Individual Project: Discover and Predict Election Trend based on Twitter data

Naga Janaki Dwadasi

Department of Computer Engineering

San Jose State University

San Jose, CA

[nagajanaki.dwadasi@sjsu.edu](mailto:nagajanaki.dwadasi@sjsu.edu)

<https://github.com/NagaJanakiDwadasi/ElectionPrediction>

***Abstract*** - **Election trend is predicted by performing a sentiment analysis on public opinion of the possible presidential candidates. Twitter is a prominent social network that contains current public opinions of the candidates. Dataset is obtained from Twitter and a lexicon based analysis is performed to determine the polarity of the tweets. Further, a comparison is made among the possible Democratic and Republican members to predict and plot the results on the USA map.**

**INTRODUCTION**

Sentiment analysis is the use of Natural Language Processing(NLP) to categorize opinions expressed by speaker, writer with regard to a topic. Twitter is one of the most popular microblogging and social networking service with millions of people posting their opinions on it every minute. In the current age of social media, Twitter is a rich source of information for opinion of the crowd. Dataset is scraped from twitter using Twitter Scrapper between the time period of 2018-2019. This data is used to perform sentiment analysis through lexicon based approach for determining the polarity. Difference between the positive and negative tweets is calculated for each state to plot the dominance accordingly on the USA map. The workflow of the project is shown in figure 1.

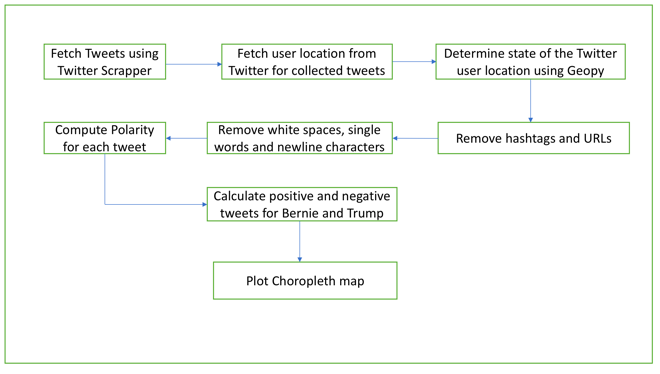


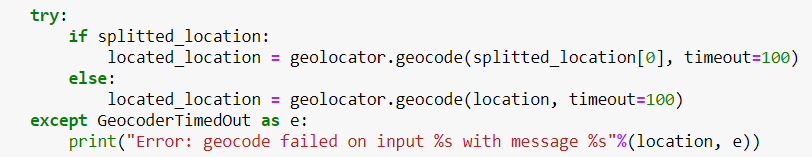
Fig (1) Workflow

**LITERATURE REVIEW**

In **[1]** sentiment analysis of tweets from twitter were collected between the time period March 2016-2017 and results were provided with an overall count of training and testing data for candidates. From 2016 election we see the majority of votes are gained by Clinton but the majority of states were won by Trump which led to his presidential win. Hence, I would like to improvise the same by collecting data specific to a location and show results comparing each candidate and who has more chances in each location. **[2]** paper has used TextBlob API to find polarity of a tweet, which is incroporated in this paper to find tweet polarity.

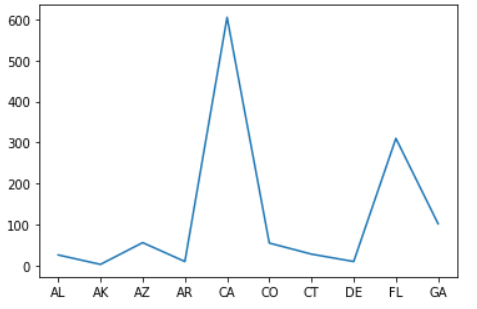
**DATA COLLECTION**

Data is collected from Twitter by using Twitter Scrapper library [3]. The data is scraped within a timeperiod of 2018-2019. Total data collected intially for this time span is 1,50,000 tweets for Donald Trump and 1,20,000 tweets for Bernie Sanders. A sample of 25,000 tweets for Donald Trump and 22,500 tweets for Bernie Sanders is used further to filter for user location to be within USA. Geopy package is used to identify location. From the sampled data, final data with locations in USA is 6,602 tweets for Donald Trump and 6,648 tweets for Bernie Sanders. Figure 2 shows the code to extract and split tweets by location.

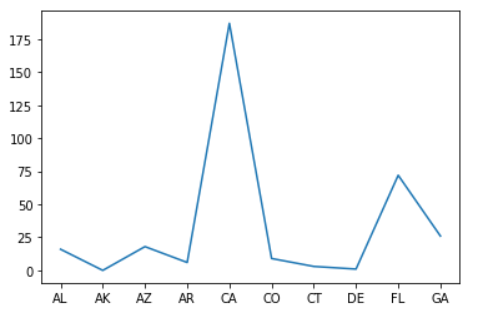


Fig(2) Code to identify location

Figure 3.1 and 3.2 shows a part of the location wise split tweets count from dataset of Donald Trump.



Fig(3.1) Location vs PositiveTweets



Fig(3.2) Location vs NegativeTweets

**DATA PREPROCESSING**

Tweets may contain hashtags, URLs, trailing whitespaces and newline characters. The preprocessing module performs the cleaning process using NLTK library for retrieving the meaningful parts of the tweet by removing the unnecessary content.

**IMPLEMENTATION**

TextBlob is a python library for processing textual data. It provides API to work on NLP(Natural Language Processing) tasks like sentiment analysis, classification and translation etc.. Using TextBlob, polarity of the tweets is identified figure 4. The polarity data is then used to find the count of number of positive and negative tweets for each state for a candidate. This data with count of positive and negative number of tweets for each state is loaded to dataframes for Donald Trump and Bernie Sanders. Figure 5 shows the first five records from the Donald Trump’s dataframe.

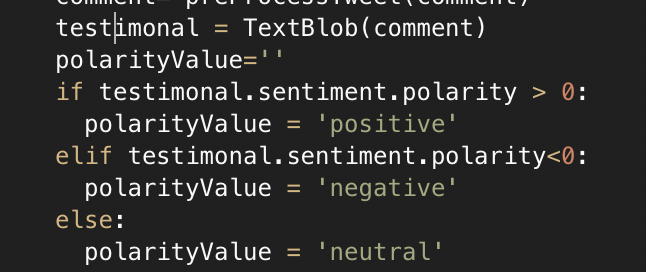
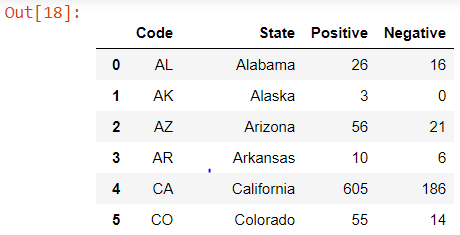


Fig (4) Code to find polarity of text

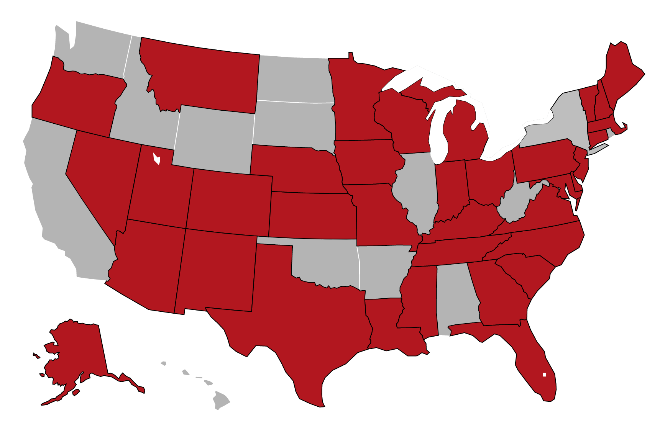


Fig(5) Records from Trump Dataframe

Now, the difference between their positive and negative scores for each state is calculated individually to get their final scores. This final score is used to predict their dominance in a particular state.

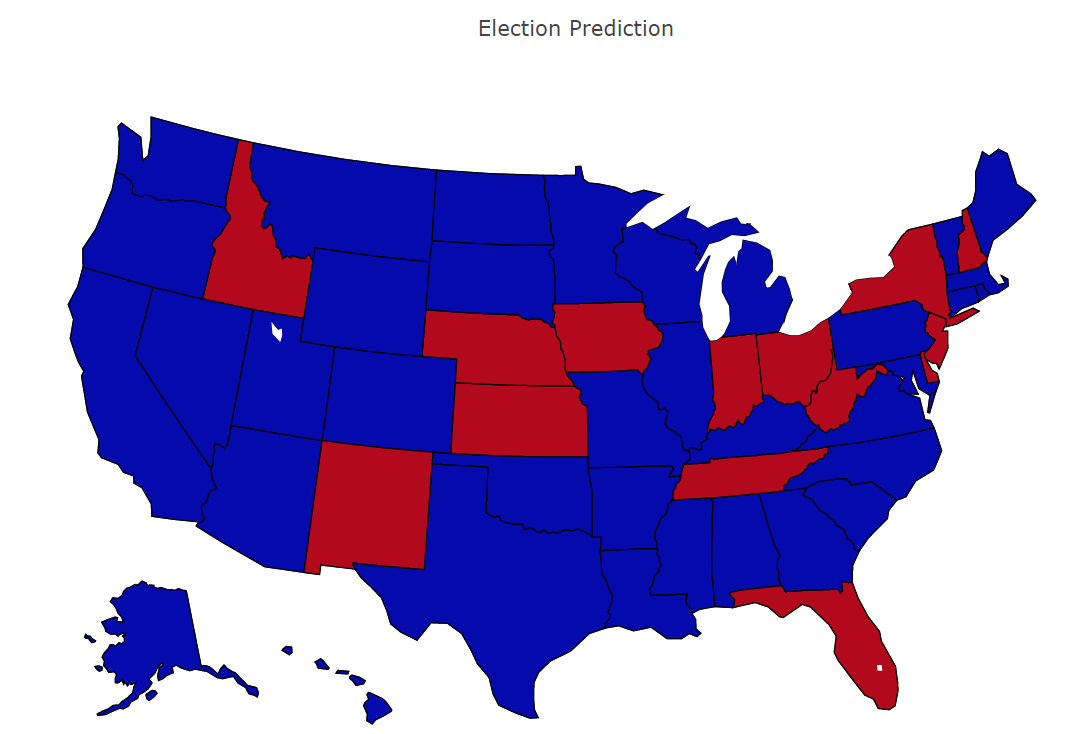
**RESULTS**

Plotly python is a open source graphing library for plotting interactive graphs or charts. Plotly package and choropleth mapis used to plot the results on the USA map. Results by using only Donald Trump’s dataframe is shown in figure 6 which shows the states in which his dominance is highlighted in red colour.



Fig(6) Donald Trump dominance plot

Results by using both Donald Trump and Bernie Sander’s dataframes together is shown in figure 7 which shows their respective dominance in red and blue colours.



Fig(7) Bernie Sanders vs Donald Trump

**CONCLUSION**

In this project, we can see how opinions on the social media like twitter could be used for predictions on future events such as election. Using Twitter Scrapper library dataset is extracted from Twitter. TextBlob, Geopy and python are used to extract the sentiment or views of people from the tweets who are likely to vote to a candidate in the general election and a prediction is made on which candidate has more support in a state. The final results of the candidate who has more positive or a negative tweets in a state is shown on the USA map using Plotly Choropleth map.

**REFERENCES**

**[1]** Ramteke, Jyoti, Samarth A. Shah, Darshan Godhia and Aadil Shaikh. “Election result prediction using Twitter sentiment analysis.” *2016 International Conference on Inventive Computation Technologies (ICICT)* 1 (2016): 1-5.

**[2]** Nausheen, Farha and Sayyada Hajera Begum. “Sentiment analysis to predict election results using Python.” *2018 2nd International Conference on Inventive Systems and Control (ICISC)* (2018): 1259-1262.

**[3]** <https://github.com/taspinar/twitterscraper> (Twitter Scrapper)

**[4]** <https://plot.ly/python/> (Plotly)

**[5]** <https://textblob.readthedocs.io/en/dev/> (TextBlob)