Twitter Data Analytics using Spark

Dara Venkata Sai Sandeep(16222219), Podili Venkata Krishna(16225398)

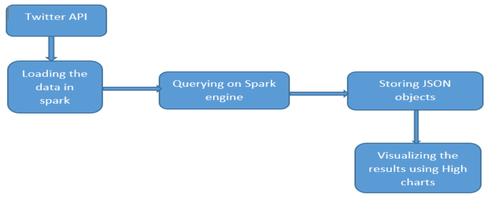
Introduction

- Big data analytics is the use of analytical techniques used against large diverse data to uncover hidden patterns, unknown correlations, market trends, customer preferences and other useful business information.
- Twitter streaming data is stored, processed to observe the current prevailing trends in the tweets.
- Twitter data was filtered using various filter criteria such as – travel, travelling, tourism, vacation, instatravel etc.
- The recent trend with most tweets at the time of traveling.
- To perform analytics we have retrieved data from twitter using API on travelling.
- Web UI was designed for the user to visualize the data in the form of graphs, charts for better understanding.

Aims & Objectives

- Collect sufficient tweet data for analytics.
- Dump the data into a database for querying.
- Write sparkSQL queries to retrieve meaningful data from the collected data
- Visualize the results using graphs using interactive User Interface.

Architecture



Query Results

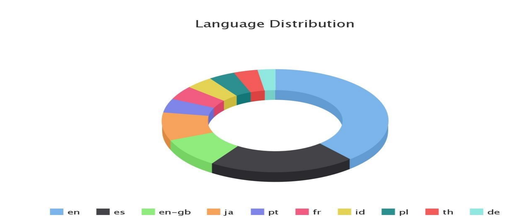
Query1: Based on Language

In this query we find the count of the language most used by travelers.

SQL Query:

select user.lang,count(*) as count from Tweets where user.lang IS not NULL group by user.lang order by count desc limit 10").toJSON.coalesce(1).saveAsTextFile("/Users/Desktop/Q1.csv")

Graph:



Query2: Based on month

In this query we find in which month most people prefer to travel.

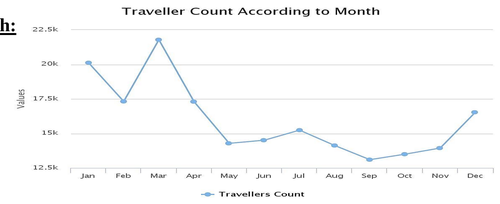
SQL Query:

val test = select substring(user.created_at,5,3) as date from Tweets where user.created_at is not null ")

test.registerTempTable("Months")

val query2=sqlContext.sql("select date,count(*) as cnt from Months group by date order by cnt desc").toJSON.coalesce(1).saveAsTextFile("/Users/Desktop/Q2.csv")

Graph:



- **Query 3: To display top places where people want to visit.**

In this query we find the frequent places and then visualized according to the people visited them.

SQL Query:

Val test = sqlContext.sql("SELECT text from Tweets where text like '%Beach%' or text like '%Mountain%' or text like '%valley%' or text like '%Dessert%' or text like '%Landscape%'")

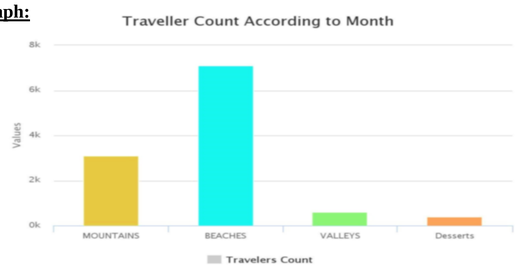
test.registerTempTable("places")

val test1 = sqlContext.sql(
"SELECT CASE WHEN text like '%Beach%' THEN 'BEACHES'" +
"WHEN text like '%Mountain%' THEN 'MOUNTAINS'" +
"WHEN text like '%valley%' THEN 'VALLEYS'" +
"WHEN text like '%Dessert%' THEN 'Desserts'" +
"WHEN text LIKE '%Landscape%' THEN 'LANDSCAPE'" +
"END AS FAMOUSPLACES from places where text is not null ")

test1.registerTempTable("test2")

val test3 = sqlContext.sql("select FAMOUSPLACES, Count(*) as Count from test2 where FAMOUSPLACES is not null group by FAMOUSPLACES order by Count DESC").toJSON.coalesce(1).saveAsTextFile("/Users/Desktop/Q3.csv")

Graph:

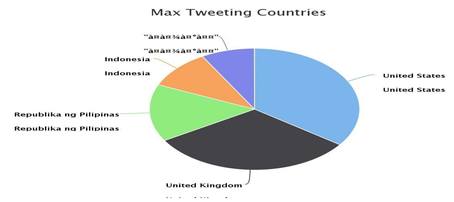


Query 4: Select the top countries who tweet about traveling

SQL Query:

Val test = sqlContext.sql(SELECT place.country,COUNT(*) AS country_count from Tweets WHERE place.country is not null GROUP by place.country order by country_count desc limit 5 ").toJSON.coalesce(1).saveAsTextFile("/Users/Desktop/Q4.csv")

Graph:

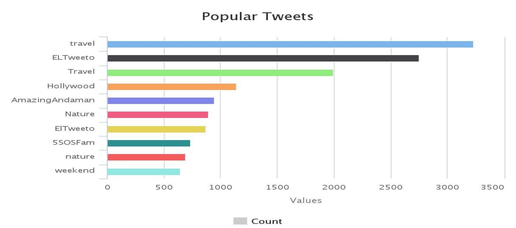


Query 5: Select the top famous hashtags

SQL Query:

Val test = sqlContext.sql ("SELECT entities.hashtags[0].text, count(entities.hashtags[0].text) as famous_tags FROM Tweets group by entities.hashtags[0].text order by famous_tags desc limit 10").toJSON.coalesce(1).saveAsTextFile("/Users/Desktop/Q5.csv")

Graph:



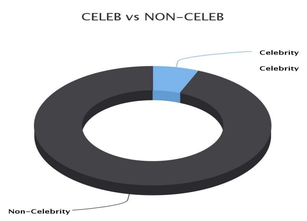
Query 6: Query to find celebrity accounts.

Celebrities vs Non-celebrities

SQL Query:

val test1 = sqlContext.sql ("select user.verified, count(distinct user.id)as count from Tweets where user.verified is not null group by user.verified") .toJSON.coalesce(1).saveAsTextFile("/Users/Desktop/Q7.csv")

Graph:



Query 7:

Time zone, Tweet count and retweet count from the data

SQL Query:

val test1 = sqlContext.sql ("select user.time_zone as time_zone, count(*) as Tweet_count from Tweets where user.time_zone is not null group by user.time_zone order by Tweet_countdesc")

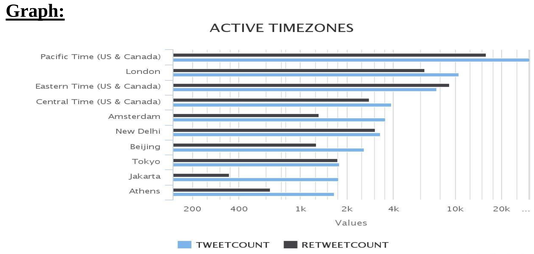
test1.registerTempTable("test1")

val test2 = sqlContext.sql("select retweeted_status.user.time_zone as time_zone, count(*) as Retweet_count from tweets where retweeted_status.user.time_zone is not null group by retweeted_status.user.time_zone order by Retweet_countdesc")

test2.registerTempTable("test2")

val Query6 = sqlContext.sql("select test1.time_zone, test1.Tweet_count, test2.Retweet_count from x1 inner join test2 on test1.time_zone = test2.time_zone order by test1.Tweet_count desc").toJSON.coalesce(1).saveAsTextFile("/Users/Desktop/Q6.csv")

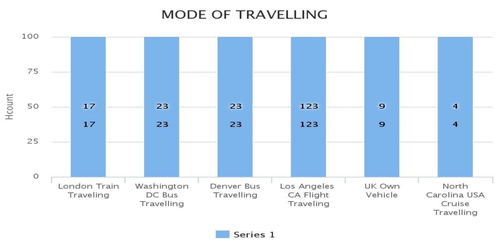
Graph:



Query 8: Based on the Favorite mode of Transportation

In this query we have taken top five modes of travelling according to the top most cities which had used that mode of travelling.

Graph:



Conclusion

We have found the several analytical results on the tweets we collected based on the travelling like most frequent places visited by the people, mode of travelling, the language they tweeted, most popular tweet hashtags, traveling according to month, celebrity vs non-celebrity count etc. In future we would like to put altogether into a kind of website like twitter data analytics.

UMKC
UNIVERSITY OF MISSOURI-KANSAS CITY