes distribution from place tag. Unlike geo tag, place tag will also address info already. However, place tag will not give us the extra location of a user. It only provides the coordinates info for the approximated location.

Query 8:

I want to check how many tweets are truncated.

Query 9:

I want to get the distribution of uses whose accounts are verified. Among the verified/unverified users, I want to check how many of them are and are not using default_profile.

Query 10:

I want to know how many users have more friends count than followers count, how many users have followers count then friends count, and how many users have same number for both friends count and followers count.

Implementation

Query 1:

In order to perform sentential analysis on the tweets, I begin by remove special characters from each tweets then use Python's package textblob to evaluate the popularity of each tweets.

Query 2:

In order to count the created year distribution for all users in my dataset, I will need to get the distinct users with their created year. Then I can get user count with year in the group by clause.

Query 3:

In order to find the most popular login sources, I will need to get the distinct user id and login source for each distinct user. Then I will write use substring function to parse the source

naem and put the query in a subquery. With the subquery, I can then put a query outside it to get the count of login source with source in the group by clause and order by count in decreasing order. Then I use limit 10 to get the top 5 records.

Query 4:

In order generate the word cloud based on hashtags' popularity. I first start by exploding the hashtags field then convert all of them to lower cases. Then I group the hashtag with individual hashtag and perform count on the hashtag and order by count in the decreasing order.

Query 5:

In order to count the occurance of null and not null for geo tag, I am using SUM with CASE statement to check the occurrence of nulls and use count on the geo tag to check the occurence of not null.

Query 6:

In order to count the country code distribution from geo tag, I will need to perform reverse the latitude and longitude into actual address under the help of Python's package geopy. I start by filtering out the rows where geo is null. THen use geopy to perform reverse lookup. Then I parsed the output to get the country code.

Query 7:

Other than use geo tag to get the country code distribution, Twitter also has place tag which also contains the country code. I start by filtering out the rows where place is null then I group the rows with column country_code and order by the count of each distinct country code in decreasing order and show the top 10 rows.

Query 8:

In order to check the percentage of tweets that are truncated, I group the rows by column truncated and use the count to divide by the total count of all rows in the table. This will give me percentage of messages that are truncated and not truncated.

Query 9:

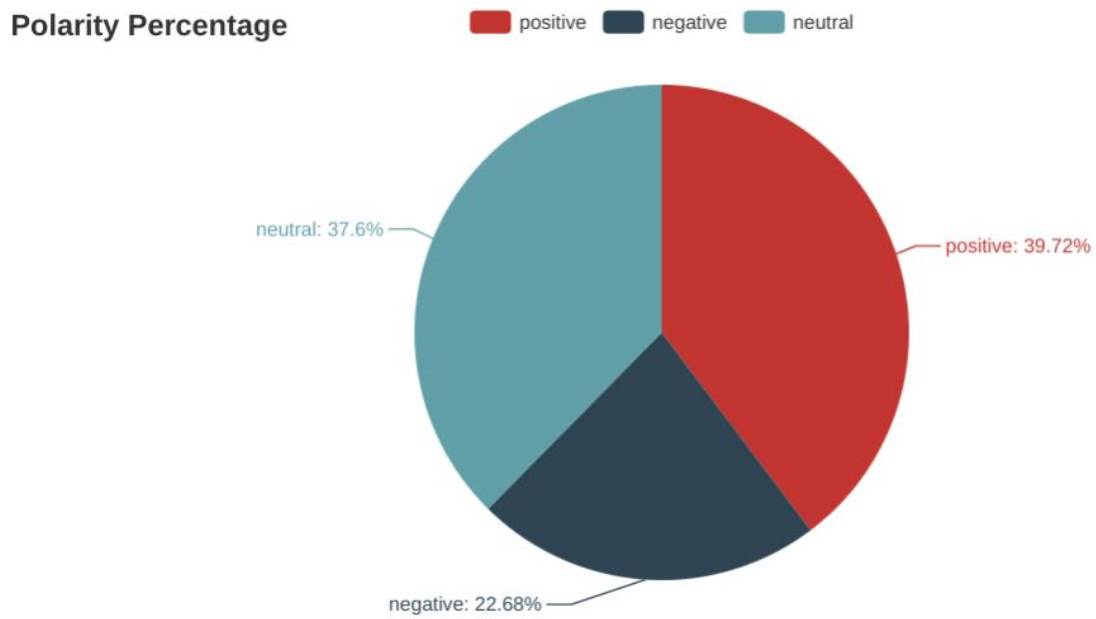
For user verified distribution and analysis, I group all rows by user.verified column to get the count of users that are using verified and unverified accounts. With this info, I then perform count on how many of each category are and are not using default profile.

Query 10:

For analysis on friends and followers count, I uses three separate queries to check how many users have friends count greater than followers count, how many users have friends count less than followers count, and how many users have same value for both friends and followers count.

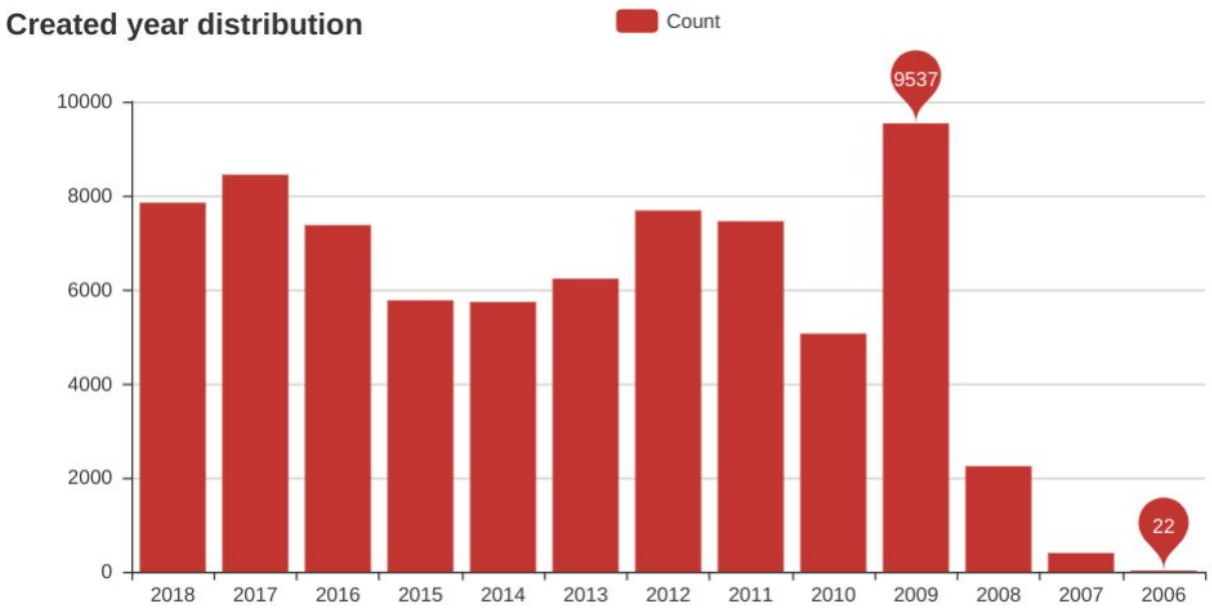
Preliminary Results

Query 1:



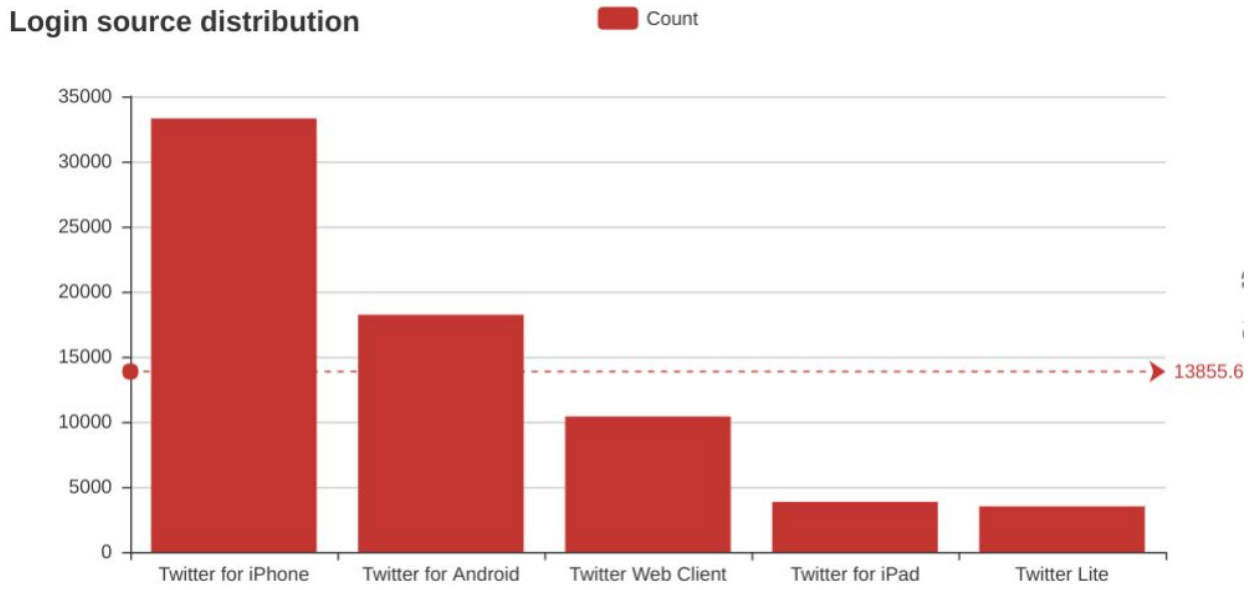
Among the 100K tweets I collected, 39.72% of the tweets are positive, 37.6% of the tweets are neutral, and 22.68% of the tweets are negative.

Query 2:



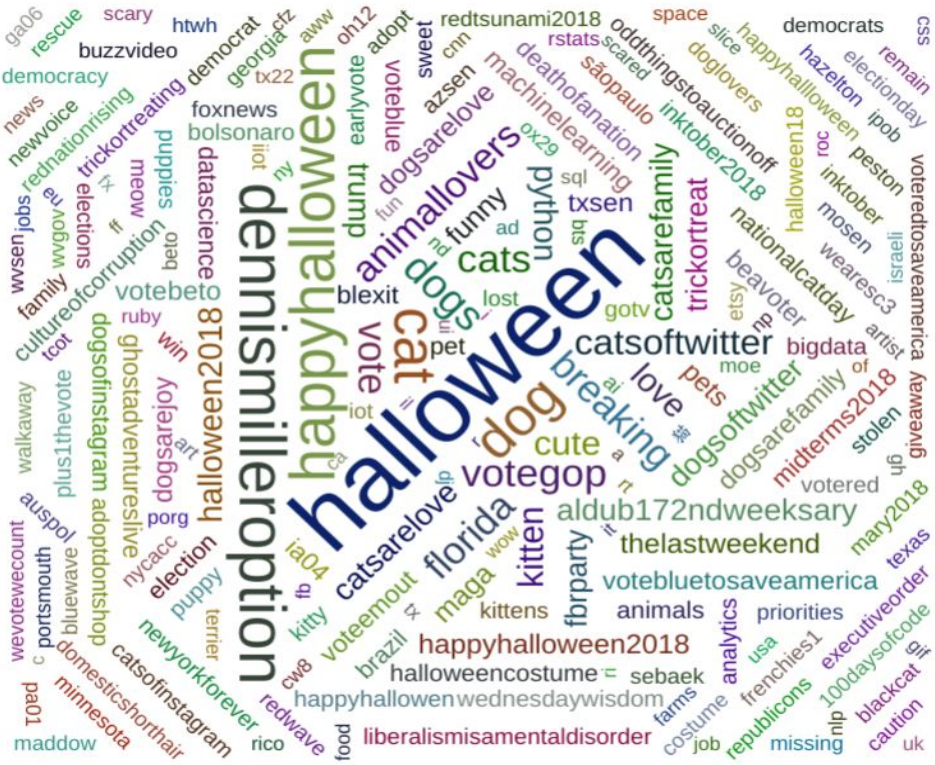
Among the 100K tweets, there are 9537 accounts were created at 2009 (highest) and only 22 of them were created at 2006 (lowest).

Query 3:



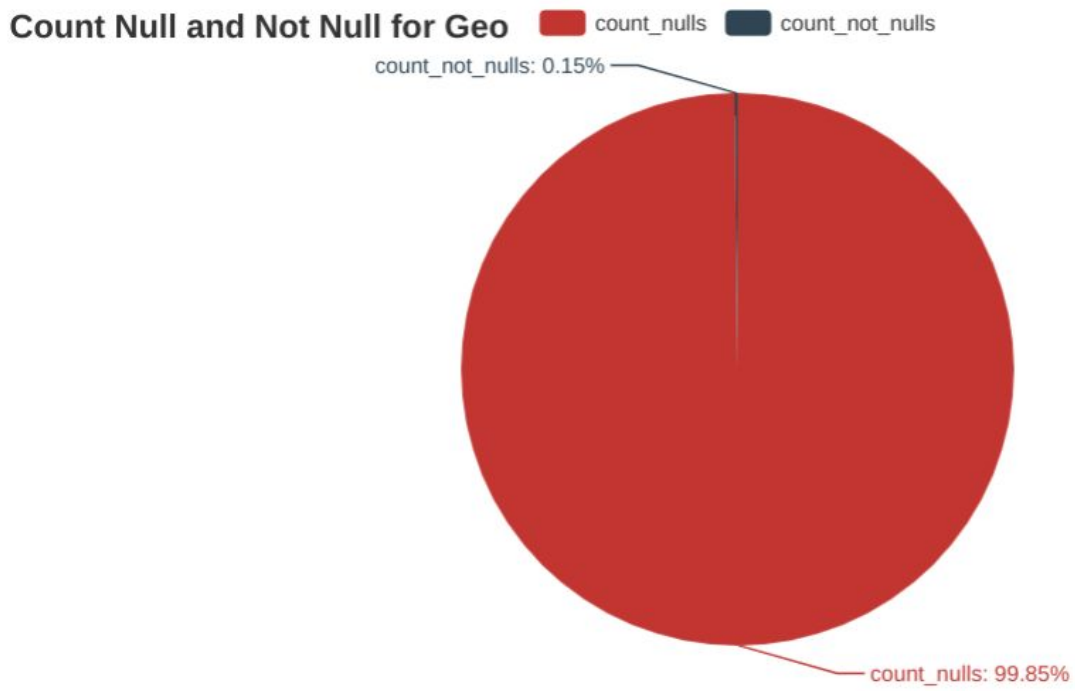
Among the 100K tweets, iPhone is the most popular, Android is the second popular, and Twitter web client is the third popular login sources for Twitter. The average login count among the top 5 login source is around 13855.

Query 4:



The size of the word in the word cloud is corresponding to the frequency of the hashtags. In this case, hashtag ‘halloween’ is the most popular hashtag people used while I was collecting these 100k tweets.

Query 5:



When check the proportion of users who has geo info enable (with geo enabled, we can get the extra location of a user), there are only 0.15% of the users have geo enable. The remaining 99.85% of the users have that setting turned off.

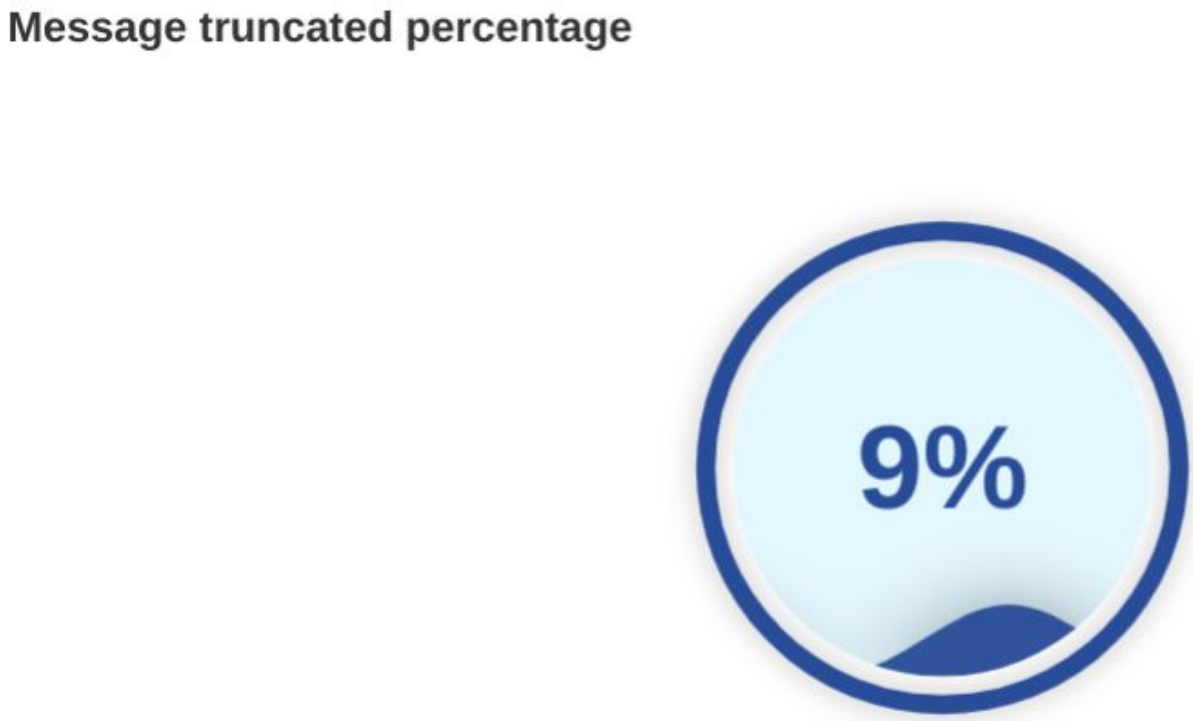
Query 6:

	Count
USA	119
Canada	11
New Zealand/Aotearoa	3
UK	3
Brasil	2
Philippines	2
ประเทศไทย	1
Chile	1
Ireland	1
Australia	1
대한민국	1
India	1

Among the users with geo enabled, most of them are from USA (119 users).
Query 7:

country_code	count
US	1147
CA	49
GB	47
AU	17
PH	10
BR	6
IN	6
JP	5
NZ	4
MY	4

Among the users with placed enabled (with place enabled, we can only get the approximated location of a user), most of them are from USA (1147 users).
Query 8:



There are only 9% of tweets that are truncated (when the message of a tweet is larger than the max size of a single tweet, Twitter will automatically truncate that tweet and send it with multiple smaller tweets instead of one large tweet).
Query 9:

verified	count
true	1221
false	98779

There are 1221 users with account verified and the remaining 98779 users have account unverified.

default_profile	count
true	218
false	1003

Among the 1221 users with accounts verified, there are 218 of them are using default profile and 1003 of them are using custom profiled.

default_profile	count
true	55598
false	43181

Among the 98779 unverified users, there are 55598 users with default profile and 43181 users with custom profile. This suggests a verified user is more likely to use custom profile.
Query 10:

```
# Users count where the number of friends_count is greater than followers_count
data_df.where('user.friends_count > user.followers_count').count()
```

```
61730
```

```
# Users count where the number of friends_count is less than followers_count
data_df.where('user.friends_count < user.followers_count').count()
```

```
37952
```

```
# Users count where the number of friends_count is the same as followers_count
data_df.where('user.friends_count = user.followers_count').count()
```

```
318
```

There are 61730 uses whose friends count is larger than the follower count. There are 37952 users whose friends count is less than followers count. There are 318 users with the same value for both friends and users counts.

Increment 1 Report - Stephanie

REPLACE_ME

Increment 1 Report - Shaun

Dataset

I pulled 100k Tweets and performed some basic queries on those Tweets to develop some insights. I used HQL and Hive to perform the analysis, so limited options were available in terms of insights that could be drawn.

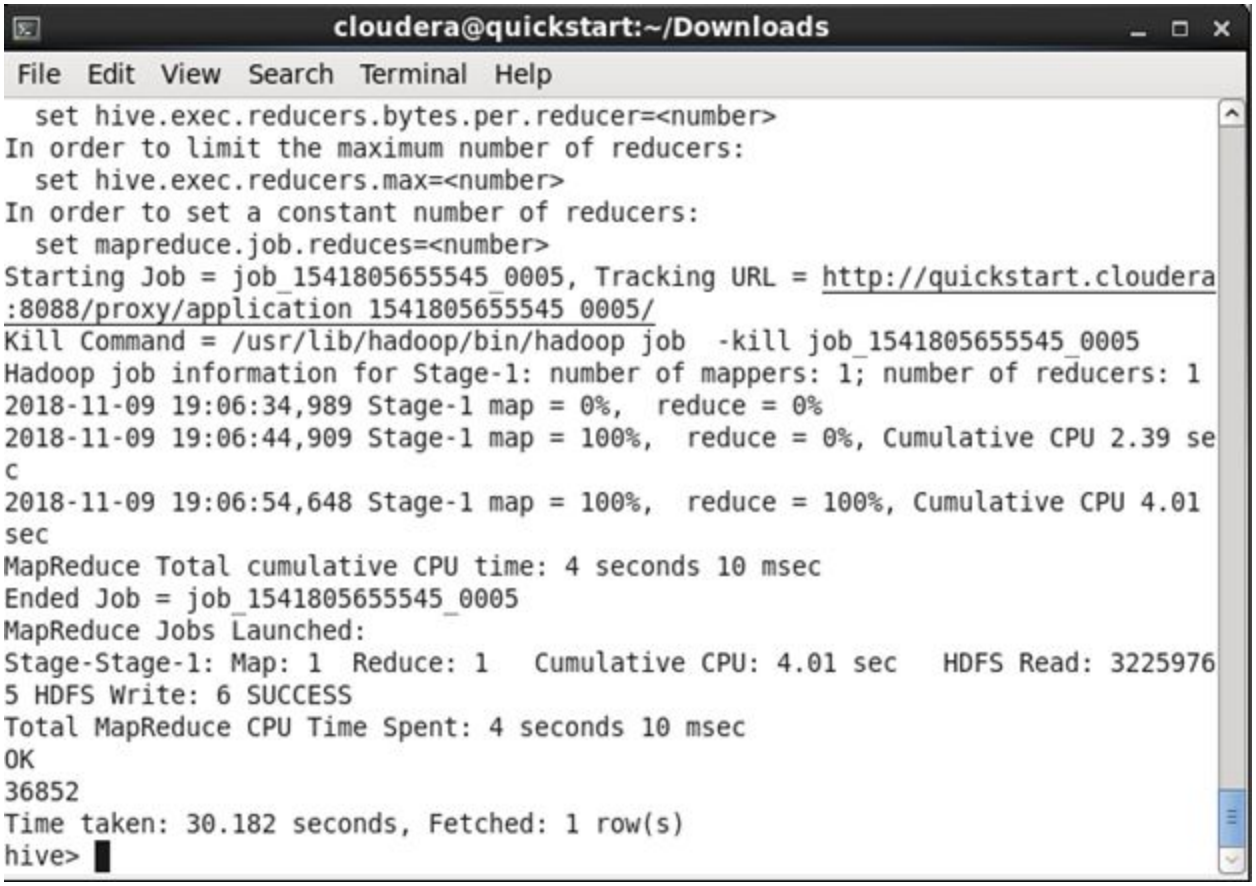
Detail Design of Features

To collect my data from Twitter I used R and RStudio in conjunction with twitteR to pull the tweets in and then exported to csv and put it into a Hive table.

Query 1:

Find total number of favorites for all the Tweets.

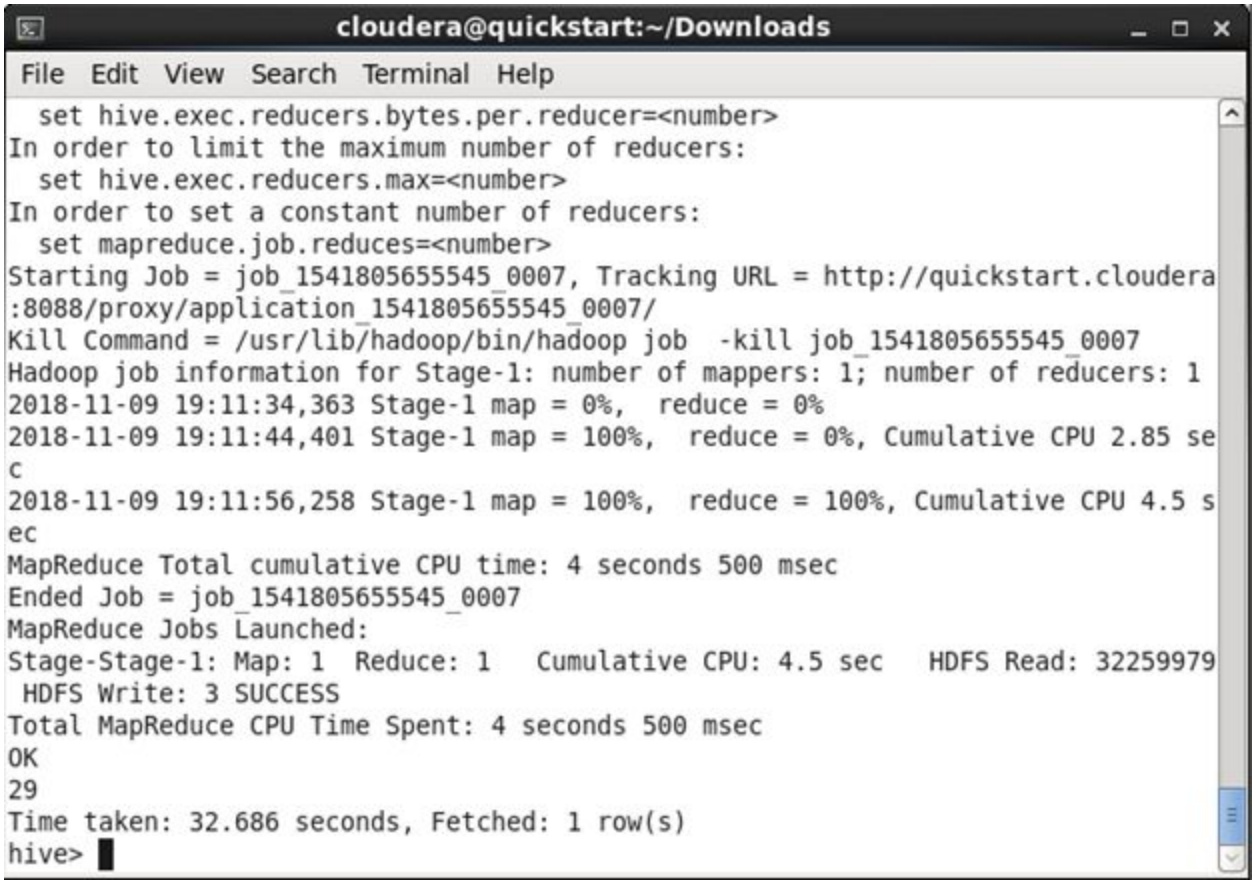
Select sum(favoritecount) from tweets;



Query 2:

Find how many tweets contain the word Trump.

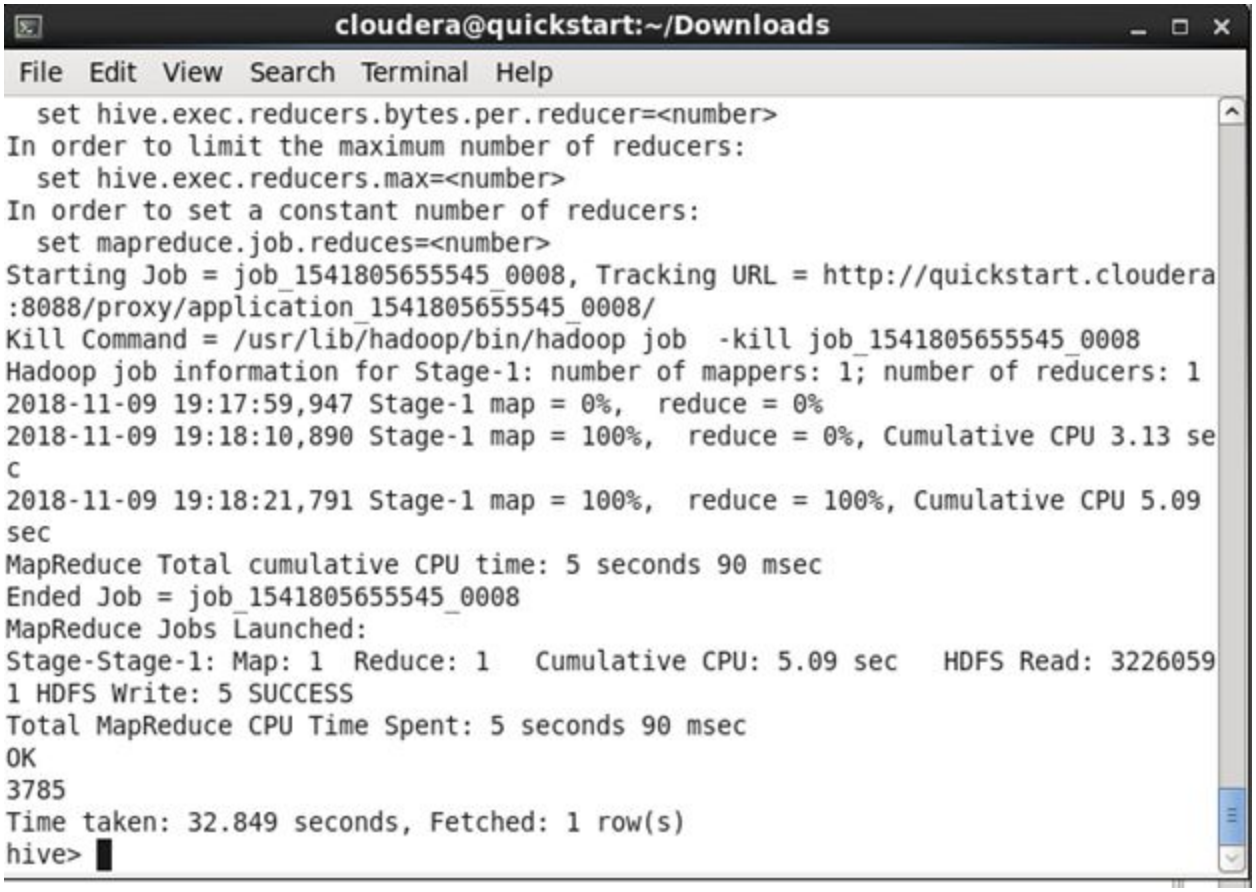
Select count(text) from tweets where text like “%Trump%” or text like “%trump%”;



Query 3

Find how many tweets are from the Northern hemisphere.

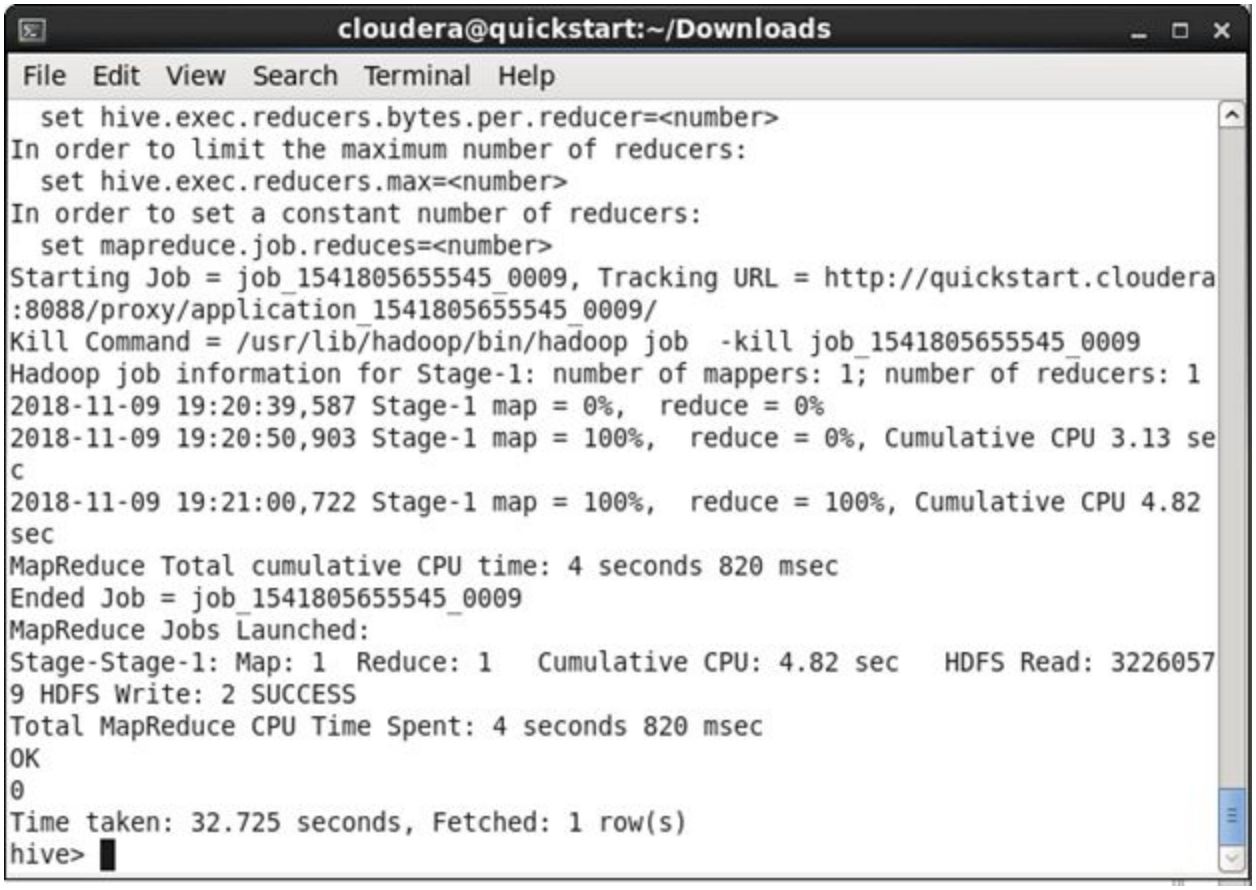
Select count(text) from tweets wh



Query 4

See how many tweets are from the southern hemisphere.

Select count(text) from tweets where latitude < 0;

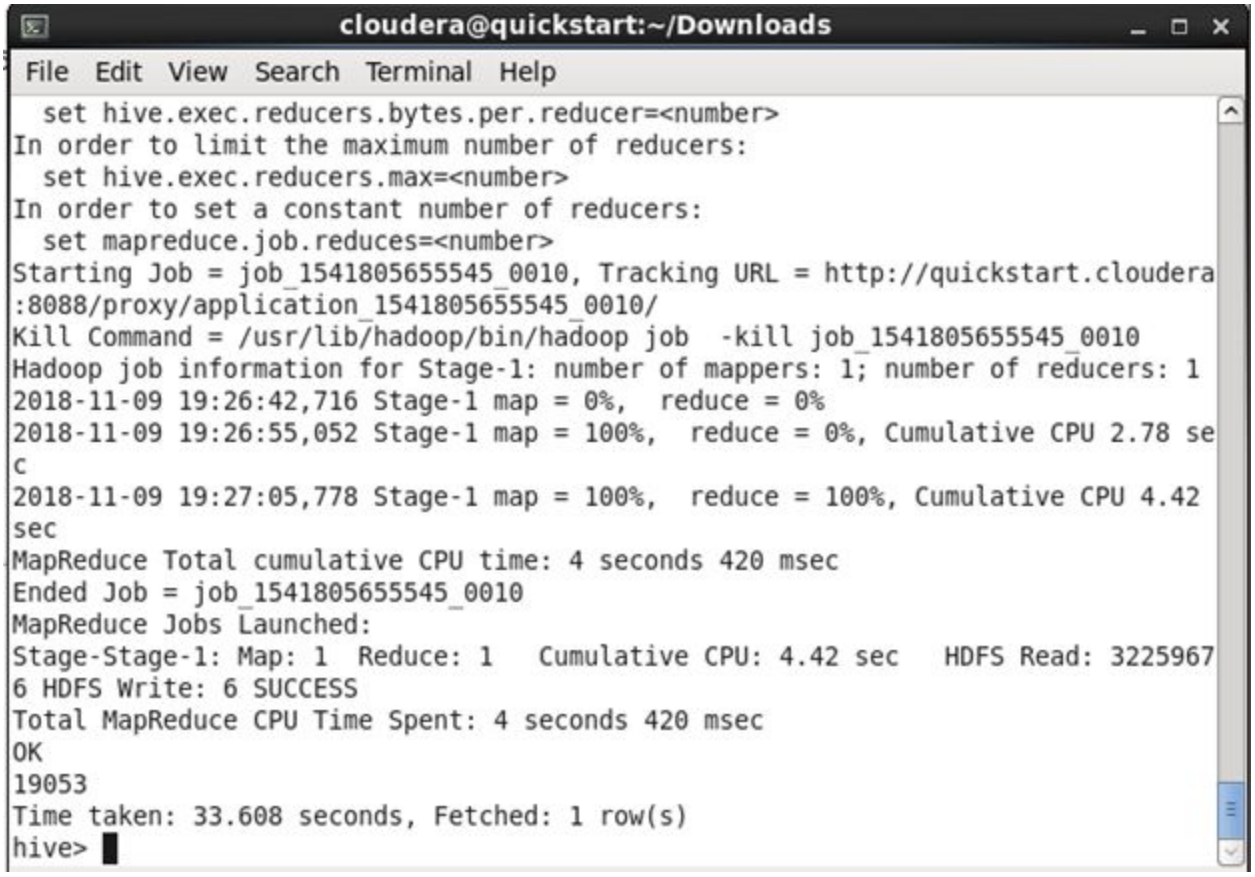


I should have gotten a total of 100000 between this query and the last. The problem turns out that a vast majority of the tweets don't share their location.

Query 5

Find out how many Tweets contain the word the.

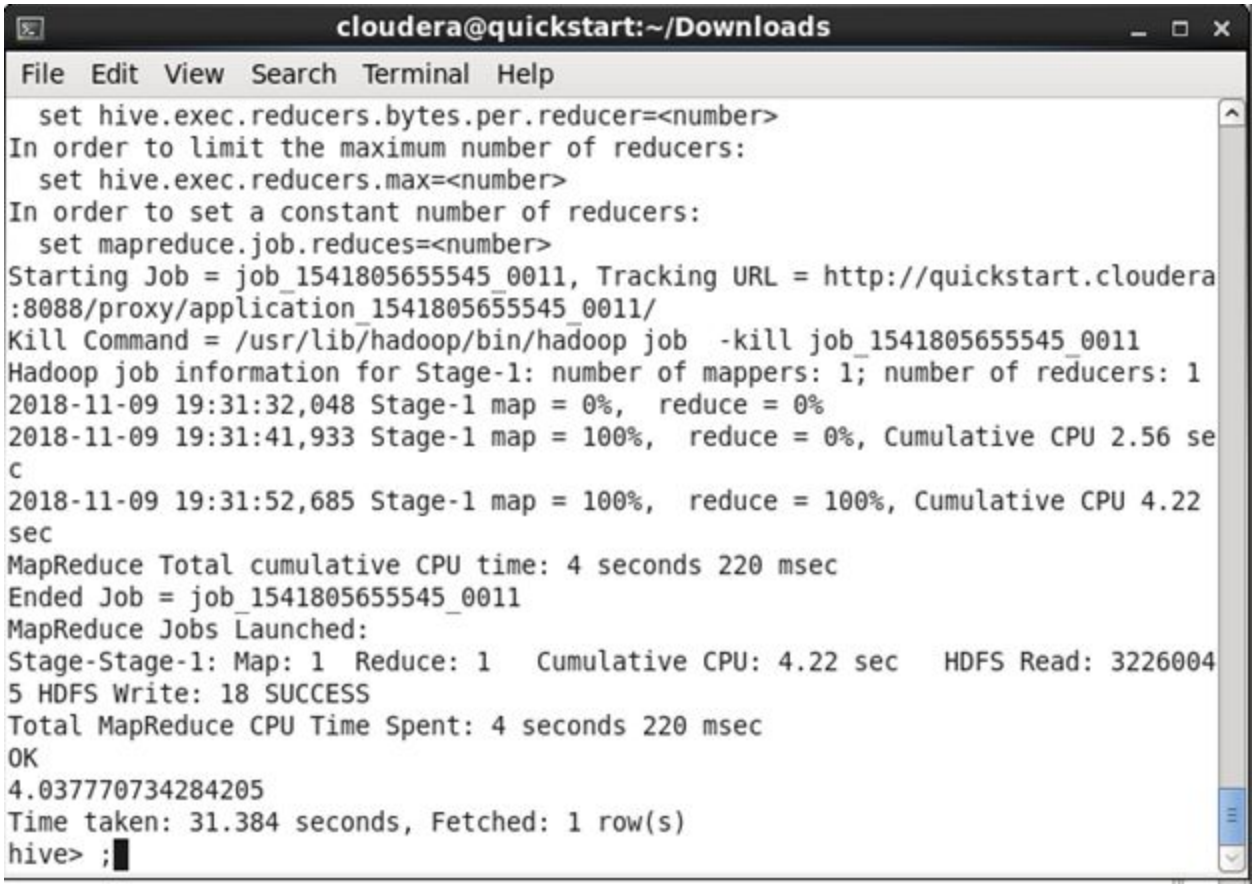
Select count(text) from tweets where text like "%the%";



Query 6

Find out the most common latitude and longitude for the tweets;

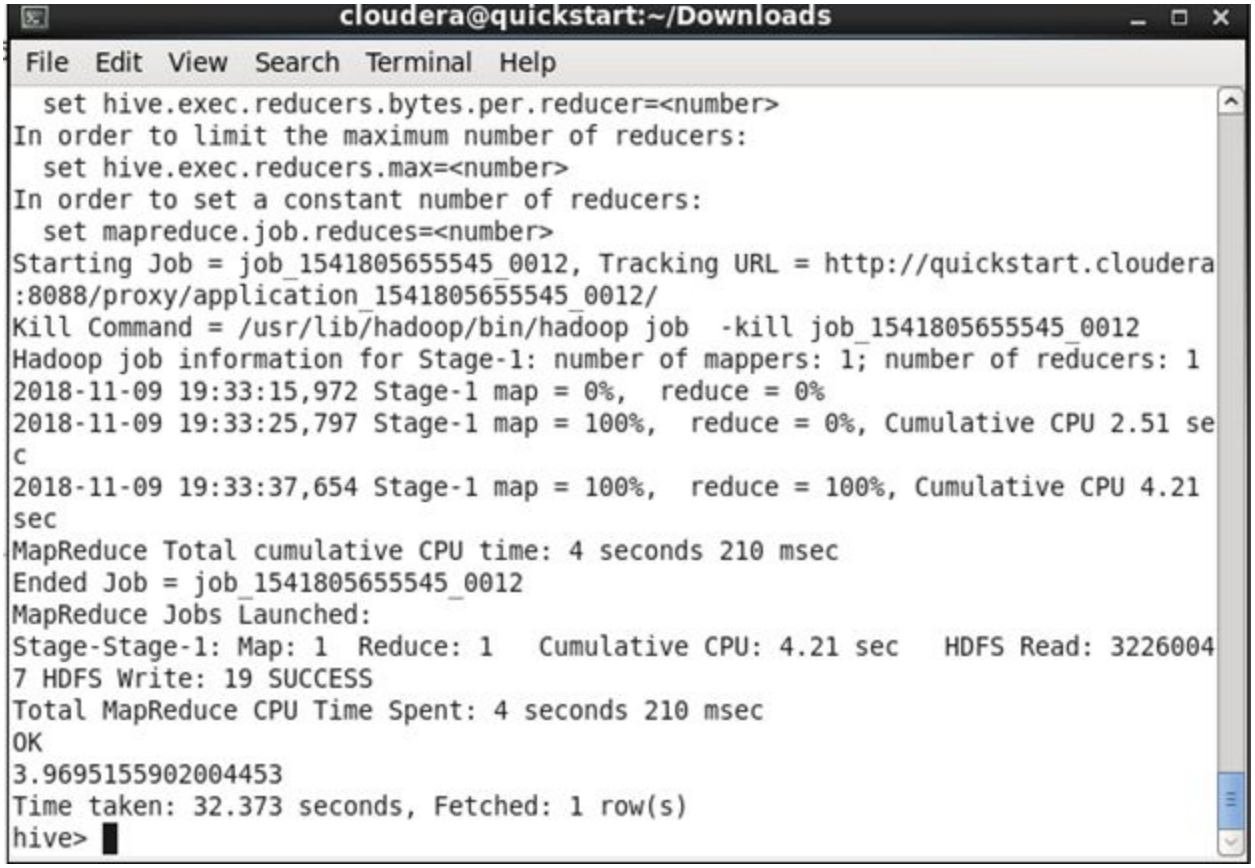
Select avg(latitude) from tweets;



Query 7

Find the most common longitude in the Tweets.

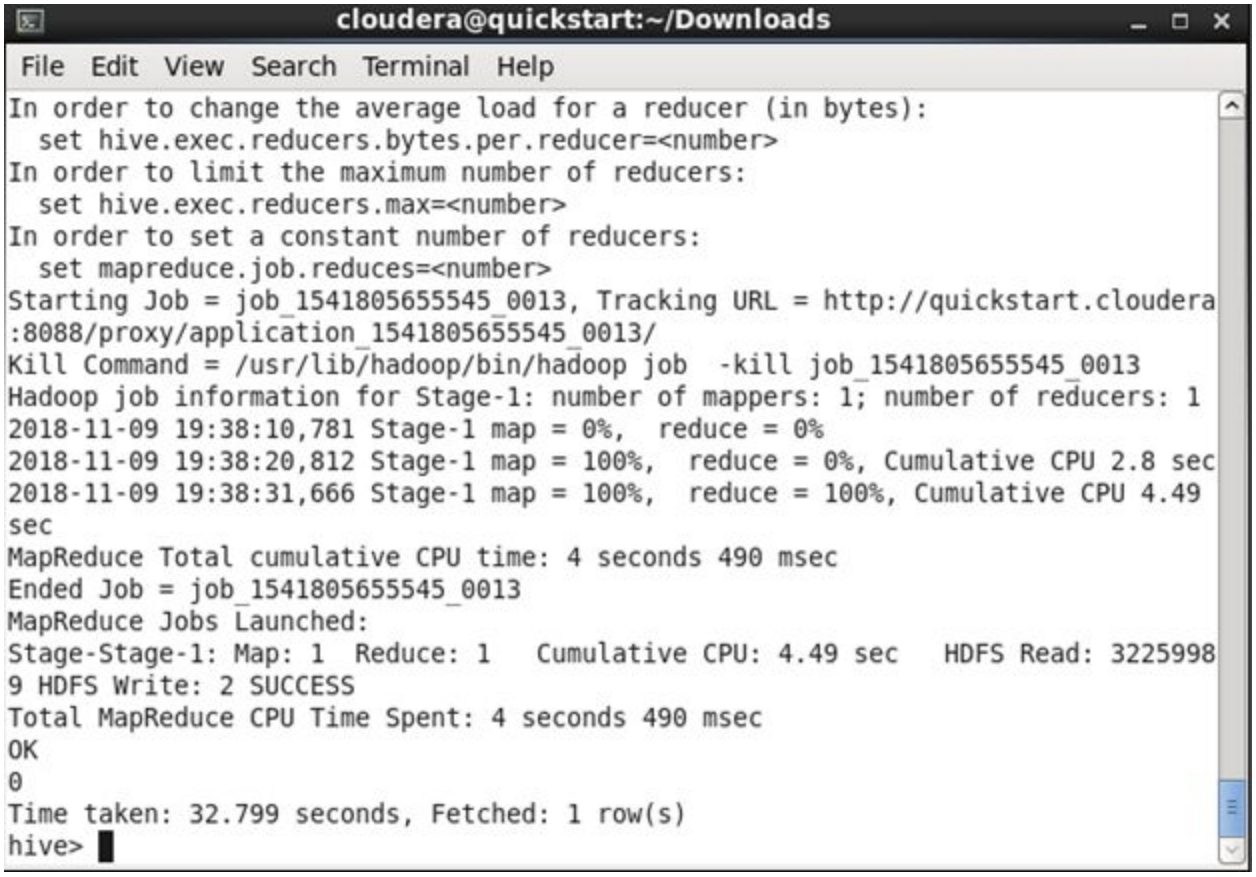
Select avg(longitude) from tweets;



Query 8

Find out what how many Tweets contain user generated content.

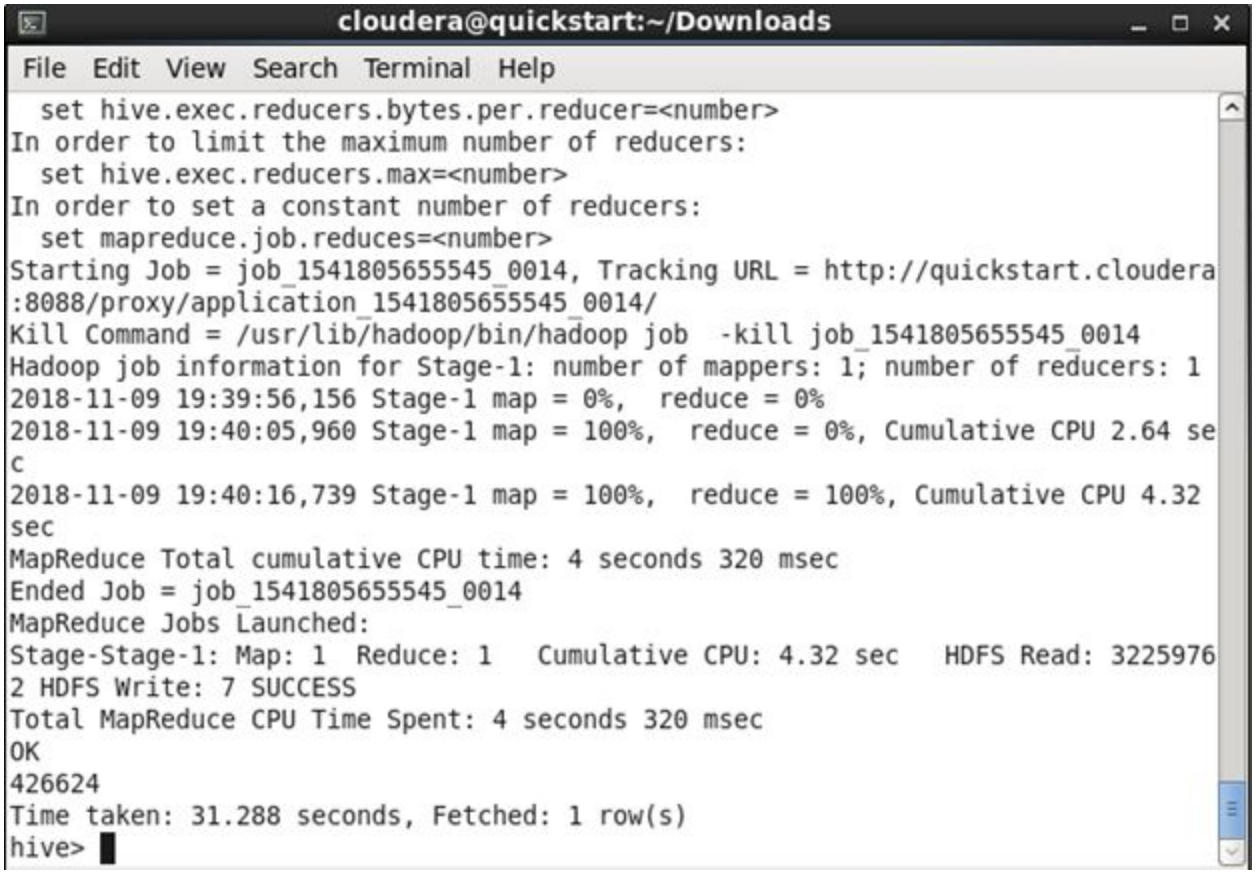
Select count(text) from tweets where text like “%User%” and text like “%Genereated%”;



Query 9

Find out how many total retweets are in the dataset.

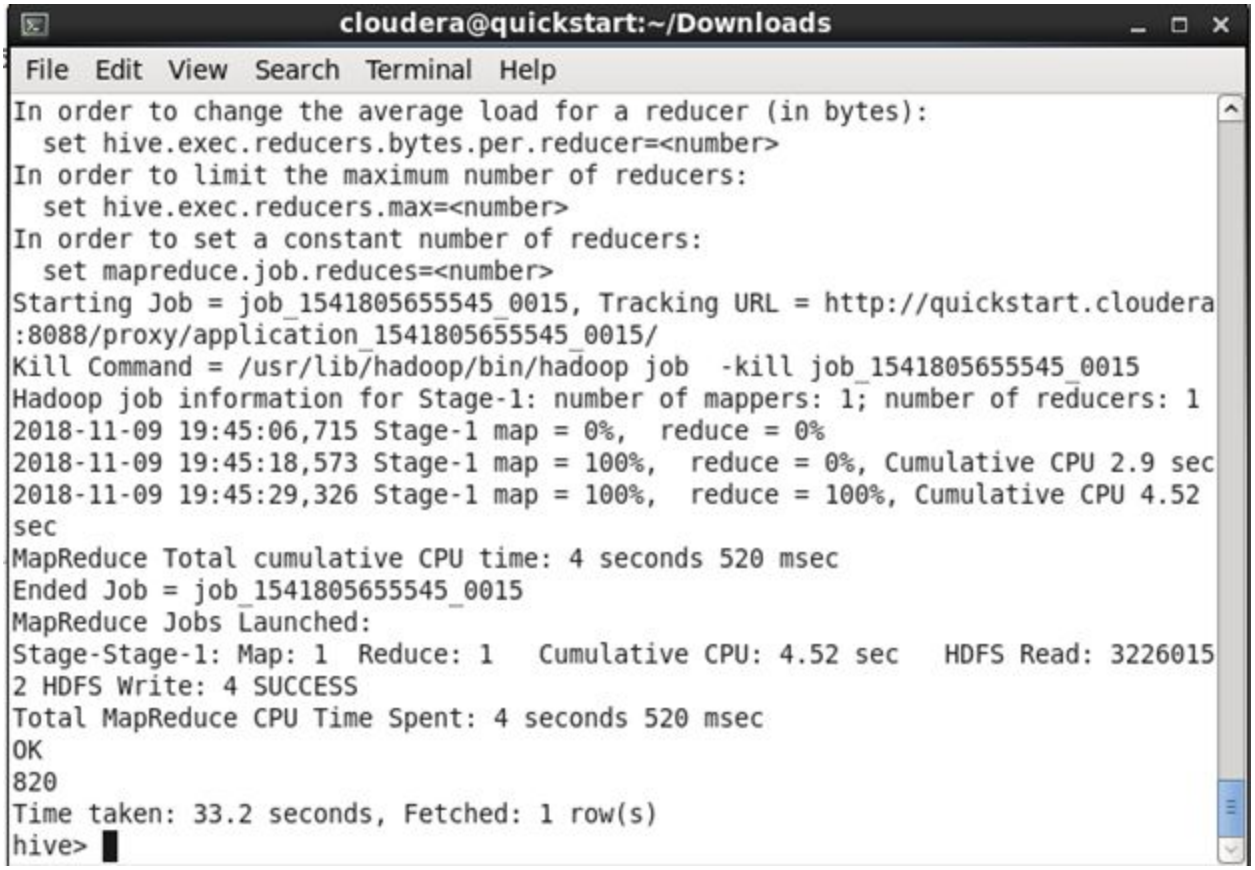
Select sum(retweetcount) from tweets;



Query 10

Find out how long the longest tweet is.

Select max(length(text)) from tweets;


cloudera@quickstart:~/Downloads
File Edit View Search Terminal Help
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_1541805655545_0015, Tracking URL = http://quickstart.cloudera
:8088/proxy/application_1541805655545_0015/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1541805655545_0015
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-09 19:45:06,715 Stage-1 map = 0%, reduce = 0%
2018-11-09 19:45:18,573 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.9 sec
2018-11-09 19:45:29,326 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 4.52
sec
MapReduce Total cumulative CPU time: 4 seconds 520 msec
Ended Job = job_1541805655545_0015
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 4.52 sec HDFS Read: 3226015
2 HDFS Write: 4 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 520 msec
OK
820
Time taken: 33.2 seconds, Fetched: 1 row(s)
hive>

Increment 1 Report - Harshil Patel

Dataset

Each member of the team is asked to collect 100K English tweets with various tracks. The analysis for each member is done on their own set of tweets (100K). Due to the size of the dataset and our previous conversation, I will not upload this dataset to GitHub.

I have downloaded 100k tweets related to [trump,senate,house,voting]

Detail Design of Features

For collecting data from Twitter, I am using tweepy package in Python. I wrote a python script to collect 100K English tweets with various tracks[trump,senate,house,voting]. With data I collected, I then used Spark and SparkSQL to perform 10 interesting queries on it.

Here are my queries:

Query 1:
Get top 10 user which have most favourites count

Query 2:
Find the tracks in the tweet and shows its distribution

Query 3:
Show tweets generated from united state and other than united states

Query 4:
Get the count of the user which have account verified

Query 5:
Get top 10 tweet that are retweeted the most

Query 6:
Get top 10 user which have most followers count

Query 7:
Get top 10 user which have most friends count

Query 8:
Get percentage of tweet based on negative, positive, neutral

Query 9:
Find gender of the user based on their name

Query 10:
Get the count of tweets

Analysis

Query 1:

I want to get the list of top 10 user which have the most favourites count.

Query 2:

I want to see how i received the tweet based upon my search criteria and want to show the distribution of the tweet in different title

Query 3:

I want to see how many people in united state tweet about trump,election,senate in us and rest of the world and shows it's comparison.

Query 4:

I want see how many people have the account verified(celebrities) on twitter.

Query 5:

I want to get the list of top 10 user tweet which are retweeted the most.

Query 6:

I want to get the list of top 10 user which have the most followers count.

Query 7:

I want to get the list of top 10 user which have the most friends count.

Query 8:

I want to use sentential analysis on the tweet to determine the polarity of each tweet then count how many tweets are positive, negative and neutral.

Query 9:

I want to use analysis on the tweet to determine the gender of the user who wrote the tweet

Query 10:

I want to get the count of the tweets

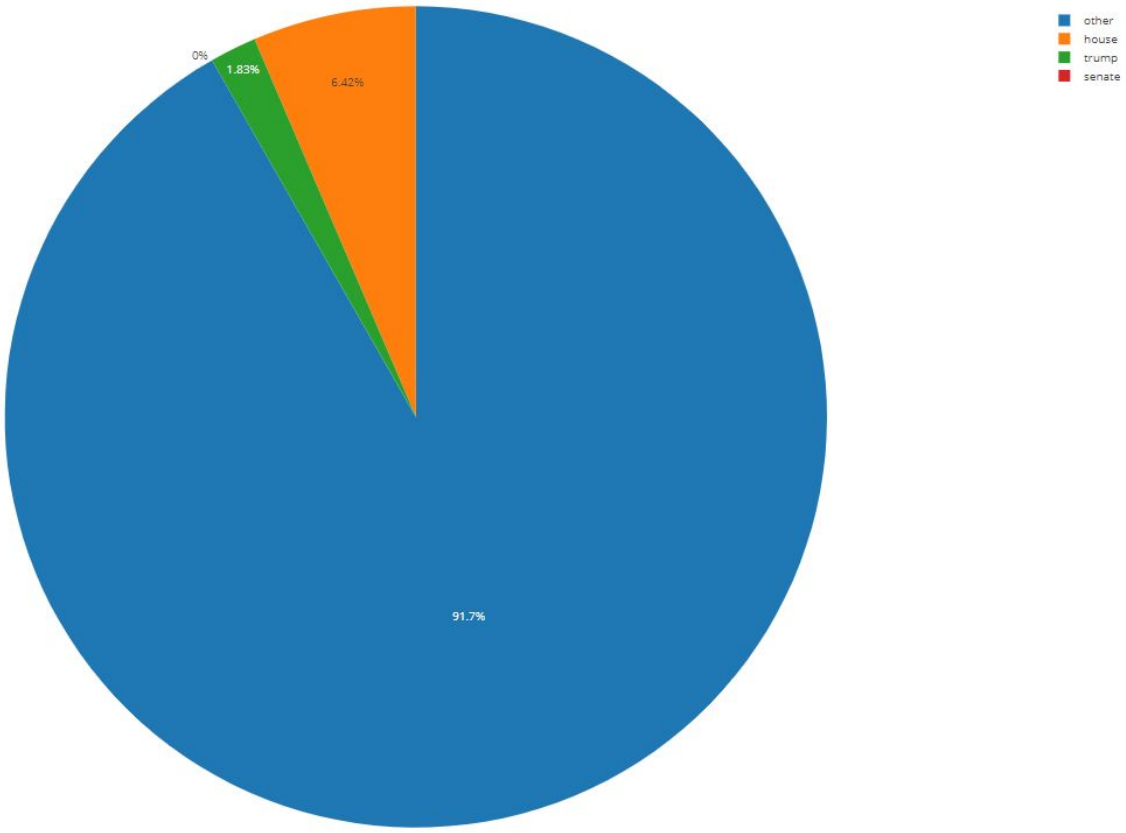
Preliminary Results

Query 1:

```
C:\Users\harsh\AppData\Local\Programs\Python\Python36-32\pyth
----------favourites_count-----
meh
1073419
Brian D.
965279
EnigMAA
851456
Madana Bhat-Khandige
823206
Monica Cates
821897
DerekPlatt
815322
rebecca lauren
812977
Brennen Burleson🗺
800649
Jeanette Baratta
799905
Grand Moff Snarkin, Surefire Intelligence CFO
772169
```

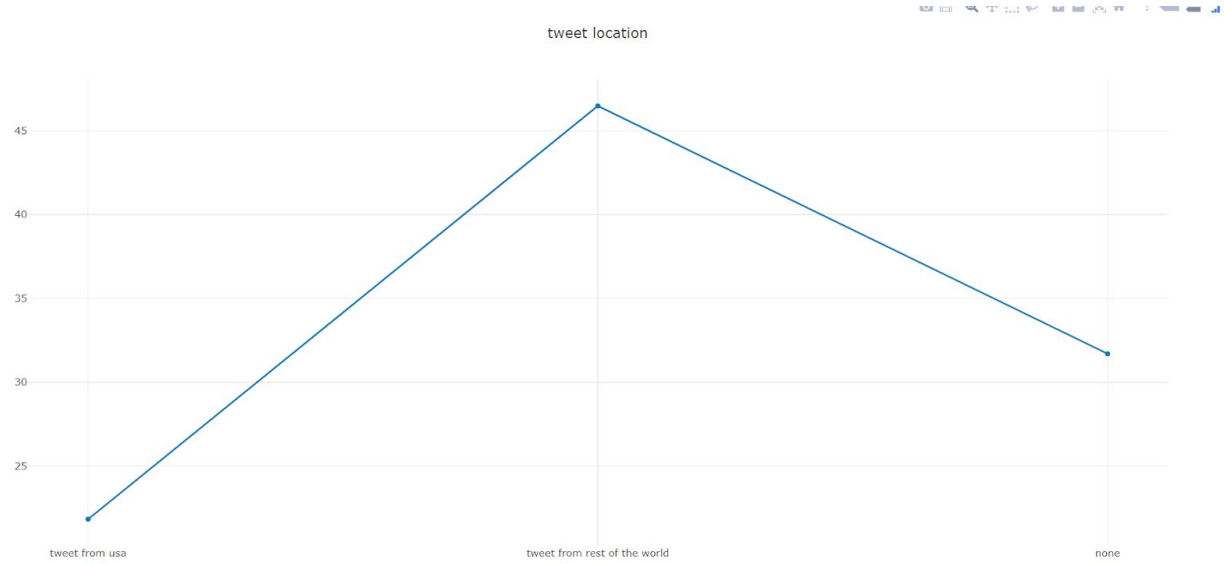
Among 100k tweet this are user names of top 10 user which have the most favourites count.the output show the username and the cout.

Query 2:



Among 100k tweet actual text containing keyword like trump,senate,house are where few.most of the tweet does not contain that keywords in there tweets.

Query 3:



Among 100k tweet 22% of tweet where originated from united states and 46% of tweets were generated from rest of the word and remaining were having location unknown.

Query 4:

```
---------account verified-----------
1319
```

Among 100k tweet there were 1319 user which have the account verified.

Query 5:

```
-------tweet that are retweeted the most------------
Rylie Geraci
633089
c h i e f
178413
BAYU ARISANDY
125011
monika bielskyte
125010
Rihanna
125009
Nasir Shakur
125007
shay
125007
RichFanAcc
118884
SaltSaltSalt
110527
rory miller
107464
```

This show the list of top 10 users tweets that were retweeted.the output show user name and number of the time the tweet was retweeted.

Query 6:

```
------followers_count-------
Donald J. Trump
55525198
President Trump
24414561
The Economist
23428140
Reuters Top News
19957413
The White House
17502570
The Washington Post
13010477
The Washington Post
13010416
China Xinhua News
11550188
Jimmy Kimmel
11427575
HuffPost
11389165
-----------------------
```

Among 100k tweet this are user names of top 10 user which have the most followers count.the output show the username and the cout.

Query 7:

```
-------friends_count--------
Ed Krassenstein
641614
Travel
572779
Jeffrey Levin
428351
Jeffrey Levin
428351
Music Lovers Fans♫
306347
Ⓜ MonsterFunder
305853
Tina Stull
236552
Reg Saddler
229134
Reg Saddler
229134
Dorian Sage ©
227725
-----------------------
```

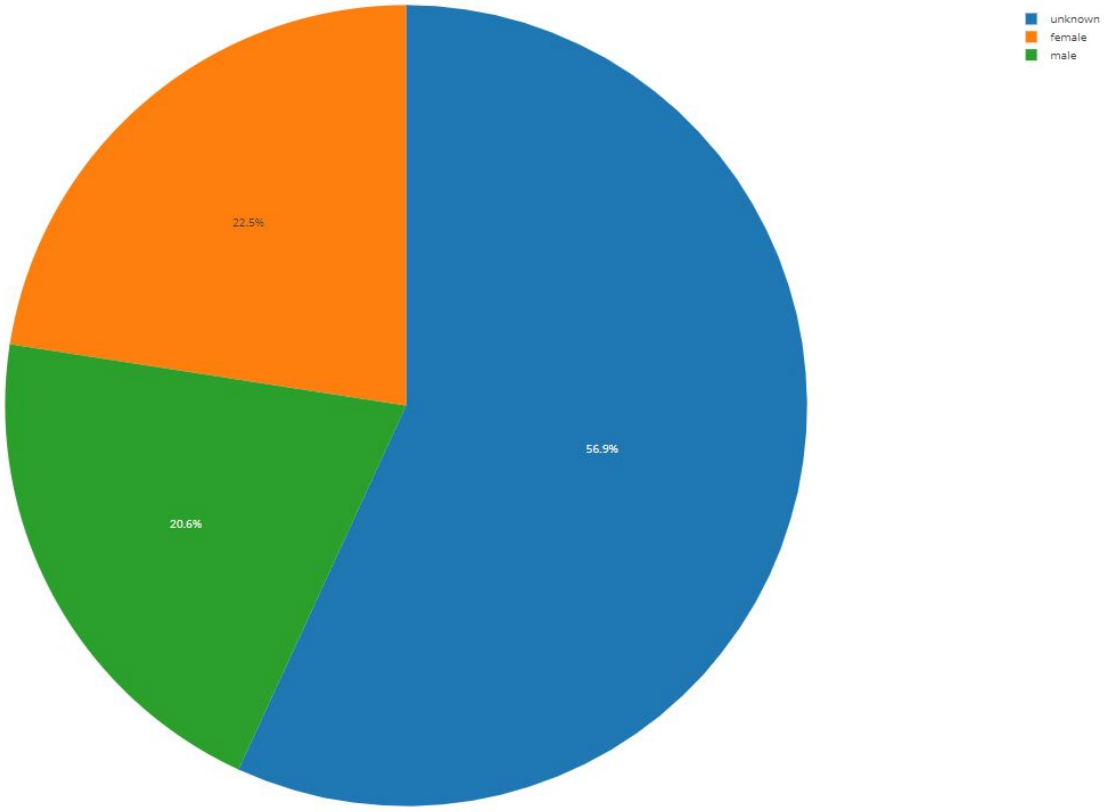
Among 100k tweet this are user names of top 10 user which have the most friends count.the output show the username and the cout.

Query 8:

```
--------negative postive tweet-------------
negative:27.449725502744972
positive:34.229657703422966
neutral:38.31961680383196
```

Among the 100K tweets, 34.23% of the tweets are positive, 38.32% of the tweets are neutral, and 27.44% of the tweets are negative.

Query 9:



Among the 100K tweets, 20.6% of the tweets are of male, 22.5% of the tweets are of female, and 56.9% of the tweets where unknown due the poor accuracy of the library.

Query 10:

```
------count-----------
100000
```

As i have downloaded 100k tweet.it show 100k count of the tweet.

Project Management - All Team Members

Worked completed

Description:

- Twitter streaming script
- Collect 100K for each member of the team
- Perform 10 interesting queries with the data we collect

Responsibility (Task, Person):

- Yong
    - Wrote twitter streaming script
    - Collected 100K tweets
    - Performed 10 interesting queries
- Stephanie
    - REPLACE_ME
- Shaun
    - Wrote R Script to pull Tweets
    - Collected 100k Tweets
    - Performed 10 queries
- Harshil
    - Wrote twitter streaming script
    - Collected 100K tweets
    - Performed 10 queries

Contributions (members/percentages):

- Each member of the team is assigned with the extra same task. The contribution for each member of the team will be 25% if they finished their assign tasks.

Work to be completed

Description:

Responsibility (Task, Person):

Issues/Concerns:

References/Bibliography