Aamer Khan
MSDS 453
12/8/18

A.4 - Final Project: Sentiment Analysis on Donald Trump's Tweets

For this project, I chose to pull my data from @realDonaldTrump's twitter account to perform sentiment analysis on. I picked this because the topic intrigued me and it involves sentiment analysis and requires a different form of text classification. I wanted to pull my knowledge from this quarter and challenge myself to use real live data to see if I can make sense of it. In this particular project, I will be using a naïve Bayes classifier that will allow me to organize the tweets from @realDonaldTrump's twitter account and somehow classify which tweets express a positive, negative, or neutral sentiment. One additional thing that I wanted to explore within the data is to find patterns or trends to the president's tweets that could be a possible explanation to the timing, frequency, and negativity of his Twitter habits. I realize that this topic can be controversial to some, but that's precisely why I wanted to do this. I tried to take the politics out of something and create an algorithm that would allow me to analyze the raw data truly. I hypothesize that president Trump's tweets will be overwhelmingly negative and my goal is to create an algorithm that will detect if this is true.

I realized to achieve these ambitious goals I needed data to get started. To retrieve the live data from the president's Twitter account, I needed to create a twitter development account. Next, I had to create a twitter application, which provided me with API keys which allowed me to retrieve tweets from twitter's server. I then created a python notebook, and within that notebook, I imported some essential libraries alongside specific libraries such as Tweetpy. Tweetpy allowed me to interface with twitters server so that I was able to pull from a detailed twitter account, in this case, @realDonaldTrump. The reason why I'm not using a precompiled dataset is that I wanted to challenge myself and preserve authenticity to achieve my goals. I also wanted to gather the amount of data I thought was necessary to accomplish this task and not have to rely on what others may have extracted from the president's Twitter account.

The next step was to create a data frame using the Pandas library. After placing the first 200 tweets in the data frame, I wanted to see what features were available per individual tweet. I was able to determine that many features can be extracted in each tweet. Some of the features that I chose in the data frame were when the tweet was created, the source of the tweet, the favorite count, and retweet count. I then cleaned the text in the tweet by removing all links and special characters using regular expression in python. Unlike some of the visualizations that we encountered in class, such as clustering, my visualizations took place after the sentiment analysis.

With the tweets prepped and ready to go, the next steps were to perform the sentiment analysis by creating the classifier. I created a utility function that classifies the polarity of a tweet using a python library using Textblob, which is based on the python library NLTK. The Textblob sentiments module provides two sentiment analysis implementations, pattern analyzer that's based on the pattern library, and lastly naïve Bayes analyzer for my modeling method, which is an NLTK classifier trained on a movie reviews corpus. Textblob is using a pre-trained model to perform the sentiment analysis. In this case, rather than creating a model from scratch for my research method, we will merely utilize the pre-trained model to obtain our results.

When you run the clean tweets through the classifier, the result is a numeric value which represents the polarity of the tweet. I wrote some conditional statements that translate those polarity results as either 1, 0, or -1 which equate to positive, neutral, or negative sentiment. The results of running 200 tweets through the classifier will be stored in a new column called SA short for sentiment analysis. The reason why I took the results and put it back in the data frame is so that I can perform various analysis from those results. We can now quickly determine what percentage of those 200 tweets from my dataset are now positive, negative, or neutral. Also, we can utilize the results of the sentiment analysis to create visualizations which allow us to grasp the distribution of positive, negative, and neutral tweets from @realDonaldTrump visually.

The results of the analysis show that 54.5% of the president's tweets are classified as positive, 28% is neutral, and 17.5% is negative. I was also able to determine the sentiment of the president's tweets per day of the week. As it turns out, the president's tweets are the most negative on Wednesday and the least negative on Friday (refer to figure 3). Lastly, I was able to determine the sentiment of President Trump's tweets per the hour of the day, in which he is the most negative around 1 PM and most positive around midnight (refer to figure 4). To evaluate the models, I acknowledge the limitations of the classifier since it was pre-trained on a movie reviews corpus. Thus the sentiment analysis may not be as accurate because we are classifying tweets rather than reviews. This can be problematic because the accuracy of results yields about 73% based on movie reviews being appropriately classified.

Aamer Khan
MSDS 453
12/8/18

Concerning implementation, all this was carried out by using the sentiment analyzer via Textblob and secondly for loops to iterate through the results of the sentiment analysis to determine the percentages of positive, negative, and neutral tweets. Lastly, I utilized Matplotlib to create four visualizations ranging from pie charts and stacked histograms. I had to develop various columns via conditional formatting to get the day of the week and hour of the day per individual tweet. Those results were then stored in newly created columns in the existing data frame.

In hindsight, if I could have made this better, I would have implemented algorithms other than naïve based such as linear SEM or logistic regression. These algorithms could have been beneficial because they are simple, easy to use, and efficient on a large number of features. We could also account for inherent biases attached with looking at how many times words occur in the text. In particular, the longer the text, the longer its features/word count will be. To fix this issue, we can implement word frequency rather than using word count followed by dividing the number of occurrences by the sequence length. Besides, we can downscale these frequencies so that words the occur numerous times, such as stop words, have lower values. In other words, I'd like to utilize IDF or inverse document frequency. Putting these two concepts together is called TFIDF which is what we had used in class. I could have also done a better job at pre-processing the tweets, which is crucial but does not need to be overly complicated. However, the better the pre-processing, the better the classifier can do at its job.

In the end, my original hypothesis of the president's tweets being overwhelmingly negative turned out to be incorrect. Based on the fundamental sentiment analysis that I performed on his 200 tweets, we now know that this is not true. In reality, 54.5% of his tweets are positive. This begs the question as to why that is. One possible explanation is the fact that the President has a habit of having a lot of self-praise and also frequently taking adverse situations and turning them into positive ones. By putting a positive spin or bias on his tweets, the classifier determined that the tweets were positive when in reality they might have been negative. We need to apply our outward knowledge of politics on top of the data to make sense of it. In this case, you would need to know a little bit about the source of our data, @realDonaldTrump, to make sense of the results of the sentiment analysis.

Aamer Khan
MSDS 453
12/8/18

Appendix: Figures 1-4

Figure 1:
Percentages of Sentiment of Trump's Tweets

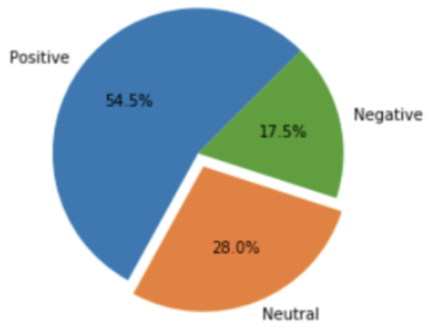


Figure 2:
Percentages of Trump's Tweets from iPhone vs. Non-Mobile Device

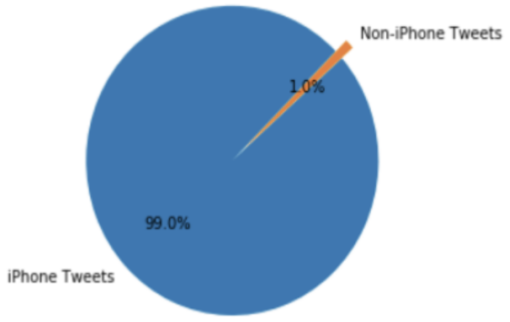


Figure 3:
Stacked Plot of Sentiment of Trump's Tweets per day of the week

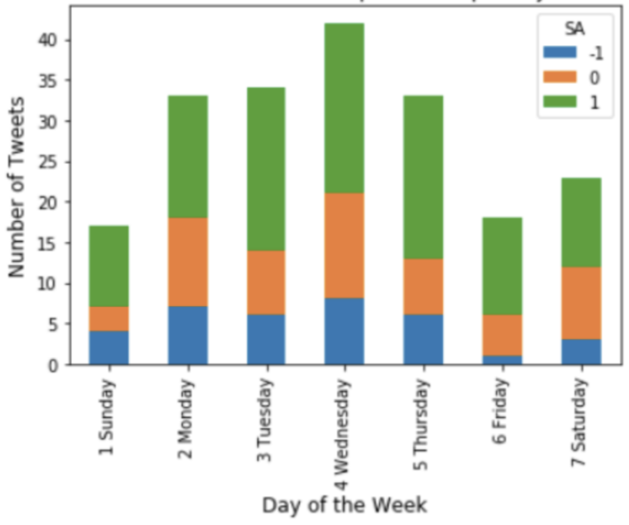
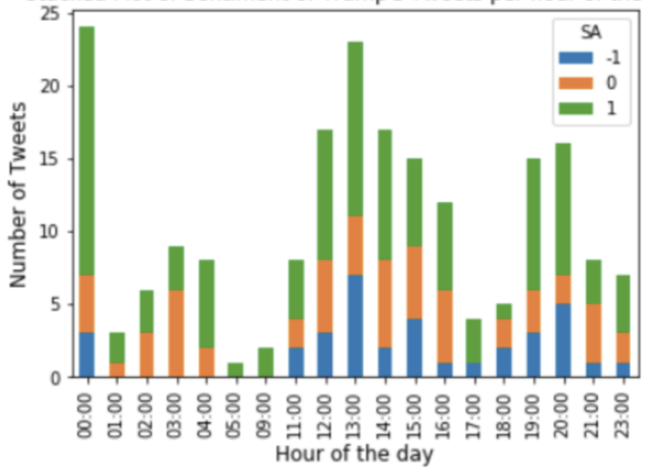


Figure 4:
Stacked Plot of Sentiment of Trump's Tweets per hour of the day


SA
Number of Tweets
00:00
01:00
02:00
03:00
04:00
05:00
09:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
23:00
Hour of the day