horizontal line

**FINAL REPORT**



**Twitter Tweets Analysis**

# 

# **ABSTRACT**

This project investigates how to organize and classify a large collection of Twitter data, working with a dataset of almost 3 Lakhs tweets collected directly from Twitter. The approach described here combines spatial analysis of the location of the tweet with content/sentiment analysis of the text and hashtags associated with the same tweet. We then look into those specific regions to try and identify the most popular Political Party and its influence on the regions and compare differences between each city, to provide a better plan for the Political Party Campaign. We expect our results to vary based on where people are tweeting from and which Political Party has more impact in the city.

# **Objective/Aim**

# Twitter is an American online news and social networking service on which users post and interact with messages known as "tweets". It is a platform widely used by people to express their opinions and display sentiments on different occasions. Sentiment analysis is extremely useful in social media monitoring as it allows the company/product or in our case the Political Party to gain an overview of the wider public opinion behind certain topics. Tweets sentiment analysis is an application of sentiment analysis on data from Twitter (tweets), in order to extract sentiments conveyed by the user. In the past decades, the research in this field has consistently grown due to the great increase in the usage of social media platforms. In our project, we aim to analyze tweets Sentiment to figure out if a Political Party is ahead of its competitor in multiple cities using the location provided in the tweets data and the sentiment analysis of the tweet.

# 

# **Group Members**

|  |  |  |
| --- | --- | --- |
| **Enrollment Number** | **Name** | **Batch** |
| 16104033 | Sarthak Aggarwal | B-12 |
| 16104032 | Vartul Tripathi | B-12 |
| 16104030 | Manushivam Maheshwari | B-12 |
| 16104026 | Akash Arora | B-12 |

# **BACKGROUND STUDY & FINDINGS**

We have used Twitter data as the dataset. Twitter is a gold mine of data. Unlike other social platforms, almost every user’s tweets are completely public and pullable. It is a huge plus point for us as there is a large amount of data to run analytics on. Twitter data is also pretty specific. Twitter’s API allows you to do complex queries like pulling every tweet about a certain topic within the last twenty minutes or pull a certain user’s non-retweeted tweets. A major application of this is analyzing what people of a particular country think about the active parties of that respective country by collecting the tweets of those people and applying sentiment analysis over it. We collected the tweets of **BJP**(Bhartiya Janta Party) and **Congress** Party which are active and major parties of INDIA. We have collected about 3.5 lac tweets of BJP and 1.5 lac tweet of Congress party.

Twitter data is used by almost every company/organization to know about the trends about their product or anything else. Tweets convey the true sentiment of its user, so it become easier for the organization to make decisions accordingly.

There are some great research which uses the tweets data to either predict something or help a organisation perform better. Spatial and Temporal Sentiment Analysis of Twitter data, Zhiwen Song and Jianhong (Cecilia) Xia\*, uses the twitter data to analyse the trends within the university according to the sentiment of the tweets. On one the major challenge for us was to calculate sentiments of the tweets written in Hinglish language(Hindi language in the mode of English alphabets). To resolve this issue we converted Hinglish text into English and then found the sentiment of the tweet.

**Designing In Details**

**Tweets Collection:**

* To collect the Tweets from Twitter, we used the API provided by Twitter itself ‘ Tweepy’. Tweepy provides us 2 methods to fetch tweets from Twitter. One of them is using live stream i.e., it provides us with live tweets, with a delay of maximum 20 minutes, the other way to access is Cursor which provides us the tweets from its dataset. Using the Cursor method we can apply different filters and fetch the tweets. We used the Cursor method to collect the Tweets relates to two major Political Parties in India, Bhartiya Janta Party(BJP) and Congress. We used different keywords related to the political parties to collect their own dataset of tweets along with its location and time stamp. Since 1 in every 60000 tweets contain the exact coordinate of the user’s location, we had to use the home city location of the user. To limit the collection of tweets in India we had to define radius under which we need the tweets because specifying the country name wasn’t effective so the radius under which the whole country in our case India would be circumcised and only relevant tweets will be fetched. Code used to fetch tweets:

|  |
| --- |
| import tweepy import csv import pandas as pd  access\_token = "XXXX" access\_token\_secret = "XXXX" consumer\_key = "XXXX" consumer\_secret = "XXXX"   auth = tweepy.OAuthHandler(consumer\_key, consumer\_secret) auth.set\_access\_token(access\_token, access\_token\_secret) api = tweepy.API(auth,wait\_on\_rate\_limit=True)  bjp = 'bjptweets.csv' congress = 'congresstweets.csv'  bjp\_keywords = '#BJP OR #bjp OR BJP OR Modi OR modi OR abkibaarmodisarkar'  congress\_keywords = '#congress OR #CONGRESS OR RahulGandhi OR Rahul OR priyankagandhi'  csvFile = open(bjp, 'a') csvWriter = csv.writer(csvFile)  for tweet in tweepy.Cursor(api.search,q=bjp\_keywords,geocode="21.146633,79.088860,1756km",count=100,lang="en").items():   print (tweet.created\_at, tweet.text, tweet.user.location, tweet.user.followers\_count)  csvWriter.writerow([tweet.created\_at, tweet.text.encode('utf-8'),tweet.user.location.encode('utf-8'), tweet.user.followers\_count]) |

**Data Pre-Processing**

After tweets were fetched, they were in irregular. Tweets contained multiple languages, emojis, different types of links, irrelevant location, all of which needed to be removed. The dataset required some pre-processing before any analysis can be done.

**Step - 1**

**Irrelevant Location**

* All the irrelevant tweets were removed on the basis of locations. The locations containing null values and irrelevant values of location for example: “behind you”, “somewhere in the world” were removed.
* Some of the locations where even in Hindi so the processing on the basis of Hindi location was also counted as irrelevant and hence was deleted during the process.

**Step - 2**

**Tweet Content**

* After the irrelevant locations were removed from our dataset, now we removed hashtags, emojis, special characters, tag names using the basic Regex.

|  |
| --- |
| def clean\_tweet(tweet):  '''  Utility function to clean tweet text by removing   links, special characters  using simple regex statements.  ''' return''.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z\t])|(\w+:\/\/\S+)", " ", tweet).split()) |
|  |

**Step - 3**

**Hinglish**

* Many of the tweets were also in Hinglish( Hindi language in the mode of English alphabets) so they needed to be translated in English for the sentiment analysis to work more accurately. This was done by using Googletrans Librarywhich converted the Hinglish tweets into English.

|  |
| --- |
| from googletrans import Translator translator = Translator(sentence) |

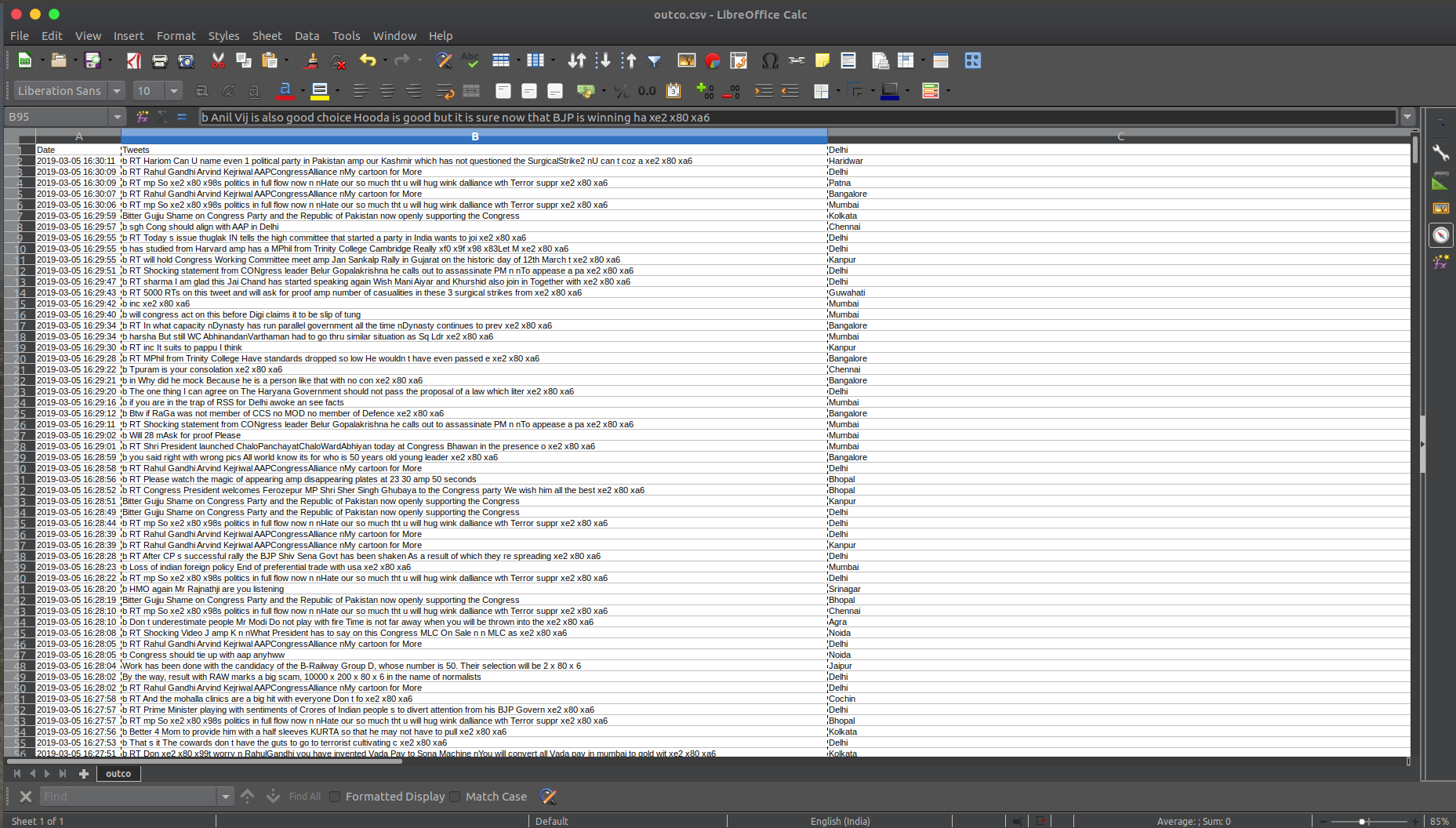
**DATASET**

* We initially collected around 1.7 Lakhs each of both BJP and Congress related tweets and after all the pre-processing methods we were left with the around 1.2 Lakhs tweets each. Following is the screenshot of the datasets :

**Tweets related to BJP(Around 1.2 lakhs):**

# 

# **Tweets related to Congress(Around 1.2 lakhs):**



**Implementation**

**Sentiment Analysis**

After the fetching and pre-processing of the tweets, now need to find the sentiment of each tweet for which we apply sentiment analysis. Using a python library named ‘Texblob**’**, used for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more. The sentiment property returns a named tuple of the form Sentiment ( polarity, subjectivity ). The polarity score is a float type within the range [-1.0, 1.0], were -1.0 refers to very negative sentiment and +1.0 refer to positive sentiment. The subjectivity is a float within the range [0.0, 1.0] where 0.0 is very objective and 1.0 is very subjective. Following code represents the sentiment analysis:

|  |
| --- |
| import csv import re import pandas as pd import numpy as np from textblob import TextBlob from googletrans import Translator  bjp = 'bjptweets.csv' congress = 'congresstweets.csv' outcongress = 'outcongress.csv' outbjp = 'outbjp.csv;  def clean\_tweet(tweet):  '''  Utility function to clean tweet text by removing links, special characters  using simple regex statements.  '''  return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)", " ", tweet).split())   with open(outbjp,'w') as writeFile:  with open(bjp, 'r') as readfile:  rows = csv.reader(readfile)  writer = csv.writer(writeFile)  writer.writerow(['Date', 'Tweets', 'Location', 'Sentiment Polarity'])  for row in rows:  sentence = row[1]  sentence = clean\_tweet(sentence)  a = Translator().translate(sentence)  sentence = a.text      blob = TextBlob(sentence)  print (sentence)  print (blob.sentiment.polarity)    date = str(row[0])  twe = str(a.text)  loc = str(row[2])  pol = float(blob.sentiment.polarity)   writer.writerow([date, twe, loc, pol]) |

After the sentiment analysis, we had polarity for each and every tweet in our database.

**Clustering**

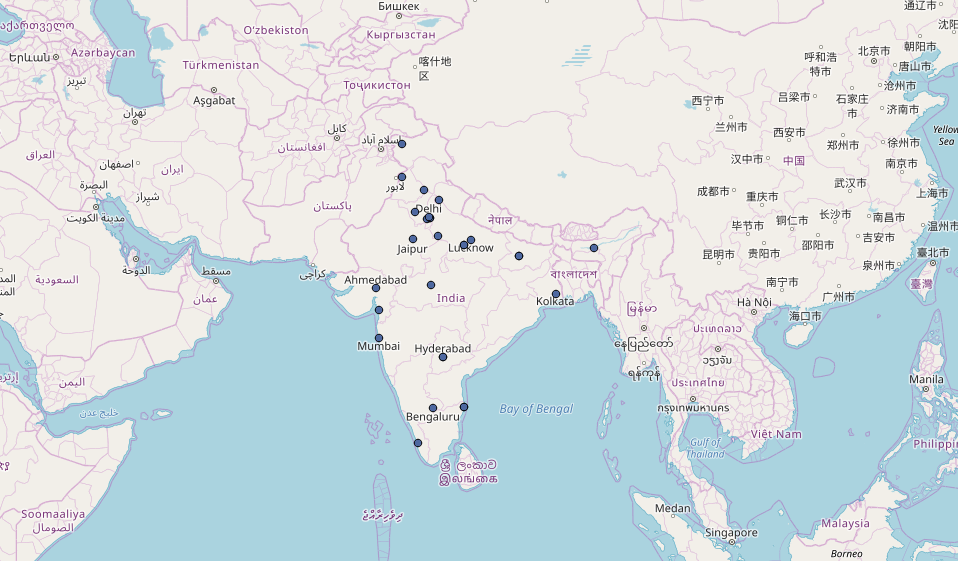
Now we applied DBSCAN algorithm to cluster the the dataset according to the cities and along with it we counted the total numbers of positive polarity, negative polarity, and neutral polarity.

|  |
| --- |
| #import libraries import numpy as np import matplotlib.pyplot as plt import pandas as pd from sklearn.cluster import DBSCAN from sklearn.preprocessing import StandardScaler  # Importing the dataset bjp = 'countbjp.csv' congress = 'countcongress.csv' dataset = pd.read\_csv(bjp) y=pd.DataFrame(dataset.iloc[:, 10].values) X=pd.DataFrame(dataset.iloc[:, 0:10].values)  from sklearn.preprocessing import LabelEncoder, OneHotEncoder labelencoder\_X = LabelEncoder() X.values[:, 2] = labelencoder\_X.fit\_transform(X.values[:, 2]) onehotencoder = OneHotEncoder(categorical\_features= [0]) X= onehotencoder.fit\_transform(X).toarray()  #data scaler scaler = StandardScaler() X\_scaled = scaler.fit\_transform(X)  # cluster the data into clusters dbscan = DBSCAN(eps=0.123, min\_samples = 2) clusters = dbscan.fit\_predict(X\_scaled) |

**Plotting**

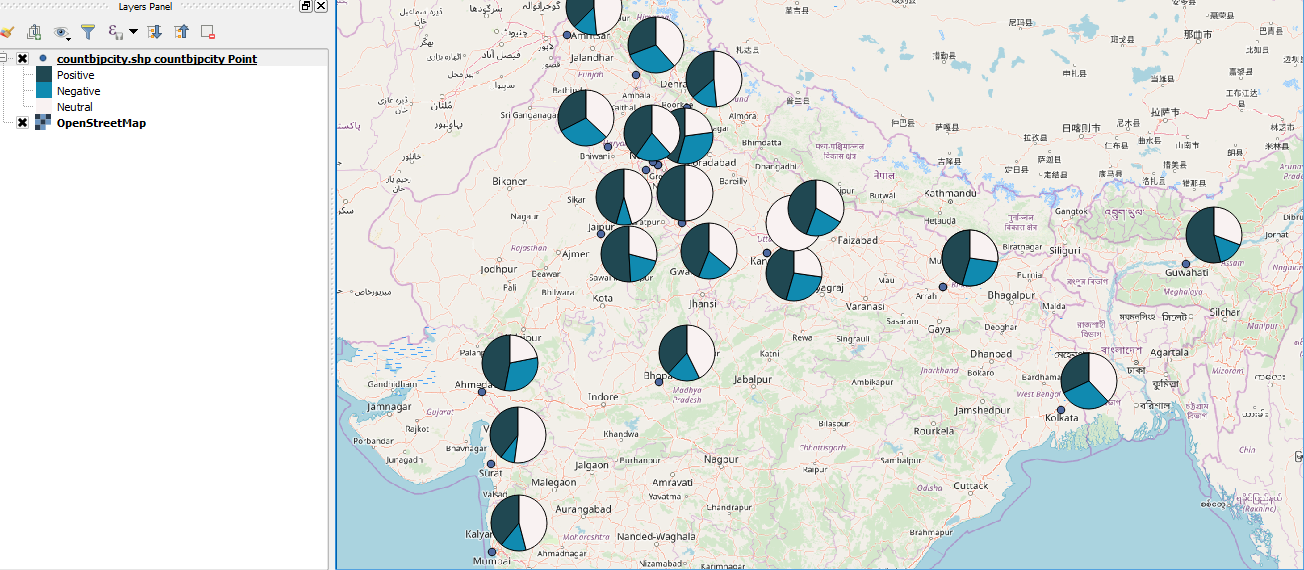
Our platform for plotting our final sentiments of respective tweets is done over **QGIS**, **geographic information system** (GIS) software, allows us to analyze and edit spatial information, in addition to composing and exporting graphical maps.QGIS supports both raster and vector layers; vector data is stored as either point, line, or polygon features. Multiple formats of raster images are supported, and the software can georeference images.

* Plotting is done over QGIS using MMQGIS plugins and Open Layer plugins used for manipulating vector layers over QGIS using open map api. It provides an alternative to the native QGIS vector plugin set, with verbose progress reporting, an intuitive user interface, direct shapefile/CSV-file access, and some additional capabilities missing from other plugin sets.
* The following image shows only the plotting of all the respective cities from which the person have tweeted from our pre-processed dataset.

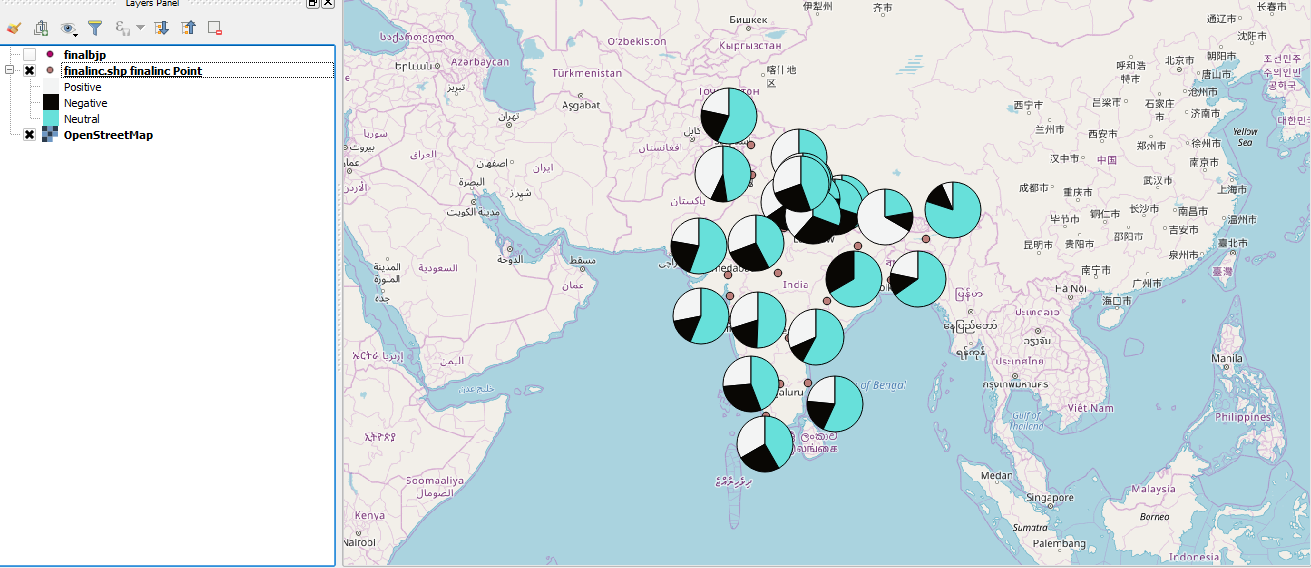


* This screenshot below shows the sentiment(positive, negative and neutral) of every tweet of all the cities using a pie chart. For example: if we found around 1000 tweets from New Delhi then we’ll collect the sentiment of all those tweets and show those collective sentiments in the map using QGIS with the pie-chart in map respectively. Same work is been executed for the rest of the mentioned cities in our pre-processed dataset on the basis of their locations under the radius mentioned, which has circumcised the whole country.

**CONGRESS PARTY TWEETS ANALYSIS**



**BJP(Bhartiya Janta Party) TWEETS ANALYSIS**

****

**Conclusion**

The approach described combines spatial analysis of the home location of a Twitter user, with content analysis of the text associated with the same tweet. Comparisons were made between the most popular Political Party in India’s different cities to make their political campaign more successful. However, further analysis and refinement of the algorithms used may be necessary to accurately draw meaningful conclusions beyond that, as the content of tweet also inform the issues that people are facing which can be resolved by the Political Parties or can be added into their manifesto.

Our analysis shows how much people of India are devoted to respective parties on the basis of their own words which we converted into sentiments(positive, negative and neutral) using sentiment analysis and then showing the collective tweets analysis from that particular city using MMQGIS plugins over QGIS platforms.

It is unfortunate that location-enabled tweets are currently sparse. This is why gathering a high volume of location-enabled data is a big challenge. Twitter users generally do not share their geolocation. Twitter API also does not allow the access of location-related information such as server origin for user privacy. To overcome this problem, future work aims to develop and integrate an algorithm content-based approach to determine a user’s location. This algorithm will examine a user’s profile, geographic region, hierarchical location, home location, travel location, city location, time zone, zip code or postal code to predict the user’s location.

# 

# 

# **REFERENCES**

* Using geolocated Twitter data to monitor the prevalence of healthy and unhealthy food references across the US, Michael J. Widener a , Wenwen Li b
* Spatial Data Mining: Clustering of Hot Spots and Pattern Recognition, Seng Chuan TAY , Wynne HSU , Kim Hwa LIM , Lee Chen YAP
* Geospatial Clustering and Classifying of Twitter Data, Taylor Graham, University of Colorado
* <https://www.researchgate.net/publication/307382684_Twitter_Sentiment_Analysis_of_the_2016_US_Presidential_Election_Using_an_Emoji_Training_Heuristic>.
* A Geospatial Analysis of Oppressive Language On Twitter, Forrest Tagg Ridler, University of Colorado Boulder, [forrest.ridler@colorado.edu](mailto:forrest.ridler@colorado.edu)
* Spatial and Temporal Sentiment Analysis of Twitter data, Zhiwen Song and Jianhong (Cecilia) Xia\*, Department of Spatial Sciences, Curtin University
* A Survey on Sentiment Analysis on Twitter Data Using Different Techniques Bholane Savita Dattu, Prof.Deipali V. Gore
* LOCATION BASED POPULARITY ANALYSIS OF TWITTER DATA By S.M. Rashel Rana Bachelor of Science in Computer Science and Engineering DIU, Bangladesh, 2004.
* <https://www.tweepy.org/>
* <https://pypi.org/project/googletrans/>