CSEE5590/490: BIG DATA PROGRAMMING

ANALYSIS ON US ELECTION USING SOCIAL MEDIA

1. TEAM MEMBERS:

1. NIKHITHA, KOLLURI (16290856)

2. CHAITANYA, MALLEPUDI (16291222)

3. ESWAR, VALLURU (16292742)

4. SRI SAI NITHIN CHOWDARY, DUKKIPATI (16291615)

2. GOALS AND OBJECTIVES:

- **MOTIVATION:**

The world has been advancing significantly day by day, so as the data. Nowadays the data has become so vast such that it could not be stored in a single cluster or a machine, thus the concept of big data arisen, and its services have been implemented in various sectors. Many tools and technologies are arriving in the market very frequently to work with this big data. We thought of analysing social media using the existing techniques, visualize the data, aggregate, and model it in a user-friendly way.

- **SIGNIFICANCE:**

People are very curious about the recent trending issues across the globe, Since the 2020 US elections have been trending now the people want to know about the different views across the country on participating election candidates. So, we are collecting social media data to analyze different views and opinions, so that people can get awareness.

- **OBJECTIVES:**

Our Undertaking's fundamental thought is to do the ETL cycle utilizing Spark Cluster Handling, Cleaning the data, performing sentimental analysis, visualizing the data, and Incorporating visualization with Web UI. The wellspring of our framework is Twitter information and we would utilize Tweepy API to gather the information. We would deploy the web UI incorporating the visualizations in the Cloud environment.

- **FEATURES:**

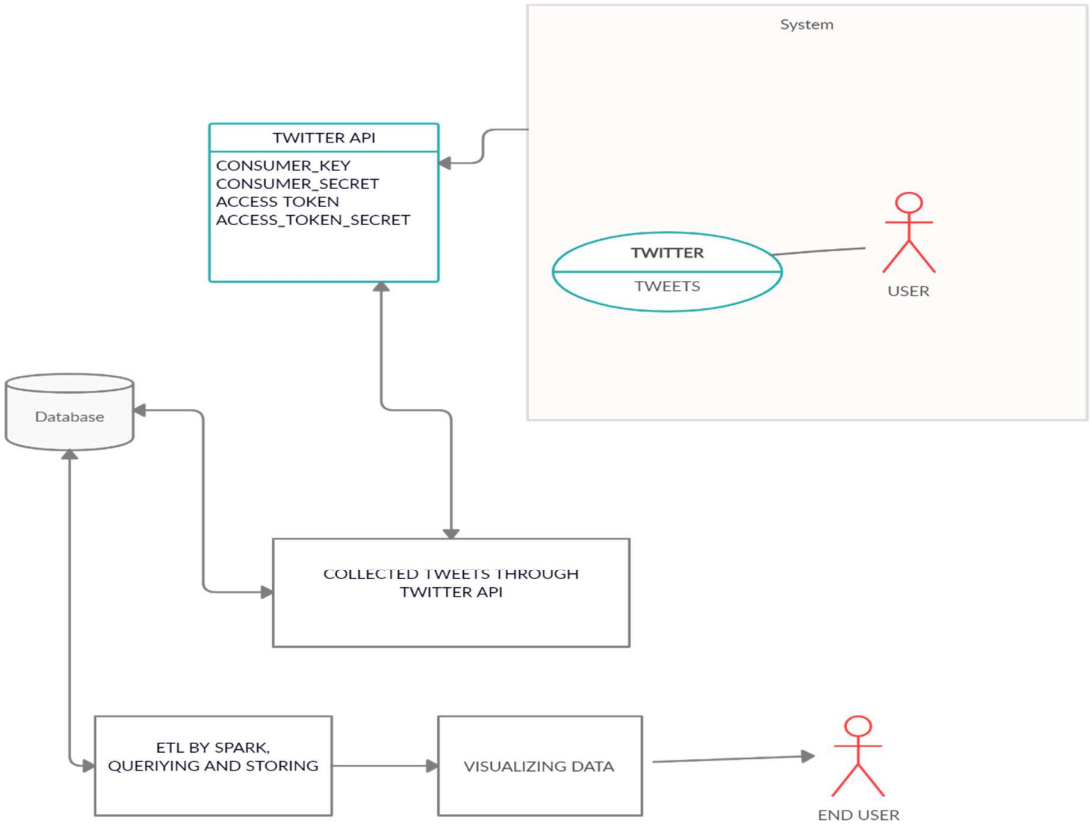
This project includes streaming of live twitter data and processing it through SPARK ETL and using TextBlob from NLTK to process the data and predict the sentiment and Creating a Web UI incorporating the visualization of data and deploying it to the Cloud environment.

3. DATA SET:

Twitter data on Donald Trump and Joe Biden (US ELECTION 2020)

This dataset has all the recent tweets about Joe Biden and Donald Trump and tweet json object has all the attributes which gives a precise and vital view about what and where things are going around the candidates.

4 DESIGN:



STEPS:

1. User will tweet to the twitter server
2. Using developer account using tweepy API we extract the tweets
3. Extracted tweets are stored
4. We use Spark ETL to process the data by querying and retrieving only the vital information from the tweets stored.
5. We visualize the information to the end user.

5. ANALYSIS

A typical tweet has the following fields

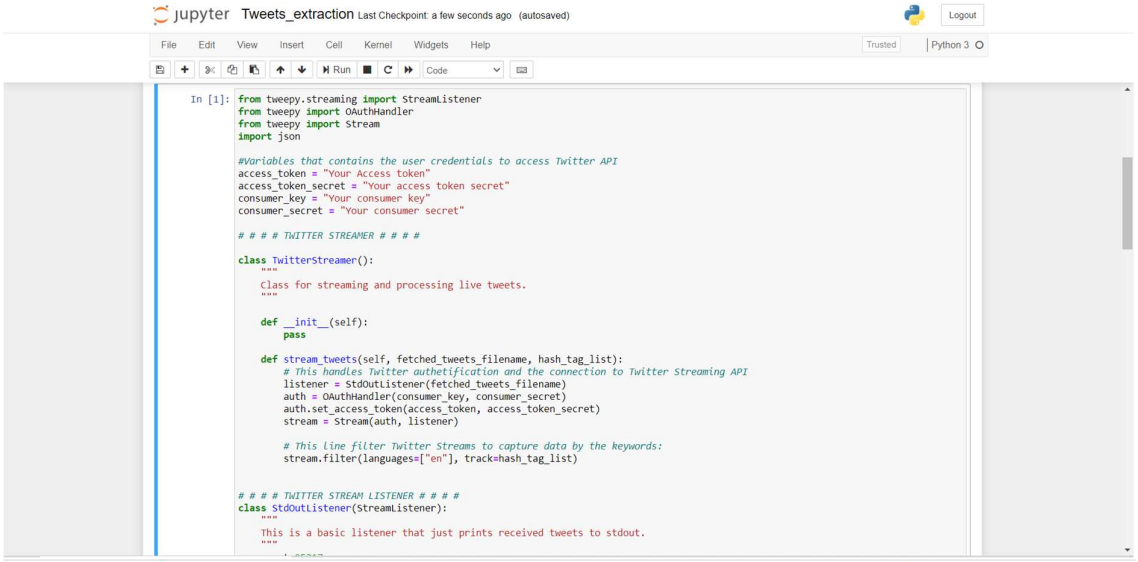

```
{
  "created_at":"Mon Oct 26 16:00:23 +0000 2020",
  "id":1320757176017784832,
  "id_str":"1320757176017784832",
  "text":"RT @JoeBiden: Character is on the ballot.\n\nVote: https:\/\/t.co\/eoxT07d7QB https:\/\/t.co\/PYq3FZTEia",
  "source":"\u003ca href=\"http:\/\/twitter.com\/download\/android\" rel=\"nofollow\"\u003eTwitter for Android\u003c\/a\u003e",
  "truncated":false,
  "in_reply_to_status_id":null,
  "in_reply_to_status_id_str":null,
  "in_reply_to_user_id":null,
  "in_reply_to_user_id_str":null,
  "in_reply_to_screen_name":null,
  "user":{
    "id":1340481373,
    "id_str":"1340481373",
    "name":"\ud83d\udc80\ud83c\udf83\ud83d\udd78\ufe0fKelli Crackel\ud83d\udc80\ud83c\udf83\ud83d\udd78\ufe0f",
    "screen_name":"KelliCrackel",
    "location":"Georgia, USA",
    "url":null,
    "description":"Wife to @felishaC14, Mom, Liberal, Redneck(yes really), Whovian, UGA football fan. Go Dawgs! #MeToo #GunControlNOW
          #NeverAgain #StrongerTogether",
    "translator_type":"none",
    "protected":false,
    "verified":false,
    "followers_count":17291,
    "friends_count":18725,
    "listed_count":15,
    "favourites_count":96739,
    "statuses_count":29117,
    "created_at":"Tue Apr 09 23:17:15 +0000 2013",
    "utc_offset":null,
    "time_zone":null,
    "geo_enabled":false,
    "lang":null,
  },
  "geo":null,
  "coordinates":null,
  "place":null,
  "contributors":null,
  "retweeted_status":{
    "created_at":"Mon Oct 26 16:00:02 +0000 2020",
    "id":1320757084225495042,
    "id_str":"1320757084225495042",
    "text":"Character is on the ballot.\n\nVote: https:\/\/t.co\/eoxT07d7QB https:\/\/t.co\/PYq3FZTEia",
    "display_text_range":[
      0,
      58
    ],
    "source":"\u003ca href=\"https:\/\/studio.twitter.com\" rel=\"nofollow\"\u003eTwitter Media Studio\u003c\/a\u003e",
    "truncated":false,
    "in_reply_to_status_id":null,
    "in_reply_to_status_id_str":null,
    "in_reply_to_user_id":null,
    "in_reply_to_user_id_str":null,
    "in_reply_to_screen_name":null,
    "user":{
      "id":939091,
      "id_str":"939091",
      "name":"Joe Biden",
      "screen_name":"JoeBiden",
      "location":"Wilmington, DE",
      "url":"http:\/\/joebiden.com",
      "description":"Senator, Vice President, 2020 candidate for President of the United States, husband to @DrBiden, proud father &
          grandfather. Loves ice cream, aviators & @Amtrak",
      "translator_type":"none",
      "protected":false,
      "verified":true,
      "followers_count":11546632,
      "friends_count":29,
      "listed_count":22750,
```

```
    "favourites_count":20,
    "statuses_count":6442,
    "created_at":"Sun Mar 11 17:51:24 +0000 2007",
    "utc_offset":null,
    "time_zone":null,
    "geo_enabled":false,
    "lang":null,
    "contributors_enabled":false,
    "is_translator":false,
  },
  "geo":null,
  "coordinates":null,
  "place":null,
  "contributors":null,
  "is_quote_status":false,
  "quote_count":2,
  "reply_count":9,
  "retweet_count":24,
  "favorite_count":110,
 ,
  "favorited":false,
  "retweeted":false,
  "possibly_sensitive":false,
  "filter_level":"low",
  "lang":"en"
},
"is_quote_status":false,
"quote_count":0,
"reply_count":0,
"retweet_count":0,
"favorite_count":0,
"entities":{
  "hashtags":[

  ],
  "urls":[
    {
      "url":"https:\/\/t.co\/eoxT07d7QB",
      "expanded_url":"http:\/\/iwillvote.com",
      "display_url":"iwillvote.com",
      "indices":[
        49,
        72
      ]
    }
  ],
  "user_mentions":[
    {
      "screen_name":"JoeBiden",
      "name":"Joe Biden",
      "id":939091,
      "id_str":"939091",
      "indices":[
        3,
        12
      ]
    }
  ],
  "symbols":[

  ],

"favorited":false,
"retweeted":false,
"possibly_sensitive":false,
"filter_level":"low",
"lang":"en",
"timestamp_ms":"1603728023901"
}
```

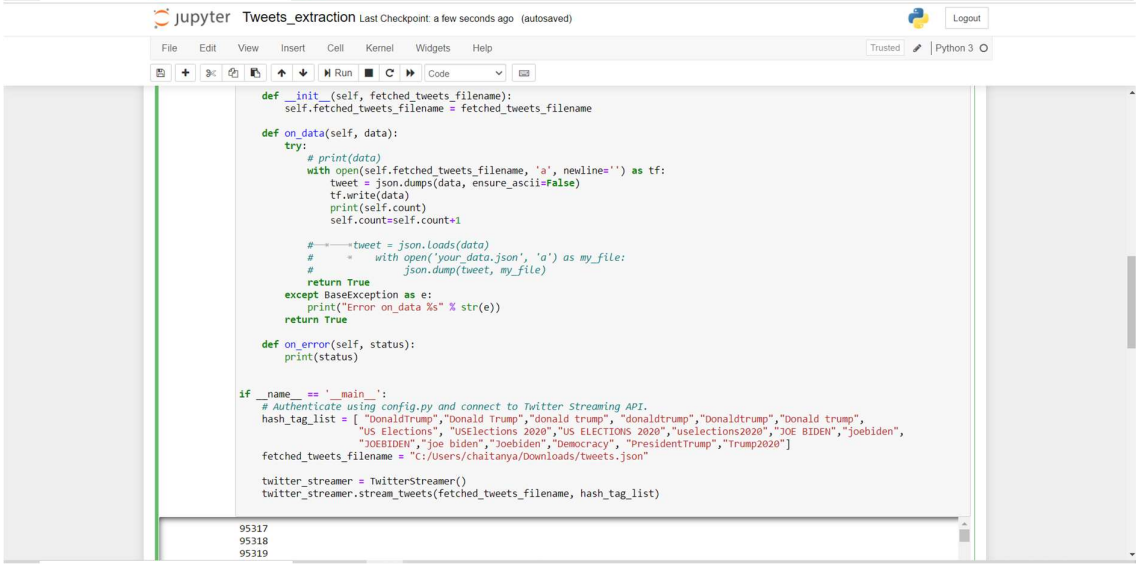
With the above data we can query the data as per our requirement and store the result. Some of the noticeable fields are retweeted, likes, text, timestamp, location, followers count, likes, geo coordinates, friends, dislikes, display name, username etc.

6. IMPLEMENTATION:

CODE FOR EXTRACTING TWEETS FROM TWITTER USING TWEEPY

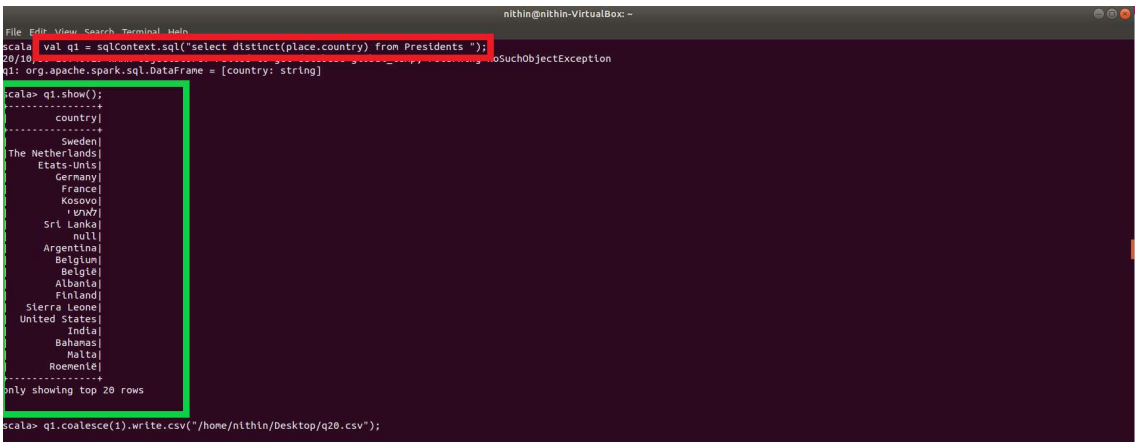


```
In [1]:
from tweepy.streaming import StreamListener
from tweepy import OAuthHandler
from tweepy import Stream
import json

#Variables that contains the user credentials to access Twitter API
access_token = "Your Access token"
access_token_secret = "Your access token secret"
consumer_key = "Your consumer key"
consumer_secret = "Your consumer secret"

# # # # TWITTER STREAMER # # # #

class TwitterStreamer():
    """
    Class for streaming and processing live tweets.
    """

    def __init__(self):
        pass

    def stream_tweets(self, fetched_tweets_filename, hash_tag_list):
        # This handles Twitter authetification and the connection to Twitter Streaming API
        listener = StdOutListener(fetched_tweets_filename)
        auth = OAuthHandler(consumer_key, consumer_secret)
        auth.set_access_token(access_token, access_token_secret)
        stream = Stream(auth, listener)

        # This line filter Twitter Streams to capture data by the keywords:
        stream.filter(languages=["en"], track=hash_tag_list)


# # # # TWITTER STREAM LISTENER # # # #
class StdOutListener(StreamListener):
    """
    This is a basic listener that just prints received tweets to stdout.
    """
```



```
    def __init__(self, fetched_tweets_filename):
        self.fetched_tweets_filename = fetched_tweets_filename

    def on_data(self, data):
        try:
            # print(data)
            with open(self.fetched_tweets_filename, 'a', newline='') as tf:
                tweet = json.dumps(data, ensure_ascii=False)
                tf.write(data)
                print(self.count)
                self.count=self.count+1

            #     tweet = json.loads(data)
            #     with open('your_data.json', 'a') as my_file:
            #         json.dump(tweet, my_file)
            return True
        except BaseException as e:
            print("Error on_data %s" % str(e))
        return True

    def on_error(self, status):
        print(status)


if __name__ == '__main__':
    # Authenticate using config.py and connect to Twitter Streaming API.
    hash_tag_list = [ "DonaldTrump","Donald Trump","donald trump", "donaldtrump","Donaldtrump","Donald trump",
                      "US Elections", "USElections 2020","US ELECTIONS 2020","uselections2020","JOE BIDEN","joebiden",
                      "JOEBIDEN","joe biden","Joebiden","Democracy", "PresidentTrump","Trump2020"]
    fetched_tweets_filename = "C:/Users/chaitanya/Downloads/tweets.json"

    twitter_streamer = TwitterStreamer()
    twitter_streamer.stream_tweets(fetched_tweets_filename, hash_tag_list)

95317
95318
95319
```

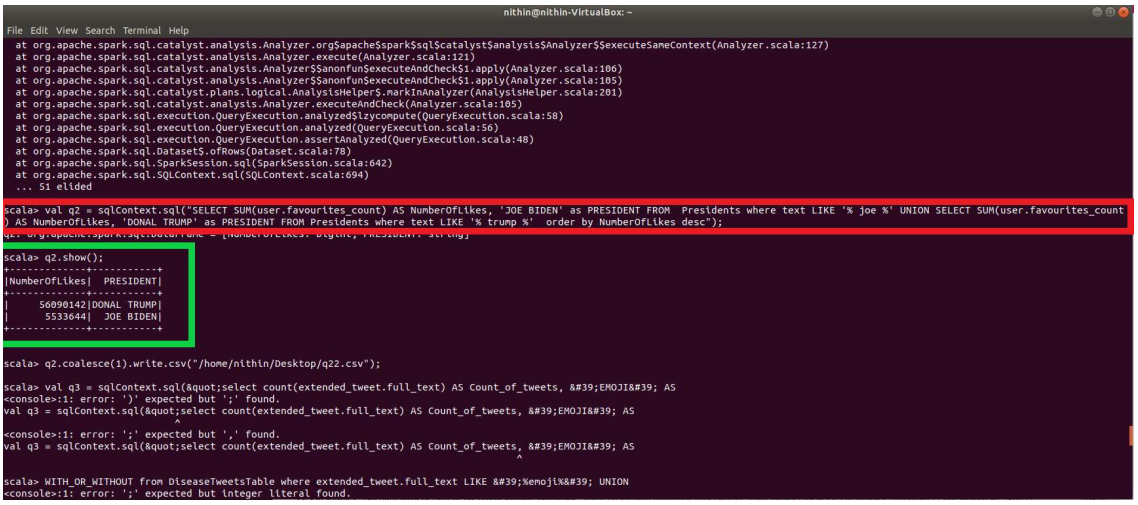
1. EXTRACTING USER LOCATION WHO TWEETED



```
nithin@nithin-VirtualBox: ~
File  Edit  View  Search  Terminal  Help
scala> val q1 = sqlContext.sql("select distinct(place.country) from Presidents ");
20/10/ ... NoSuchObjectException
q1: org.apache.spark.sql.DataFrame = [country: string]

scala> q1.show();
+--------------+
|       country|
+--------------+
|        Sweden|
|The Netherlands|
|     Etats-Unis|
|       Germany|
|        France|
|        Kosovo|
|        ישראל|
|     Sri Lanka|
|          null|
|     Argentina|
|       Belgium|
|        België|
|       Albania|
|       Finland|
|  Sierra Leone|
| United States|
|         India|
|       Bahamas|
|         Malta|
|      Roemenië|
+--------------+
only showing top 20 rows

scala> q1.coalesce(1).write.csv("/home/nithin/Desktop/q20.csv");
```

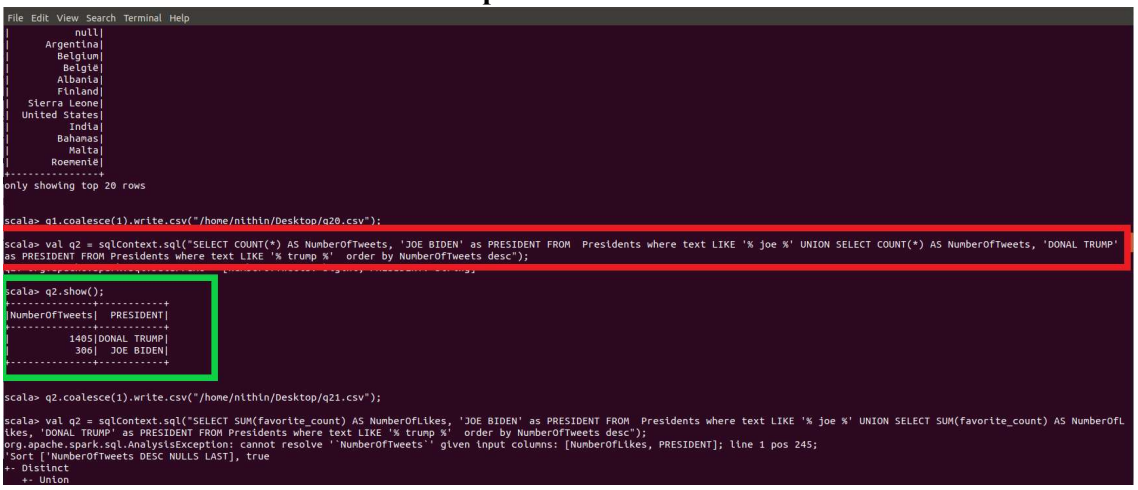
This is information will be useful to know where this topic is going trending.

2. NUMBER OF TWEETS FOR PARTICULAR CANDIDATE



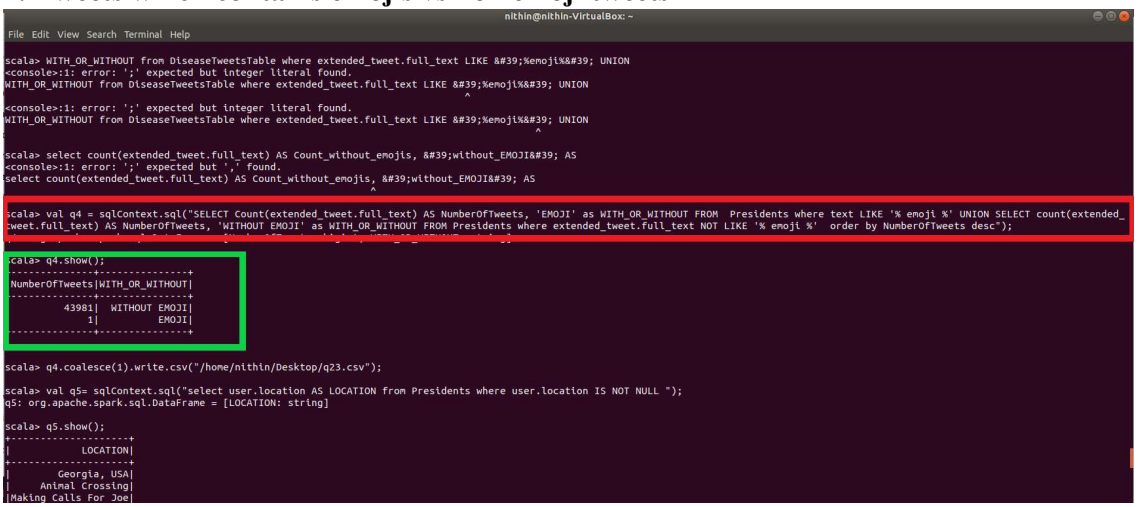
Through this query we can have a picture who is more trending on twitter, on further analysis we can say it whether it is positive or negative.

3. Number of Likes for Tweets about particular candidate.



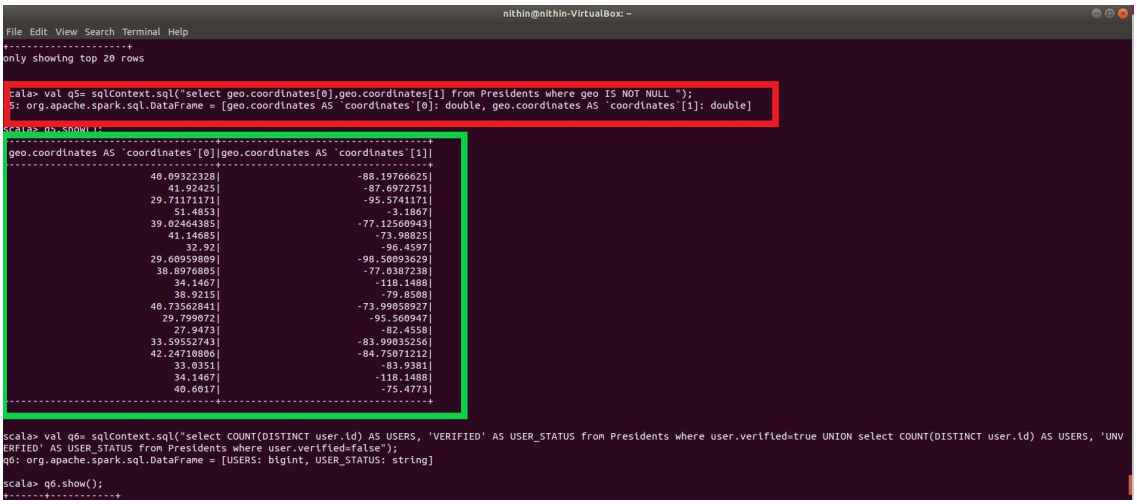
After analysing the data whether we have to know the support of people for that tweet, this query data can be vital in knowing that.

4. Tweets which contains emojis vs non emoji tweets



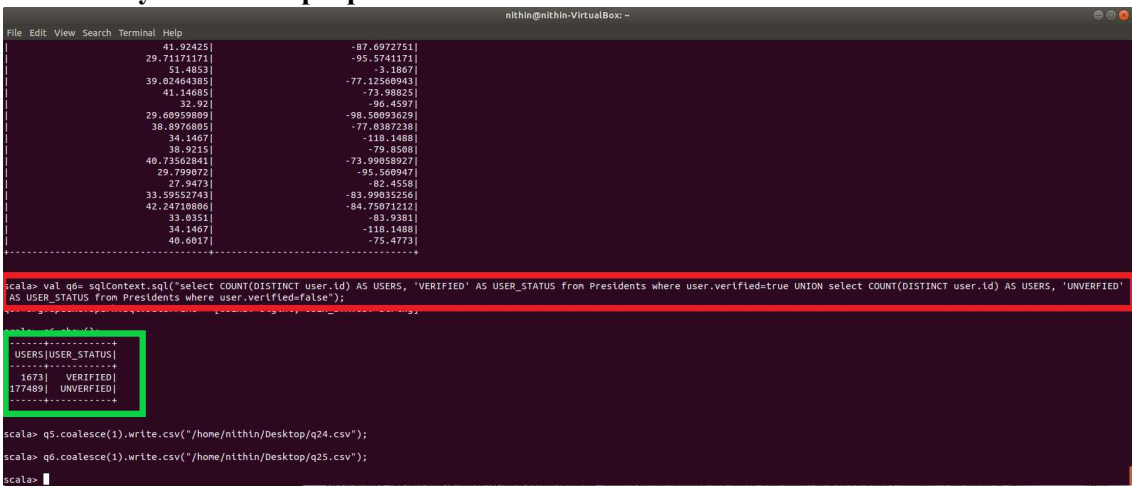
This query can be useful while processing the data(during sentimental analysis).

5. Geo location of user



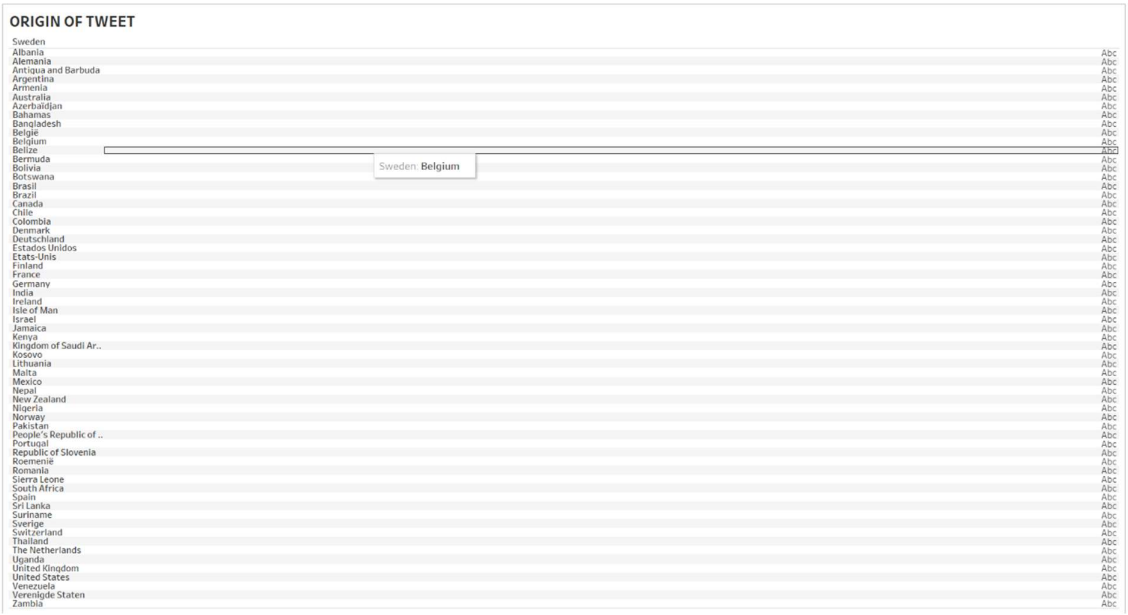
This can be helpful in knowing the hot spot about the discussion going about candidates.

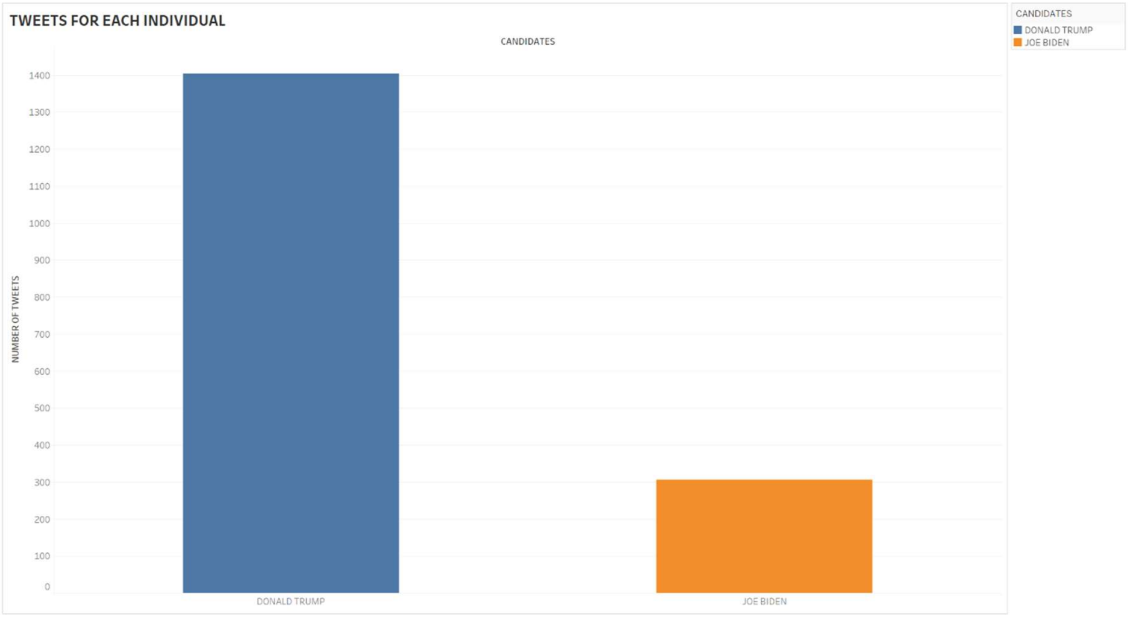
6. Celebrity vs Normal people.

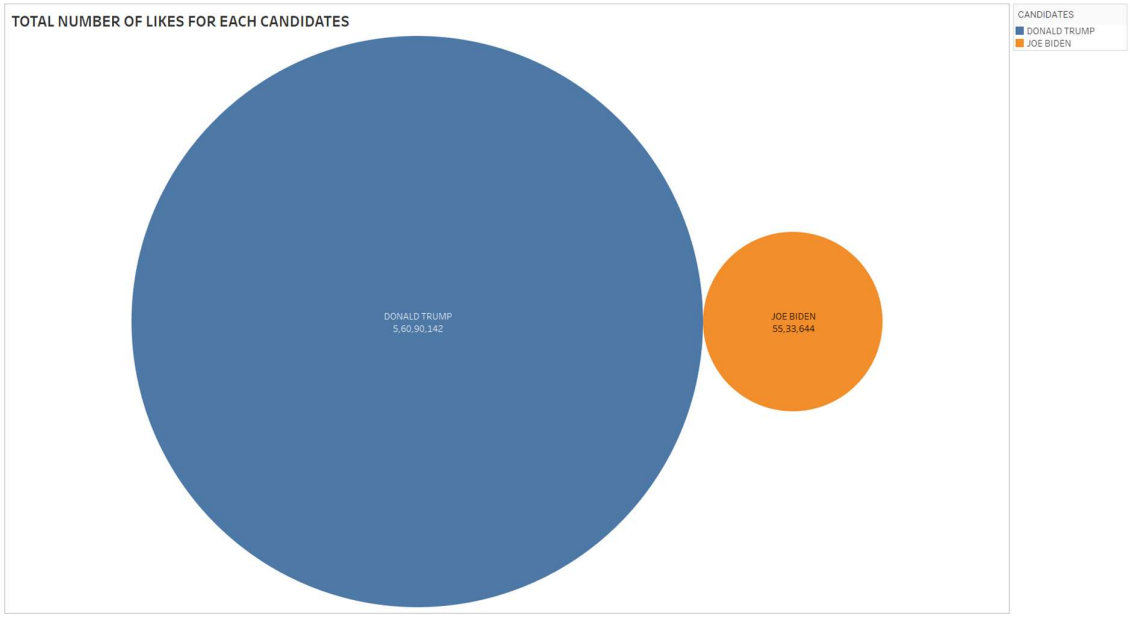


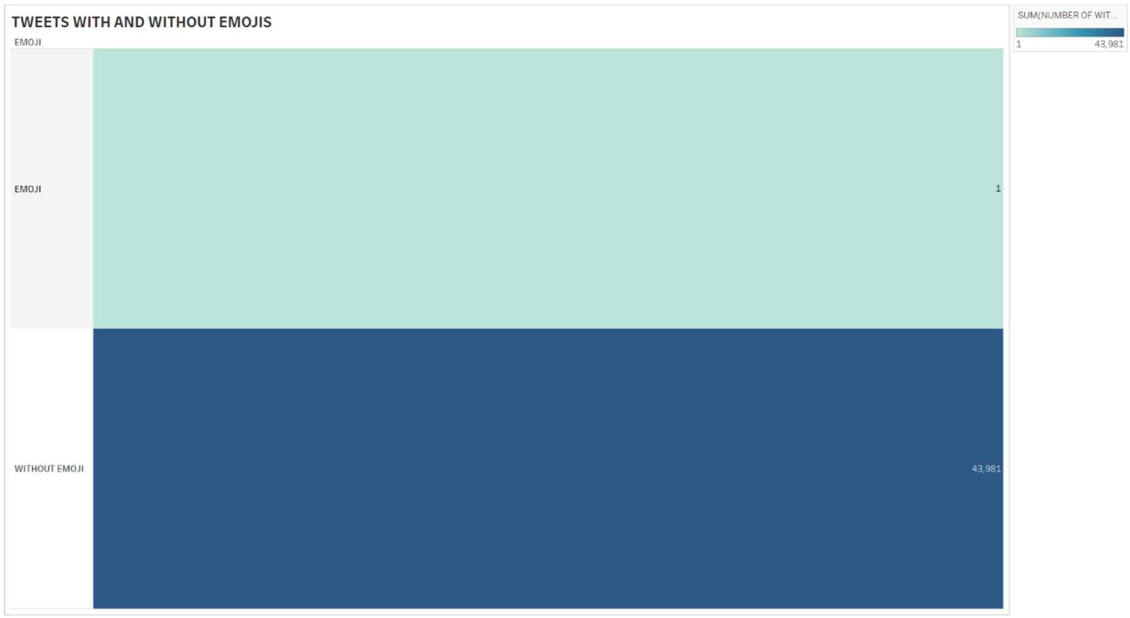
In general case we can trust more on tweets that is been given by celebrity. In order to give weightage to a tweet we can use this information.

7. Visualization Report:

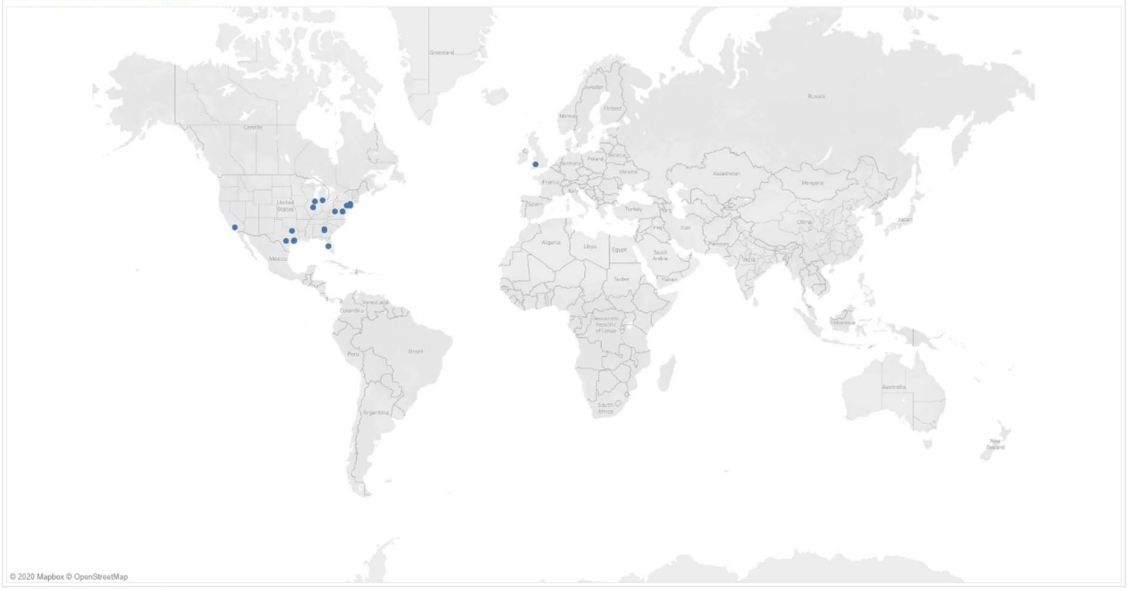




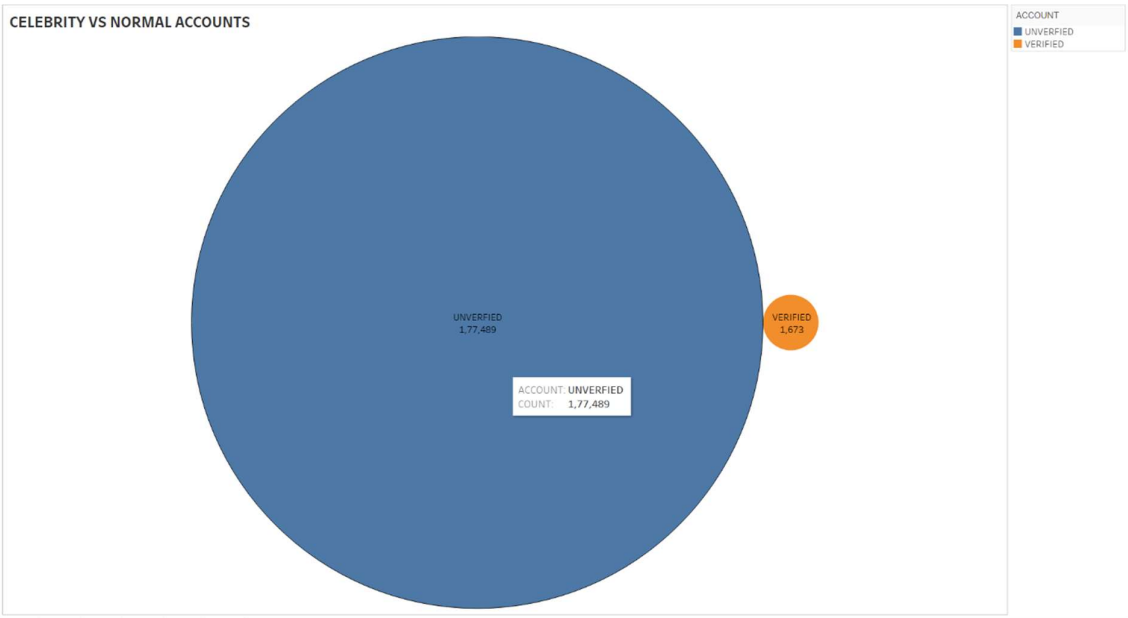

TOTAL NUMBER OF LIKES FOR EACH CANDIDATES
CANDIDATES
DONALD TRUMP
JOE BIDEN
DONALD TRUMP
5,60,90,142
JOE BIDEN
55,33,644


TWEETS WITH AND WITHOUT EMOJIS
EMOJI
SUM(NUMBER OF WIT...
1
43,981
EMOJI
1
WITHOUT EMOJI
43,981

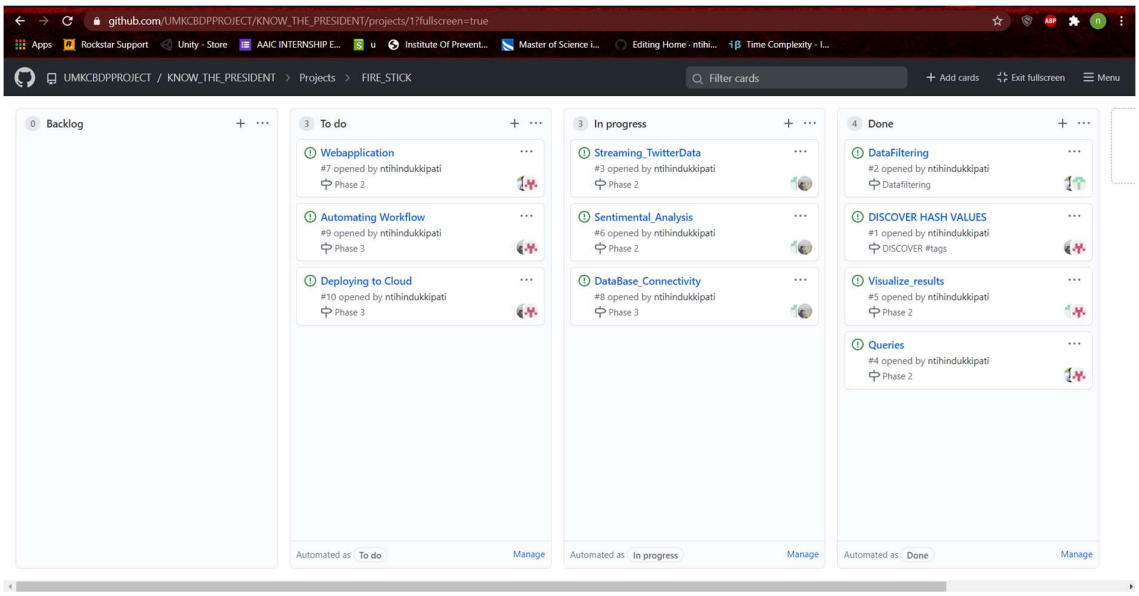
GEO-GRAPHICAL LOCATION



CELEBRITY VS NORMAL ACCOUNTS



PROJECT MANAGEMENT:



WORK COMPLETED:

1. Gathering requirements for the use case (Chaitanya, Eswar, Nikhitha, Nithin)
2. Tweets extracting using Hash Tags (Chaitanya, Nikhitha)
3. Querying the data using spark ETL in Scala prompt (Eswar, Nithin)
4. Visualizing data using Tableau (Eswar, Nithin)

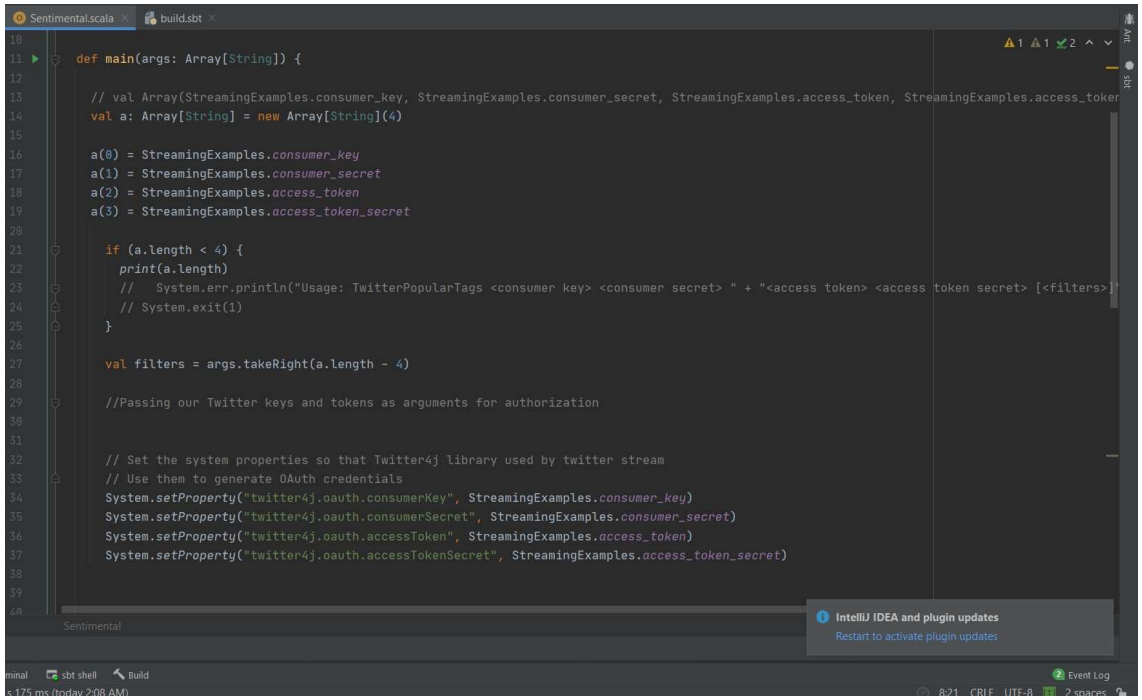
WORK IN PROGRESS:

1. Streaming data from twitter using Spark Streaming Library (Scala). (Chaitanya, Nithin)
2. Designing website for visualizing real time data. (Nikhitha, Eswar)

ISSUES AND CONCERNS:

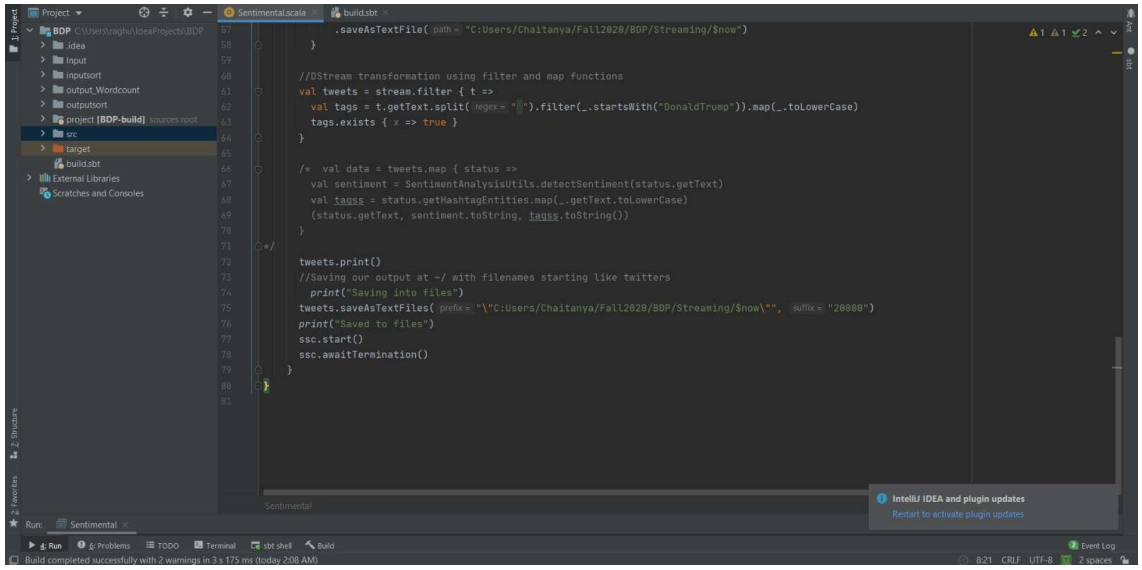
Spark Streaming library and dependency issues while performing Sentiment Analysis with streaming data

CODE:



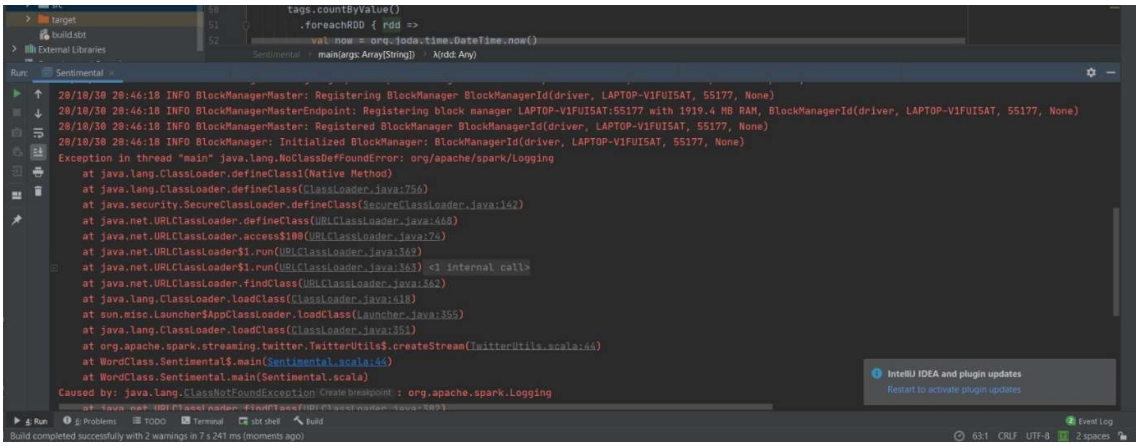
```
def main(args: Array[String]) {

  // val Array(StreamingExamples.consumer_key, StreamingExamples.consumer_secret, StreamingExamples.access_token, StreamingExamples.access_toker
  val a: Array[String] = new Array[String](4)

  a(0) = StreamingExamples.consumer_key
  a(1) = StreamingExamples.consumer_secret
  a(2) = StreamingExamples.access_token
  a(3) = StreamingExamples.access_token_secret

  if (a.length < 4) {
    print(a.length)
    //   System.err.println("Usage: TwitterPopularTags <consumer key> <consumer secret> " + "<access token> <access token secret> [<filters>]"
    // System.exit(1)
  }

  val filters = args.takeRight(a.length - 4)

  //Passing our Twitter keys and tokens as arguments for authorization


  // Set the system properties so that Twitter4j library used by twitter stream
  // Use them to generate OAuth credentials
  System.setProperty("twitter4j.oauth.consumerKey", StreamingExamples.consumer_key)
  System.setProperty("twitter4j.oauth.consumerSecret", StreamingExamples.consumer_secret)
  System.setProperty("twitter4j.oauth.accessToken", StreamingExamples.access_token)
  System.setProperty("twitter4j.oauth.accessTokenSecret", StreamingExamples.access_token_secret)
```

```
val sparkConf = new SparkConf().setAppName("twitterSentiment").setMaster("local[2]")
  val ssc = new StreamingContext(sparkConf, seconds( seconds = 5))
  val stream = TwitterUtils.createStream(ssc,None)

  //Input DStream transformation using flatMap
  val tags = stream.flatMap { status => status.getHashtagEntities.map(_.getText) }

  //RDD transformation using sortBy and then map function
  tags.countByValue()
    .foreachRDD { rdd =>
      val now = org.joda.time.DateTime.now()
      rdd
        .sortBy(_._2)
        .map(x => (x, now))
        //Saving our output to local directory
        .saveAsTextFile( path = "C:Users/Chaitanya/Fall2020/BDP/Streaming/$now")
    }

  //DStream transformation using filter and map functions
  val tweets = stream.filter { t =>
    val tags = t.getText.split( regex = " ").filter(_.startsWith("DonaldTrump")).map(_.toLowerCase)
    tags.exists { x => true }
  }
```



```
.saveAsTextFile( path = "C:Users/Chaitanya/Fall2020/BDP/Streaming/$now")
    }

  //DStream transformation using filter and map functions
  val tweets = stream.filter { t =>
    val tags = t.getText.split( regex = " ").filter(_.startsWith("DonaldTrump")).map(_.toLowerCase)
    tags.exists { x => true }
  }

  /*  val data = tweets.map { status =>
    val sentiment = SentimentAnalysisUtils.detectSentiment(status.getText)
    val tagss = status.getHashtagEntities.map(_.getText.toLowerCase)
    (status.getText, sentiment.toString, tagss.toString())
  }
*/

  tweets.print()
  //Saving our output at ~/ with filenames starting like twitters
    print("Saving into files")
  tweets.saveAsTextFiles( prefix = "\"C:Users/Chaitanya/Fall2020/BDP/Streaming/$now\"", suffix = "20080")
  print("Saved to files")
  ssc.start()
  ssc.awaitTermination()
}
}
```

ISSUE:



8. STORY

- **WHO**?
  The dataset is about Joe Biden and Donald Trump related to US 2020 Elections, there are no under sampled data and over sampled data. There is no risk in disclosing information as there is no sampling issue.
- **WHAT?**
  From the collected data noticeable fields like retweeted, likes, text, timestamp, location, followers count, likes, geo coordinates, friends, dislikes, display name, username etc. are queried and visualized.
- **WHEN?**
  During US ELECTION 2020 the data was streamed form the twitter mostly real time data, the data collected is longitudinal data as it is only collect at this point of particular time across the world. The generalisation can be done whenever required for at different point of events by changing the hashtags while collecting the data.
- **WHERE?**
  The event is taking place in US, this data is been streamed from twitter over time, This data is collected across the globe and contains GIS information of user like geo coordinates. The generalisation can be done wherever required for at different locations by changing the hashtags while collecting the data.
- **WHY?**
  The data was collected to bring awareness among the people about the US elections by considering various attributes.

REFERENCES:

https://medium.com/@anicolaspp/spark-streaming-and-twitter-sentiment-analysis-c860938d484

https://www.youtube.com/watch?v=uD_q4Rm4i2Q

https://www.edureka.co/blog/spark-streaming/

https://stackoverflow.com/questions/31466435/how-to-find-source-of-scala-matcherror

https://github.com/PacktPublishing/Hands-On-Deep-Learning-with-Apache-Spark/issues/1