ANALYZING & VISUALIZING TWITTER DATA

Under the Guidance of:
Prof. Praveen Rao

Puri, Varun(vp4gb@mail.umkc.edu**)**

Topic: Traveling

Phase-2

Twitter Big Data Analytics using Spark

CS5540PB Project Report

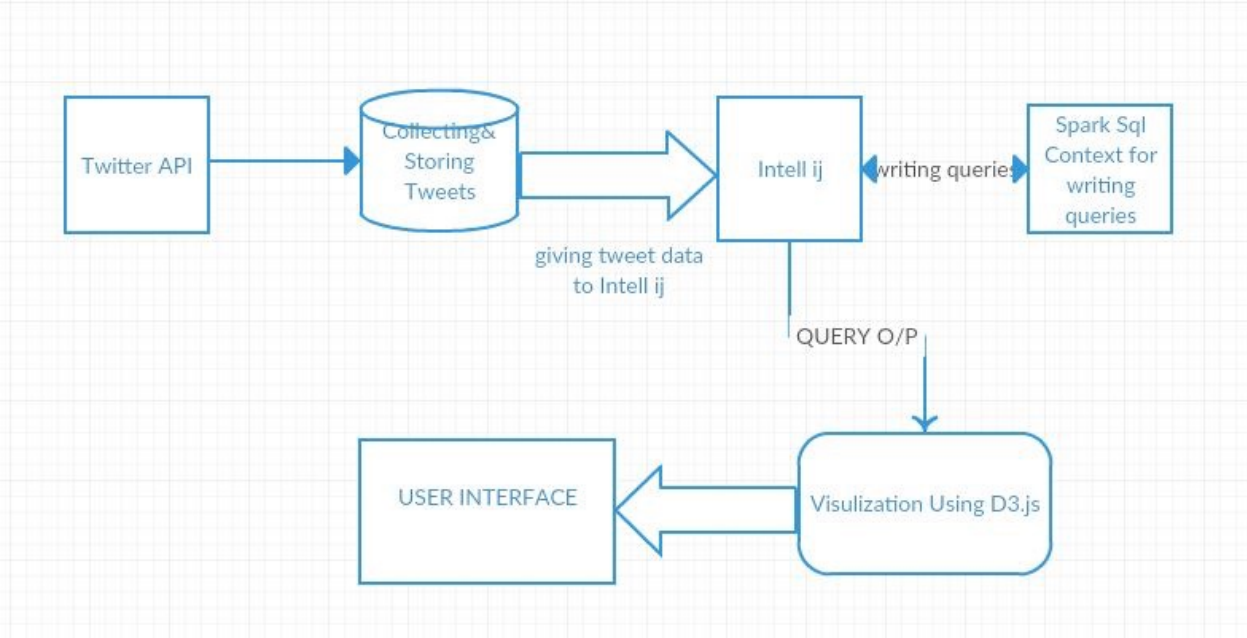
Introduction:-

Big Data is an evolving term that describes any voluminous amount of structured and unstructured data that has the potential to be mined for information. Big-data analytics is the process of examining large datasets containing a variety of datatypes i.e. biodata to uncover hidden patterns, unknown correlations, market trends, customer preferences and other useful business information. With Biodata analytics, data scientists and other can analyze huge volumes of data that conventional analysts and business intelligence solutions can't touch. A latest and trending feather in the Hadoop's Cap is Apache SPARK. Apache spark is an open source engine developed specifically for handling large scale data processing and analytics. Spark enable user to run large scale data analytics application across various clustered computers. In our Project we have use the IntelliJ Scala plugin to analysis our data and get the results.

Project Description:-

"We travel not to escape life, but for life not to escape us". Traveling is about exploring world to yourself. Now-a-days, one of the best part of social Networking world is that you can live the Journeys or Moment of your friends, your icons from your mobile or laptop screen. People share their beautiful experience, their journeys, their activities, by posting their photographs, Videos on social networking sites. This ever growing unstructured data provide seamless opportunities in every dimension for everyone for example many startup or establish traveling business to identify their clients. In our project we have tried to collected the tweets based on topics with Hashtag traveling, Mountain, Beaches, Vacation, Wanderlust, Tourism and then perform some analysis on the collected data using Spark SQL Context by performing the analytical queries on it. Finally we have some visualizations like big sheets tools to display the output.

ARCHITECTURAL DIAGARAM:-



Software Modules:

1) Data Collection:

We have collected tweets from twitter and stored them in the form of JSON files. The collection of tweets is done using python twitter streaming. The search is done based upon the following keywords: #Travel, #Beaches, #Beautiful, #Holidays, #Nature, #Tourism, #vacation, #wanderlust, #traveler-gram, #travelling, #traveldairies, #instatravel, #weekendtravel etc.

2) Data Analysis:

We used JSONLINT tool (on http://jsonlint.com/) to analyze the schema of the data collected. Based on the schema we designed our queries.

3) Data processing:

Here are the steps that we followed for data processing

- Tweets: - Created one single JSON file Tweets by combining all collected files using code. Contains all tweets collected based on the hashtags mentioned above.
- Based on my analysis created three tables UserInfo, Tweet_Info, Tweet_User using scala queries. Here is the schema of the three tables.
  - ➢ tweet_Info:
    tweet_id, favourites_count, retweet_count, created_at, description
  - ➢ tweet_user_info: id, tweet_id
  - ➢ user_info: id,name, location, followers_count, verified,
    profile_image_url
- Configured Scala on IntelliJ (installed Scala plugins on IntelliJ) and wrote queries using spark SQL context
- Query used to create tweet_info file.

➢ **val test = sqlContext.sql("select id as tweet_id, user.favourites_count,retweeted_status.retweet_count,created_a**

t,user.description from userinfo where id IS NOT null AND user.favourites_count IS NOT null AND retweeted_status.retweet_count IS NOT null AND created_at IS NOT null AND user.description IS NOT null ")

- **test.registerTempTable("tweet_info")**
- Query used to create tweet_user_info file

- **val test1 = sqlContext.sql("select user.id, id as tweet_id from userinfo where user.id IS NOT null AND id IS NOT null")**
- **test1.registerTempTable("tweet_user_info")**
- Query used to create user_info file.

- **val test2 = sqlContext.sql("select user.id, user.name, user.location, user.followers_count, user.verified, user.profile_image_url from userinfo where user.id IS NOT null AND user.name IS NOT null AND user.location IS NOT null AND user.followers_count IS NOT null AND user.verified IS NOT null and user.profile_image_url IS NOT null")**
- **test2.registerTempTable("user_info")**

4) Data Visualization:

For data visualization I have used big-sheets.I installed IBM Big Insight 3.0.

Analytical Queries & Visualization:

Query1: Based on Location where People tweet Most about Traveling

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

Explanation:

Selected location and counted tweet_id using count function for those locations and used GROUP by and ORDER By clause to show top 5 countries who tweet most about traveling.

Query:

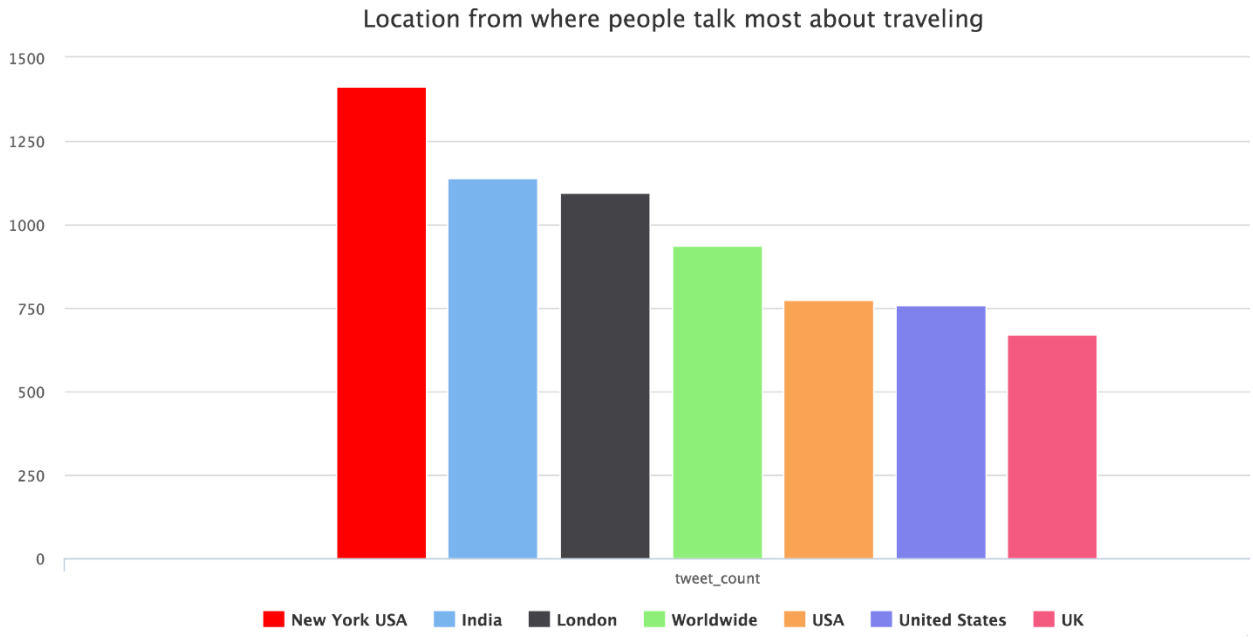
```
val NewQ1 = sqlContext.sql("select t1.location,count(distinct t2.tweet_id) AS
tweet_count from user_info t1, tweet_user_info t2  WHERE t1.id = t2.id
group by t1.location order by tweet_count desc
```

Output:


Q1.txt

Visualization:

We have converted data in output file to CVS format as High charts accepts CSV data.



Query2: Based on Celebrity and Non Celebrity Account.

To find count of celebrity and non-celeb accounts who tweet about travelling

Explanation:

Selected verified and counted distinct id count using distict and count keywords. Used group by clause to show count of ids by verified columns value. Verified coulumn value is true for celeb accounts and false otherwise.

Query:

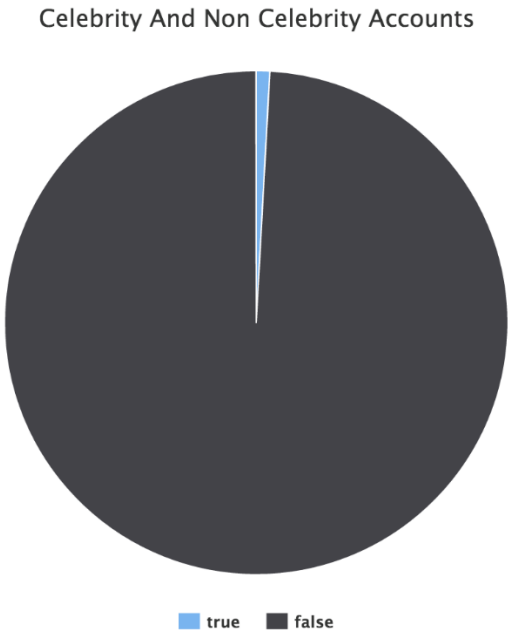
sqlContext.sql("select verified, count(distinct id) as User_count from userinfo group by verified").toJSON.coalesce(1).saveAsTextFile("/Users/shashibisht/Desktop/Q2.json")

Output:


Q2.txt

Visualization:

We have converted data in output file to CVS format as High charts accepts CSV data.



Query 3: To find 10 top celebrity who have tweeted most about travelling.

Explanation:-

I have used alias names of two tables user_info and tweet_user_info to refer to their column name in order to get the top 10 celebrities. I first selected the user ids from user_info and counted distinct tweet ids from tweet_user_info where user ids on both columns are same.

Query:

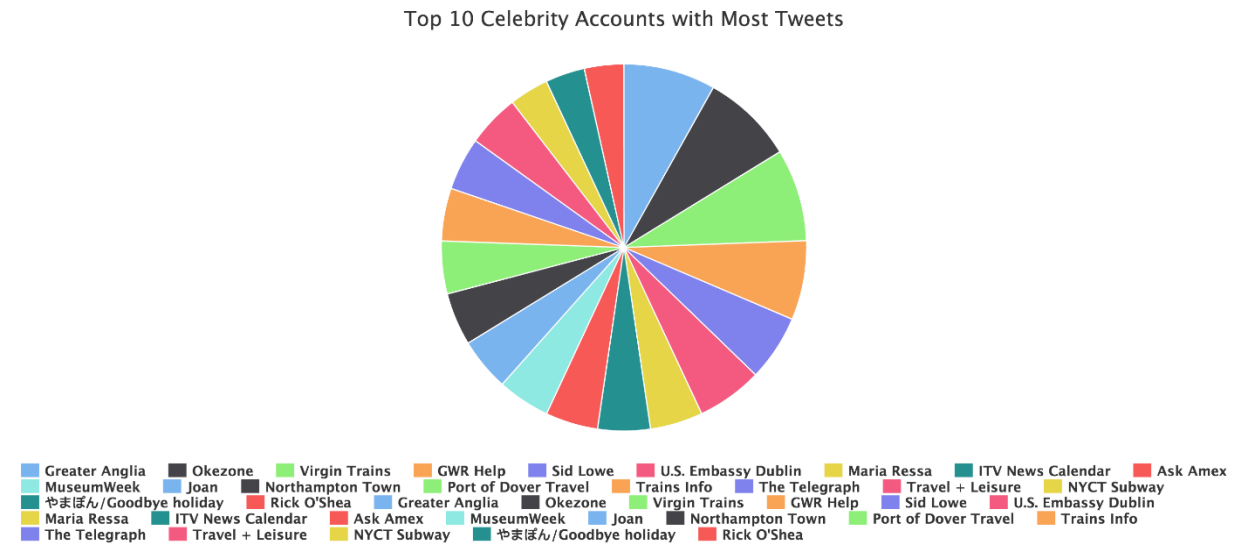
sqlContext.sql("select t1.name , t1.id, count(distinct t2.tweet_id) as COUNT from user_info t1m tweet_info t2 where t1.verified = true and t1.id = t2.id group by t1.name, t2.id order by COUNT desc limit 20").toJSON.coalesce(1).saveAsTextFile("/Users /Desktop/Q3.json")

OutPut :-


Q3 (2).txt

Visualization:

We have converted data in output file to CVS format as High charts accepts CSV data.



Query 4: Query to find 10 most popular tweets.

Explanation:-

I have used two queries to obtain the result.

In first query I joined all three tables where ids are equal and tweet_ids are equal and selected tweet content (description) and added using sum function the favourite count and retweet count. Stored the result in another table fav_tweet.

In the second query I joined tables fave_tweet and user_info where both tables have same values in name column. I used group by clause to display the result in group of count and column.

Query:

```
val test5 = df3.sqlContext.sql("select  t1.description, t3.name as name, sum(t1.favourites_count + t1.retweet_count) as my_count from tweet_info t1, tweet_user_info t2, user_info t3 where t1.tweet_id = t2.tweet_id and t2.id = t3.id and t1.description LIKE '%travel%'  group by t1.description, t3.name order by my_count desc limit 10")

test5.registerTempTable("fav_tweet")
```

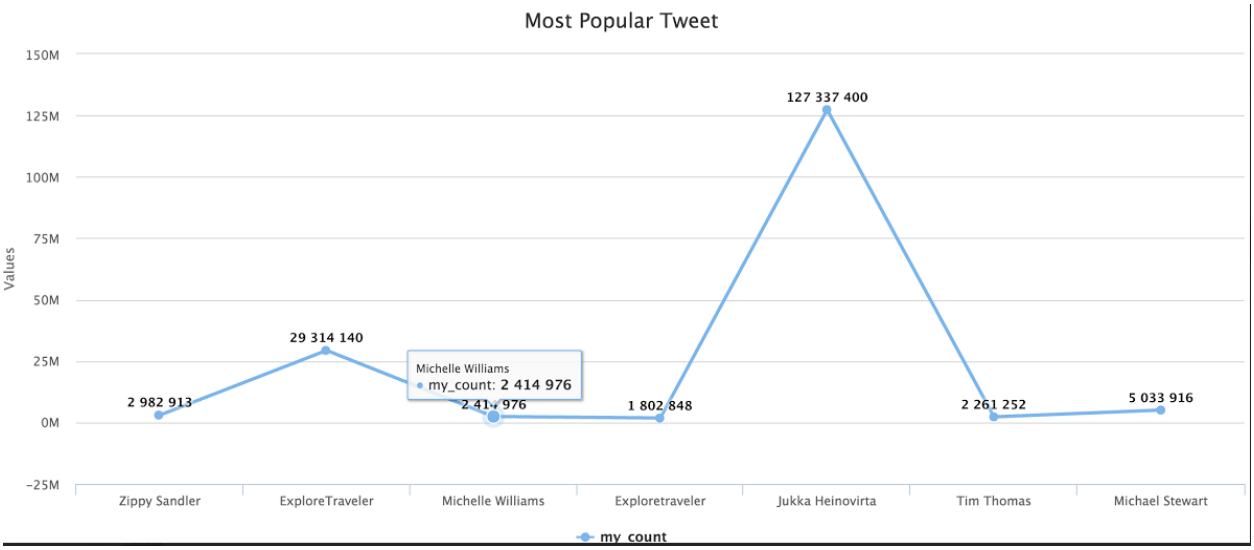
sqlContext.sql("select t1.name, t2.my_count from user_info t1, fav_tweet t2 where t1.name = t2.name group by t2.my_count, t1.name").toJSON.coalesce(1).saveAsTextFile("/Users/shashibisht/Desktop/Q4.json")

OutPut :-


Q4.txt

Visualization:

We have converted data in output file to CVS format as High charts accepts CSV data.



Query 5: Query to find the months in which people prefer to travel most

Explanation:-

I have used two queries to obtain the result.

In first query I took a substring from created_at field of table tweet_info, that substring gives us the month. I saved the output in tsbl MONTH.

In the second query I selected date column from table MONTHS and counted occurrences of date using count method and stored the output on Q5.json file.

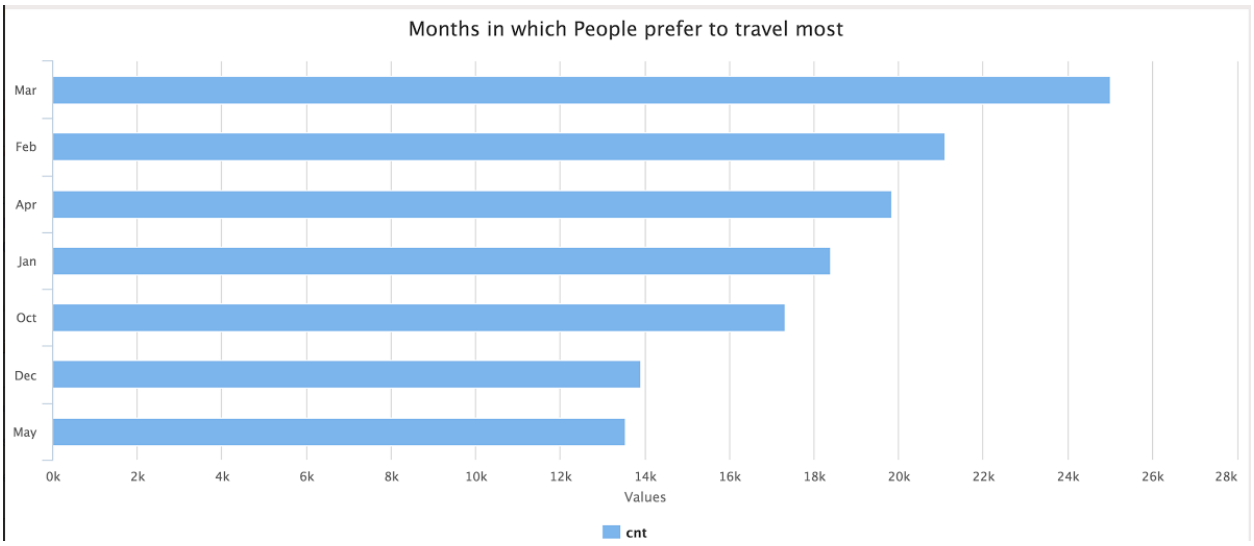
Query:

```
val test8 =  sqlContext.sql("select substring(created_at,5,3) as DATE from tweet_info")
test8.registerTempTable("MONTH")
sqlContext.sql("select DATE, count (*) as COUNT from MONTH group by DATE order by COUNT desc").toJSON.coalesce(1).saveAsTextFile("/Users/shashibisht/Desktop/Q5.json")
```

OutPut :-


Q5.txt

Visualization:


Months in which People prefer to travel most
Mar
Feb
Apr
Jan
Oct
Dec
May
0k
2k
4k
6k
8k
10k
12k
14k
16k
18k
20k
22k
24k
26k
28k
Values
cnt