2020 Presidential Election Sentiment Analysis

Project Proposal

10/14/20
IST 718: Big Data Analytics
Prof. Willard Williamson
Fall 2020

Group 9:
Jenny Cao
Prathamesh Datar
Ryan Ondocin
Shripad Laddha

Objective:

The aim of this project is to perform sentiment analysis for the 2020 US Presidential election for candidates Joe Biden and Donald Trump. Additionally, we aim to derive main issues via LDA Topic Modeling. We will attempt to give social media users a new vantage point for interpreting the sentimental pulse of the election by topically clustering tweets via the use of a variety of NLP and ML techniques in pyspark.

Data Set Description:

We are planning to use 2 datasets; one containing tweets used for training our sentiment analysis model while the other corresponds to a testing dataset containing specific tweets regarding the presidential candidates we are trying to derive sentimental predictions from.

Twitter's standard API allows users to retrieve tweets within the last Seven days and is limited to scraping 18,000 tweets within every 15 minutes. The primary data set contains 97,200 rows and 7 columns which are as follows: **user, id, location, date, favourites counts, text (truncated = False) and retweet (count)**.

- **Sample Predictors**

Vocabulary generated by NLTK to predict the sentiment of each tweet

- **Link for the data:**

https://drive.google.com/drive/folders/1TULUVu80hbnMyfPT81rXsAjg8sEuK4Jh?usp=sharing

- **Initial Impressions of the data:**
    - Locational data was inconsistently inputted by twitter users. (Syracuse, NY, | CUSE, NYC | Manhattan, NY)
    - Retweets seemed to constitute a large portion of our dataset meaning we will need to be prudent in how we vectorize our data to avoid words with high frequencies carrying disproportionate weight in our vocabulary.
    - Tweets were rife with Emojis, hashtags, and hyperlinks that demand us to diverge from conventional NLP techniques to effectively clean, tokenize and vectorize our data. Relying solely on metadata (time/retweets/location) would deduct from the value of the unique approach we are attempting to pursue. This philosophy allows us to fully leverage NLP techniques that can be integrated into a semi-supervised learning algorithm to gauge public sentiment pertaining to a variety of topics and candidates for this election cycle.

Preliminary Data Exploration:

Locational EDA: Visualize the distribution of supporters of each candidate (derived from sentiment label) based on the locational data we have.

Textual EDA: After using the NLTK package to provide sentiment labels for our training data, we can obtain frequency distributions for the overall amount of positive vs. negative tweets. This may give us a general idea of how many people support each candidate after we figure out a means to subset the data based on hashtags (#Trump2020 vs. #Biden2020). Also, we have tokenized the text to see word counts along with their frequency. For example, from the unique words plot pictured below in Figure 1, we can see the relative length of unique words in popular retweeted election data. The word frequency plot, shown in Figure 2, reveals that 'trump', and 'donald' are the most frequently occurring words in election tweets, suggesting that we may need to conduct bigram analysis on our data.

Figure 1. The Density unique words within tweets

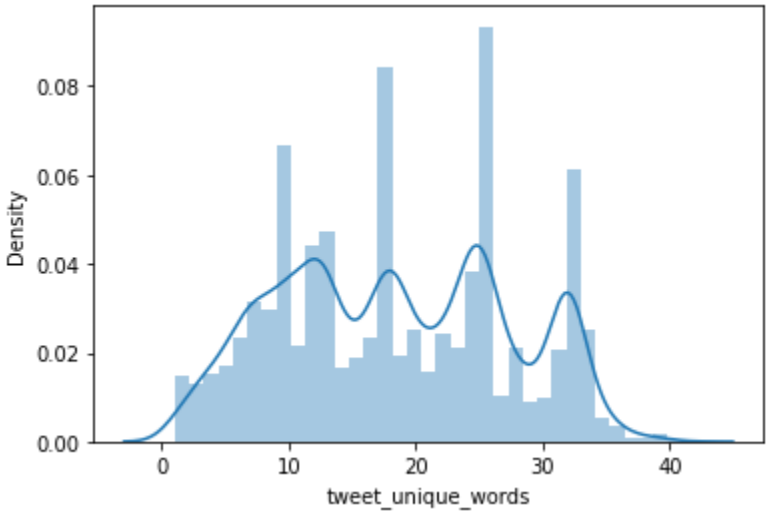
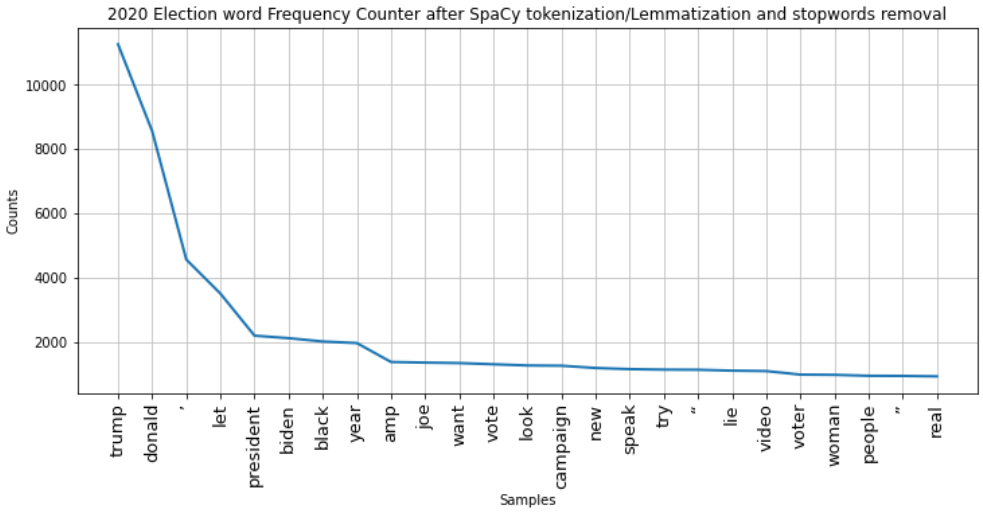


Figure 2. Word Frequency



Retweeted Data: This column represents if a tweet was reposted. It is of a boolean data type and will help us understand the most commonly spread information on twitter regarding the election.

Predictions:

Predicting sentiment that is positive or negative about election candidates / LDA derived topics.

Inference:

Every word in our vocabulary would be considered as a useful variable in predicting sentiment. Therefore, we are interested in finding words that have high predictive power for detecting positive or a negative sentiment.

Non Spark Package:

- **NLTK**: To work with corpora, categorizing text, analyzing linguistic structure, classifying and more by using libraries such as HashingTF, IDF, CountVectorizer, tokenization, tagging, parsing, etc.
- **Plotly**: To visualize the data. Plotly is an interactive and open-source plotting package, so when hovering over the plot, it is easy to see the details of what each bar/line represents and their statistical information. We also can plot animations through plotly, that can help us observe the changes of data over time more clearly.
- **Matplotlib/Seaborn**: To visualize the data for the creation of word clouds, displaying feature distributions, etc.