Han Ambrose, Pranav Shah, Soren Nelson

Data Mining CS 6140

**Intermediate Report**

1. **What progress have you made towards your proposed goal?**

We collected popular tweets during the Democratic debates on February 7th and February 19th as well as each candidate's tweets. We then tried multiple methods to gain insights into our data. How we gathered our data and our methods used to inspect the data is explained in detail below.

1. **If you tried some basic approaches: what worked well and what did not?**

Running our algorithms prior to preprocessing did not work well. We also tried running k-means on the words and not using features like polarity/subjectivity, this did not give us anything meaningful. 2-grams gave much more context than 1-grams and thus proved better. There was too much noise in the tweets to gather that much from similarity between tweets. For the polarity methods, approach “a” worked well than approach “b”. For the per candidate polarity and subjectivity, Lloyds algorithm seemed to work well. It would be interesting to use other features as opposed to polarity/subjectivity.

1. **What could be done to improve the basic approaches?**

For our polarity methods, approach “b” did not work well because the size of wordlist used to calculate polarity was very small. We are still trying to find a better wordlist/corpus of positive and negative words to improve the working of approach “b.” Or else, we will stick to approach “a” and use data mining methods over the results of approach “a” to draw better conclusions. We could try a different clustering algorithm other than Lloyds or use various other features. Using what we have currently side by side with data from the transcripts will improve our insights as well

1. **What experiments have you run and are you planning to run to demonstrate the effectiveness?**

We will look at how our analysis of candidates compares with how well they do in the primaries. Maybe more positive candidates do better or something along these lines. We will continue to try to draw extra insights with different algorithms such as LSH, other clustering methods, and regression techniques.

**Data Collection and Preprocessing:**

We started our data collection by getting the most popular political hashtags. Using Tweepy, we gathered 48 for first and 89 hashtags for the second debate. We then used these hashtags as search terms to get all tweets using these hashtags with more than 5 favorites. We used regular expressions to preprocess the data and removed all the stop words, punctuation marks and the words which have length less than 3.

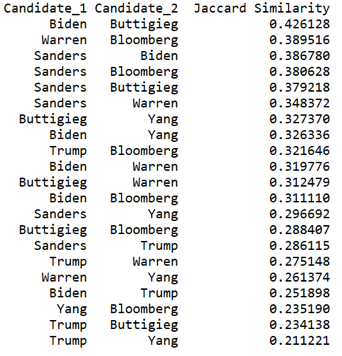
**Frequency:**

The first step to understand an idea about the data is to extract the frequency of important words used. This can be done using bag of words with n- grams. While frequent words represent a clear popular topic, one term frequencies do not give us a deep explanation of what the context is about, which leads us to the next step of trying 2-gram words. We can see that words make more sense when they are together.

[(('three', 'houses'), 15), (('even', 'Democrat'), 12), (('working', 'class'), 12), (('Democratic', 'Party'), 12), (('Super', 'Tuesday'), 10), (('beat', 'Trump'), 10), (('socialist', 'country'), 10), (('Trump', 'hes'), 10), (('medical', 'records'), 9), (('I', 'know'), 9), (('know', 'Trump'), 9), (('hes', 'great'), 9), (('great', 'guy'), 9), (('guy', 'Im'), 9), (('Im', 'big'), 9), (('big', 'fan'), 9), (('fan', 'Trump'), 9), (('presidential', 'campaign'), 9), (('3', 'houses'), 9), (('goes', 'convention'), 9), (('2', '3'), 8), (('Chuck', 'Todd'), 8), (('best', 'known'), 8), (('known', 'socialist'), 8), (('millionaire', 'three'), 8), (('convention', 'votes'), 8), (('votes', 'receive'), 8), (('receive', 'nomination'), 8), (('nomination', 'support'), 8), (('support', 'nominee'), 8), (('3', '4'), 7), (('Hillary', 'Clinton'), 7), (('every', 'single'), 7), (('Steve', 'Scalise'), 7), (('guy', 'helped'), 7), (('happens', 'millionaire'), 7), (('2016', 'presidential'), 7), (('radical', 'idea'), 6), (('health', 'care'), 6), (('Democratic', 'Presidential'), 6), (('lose', 'Trump'), 6), (('gun', 'control'), 6), (('cant', 'think'), 6), (('think', 'bro'), 6), (('Scalise', 'CRUSHES'), 6), (('heart', 'attack'), 6), (('4', 'years'), 6), (('look', 'like'), 6), (('1', '2'), 6), (('Democratic', 'nominee'), 6)]

**Measuring Similarity of tweets between pairs of candidates:**

Jaccard Similarity is used to find similarity between pairs of candidates. We break it into n-grams, in this case, we tried 3 grams of characters. By doing pairwise comparison, we receive the results below:



It is difficult to measure similarity among tweets as there are a lot of noise and we receive a low percentage of similarity. However, overall, the results make sense in terms of political views among tweets related to the candidates. There are more similarities among democratic candidates versus Trump is in the lower end of similarity. People who support Yang generally do not support Trump and vice versa. Buttigieg and Biden’s similarity is quite higher than other pairs.

**Tweet Polarity Approaches**

We used two approaches to estimate the polarity of a tweet text which can be later used to draw more conclusions from the tweets. The two approaches which used are as follows:

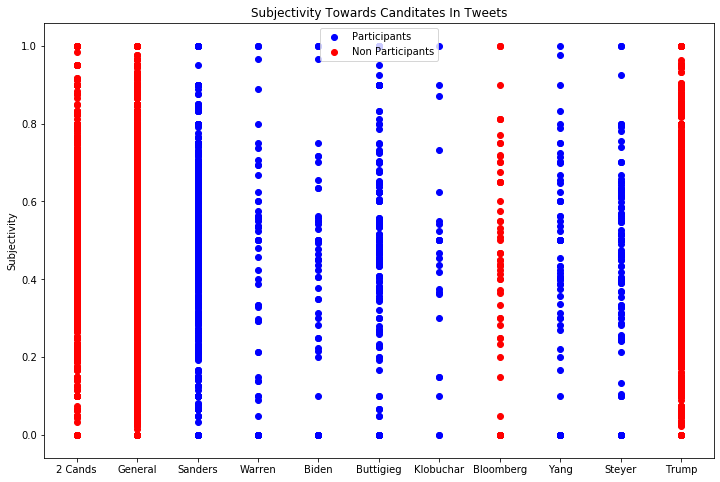
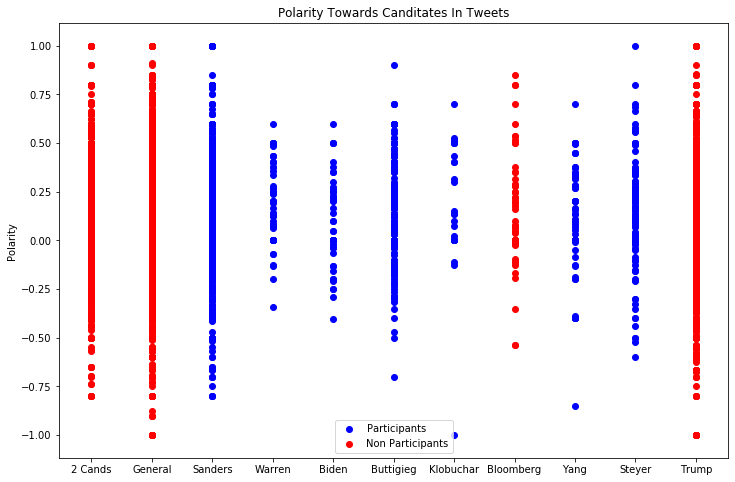
1. Using nltk to calculate polarity: In this approach we used NLTK library to calculate polarity of each tweet text and used matplotlib to visualize it. This seemed to be too straight forward approach, so we tried method b.
2. Using a corpus of positive and negative words: After pre-processing the data, all the words in a text are stemmed using Porter Stemmer. Later, these words are checked if they belong to the positive wordlist or negative wordlist and then the polarity of each tweet text is calculated.

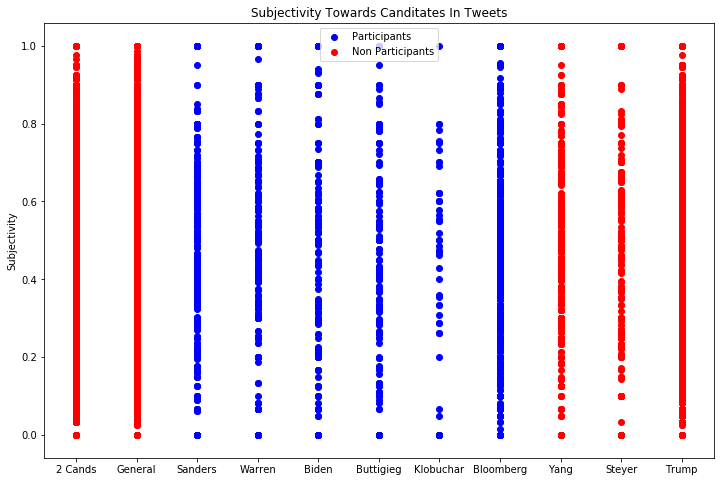
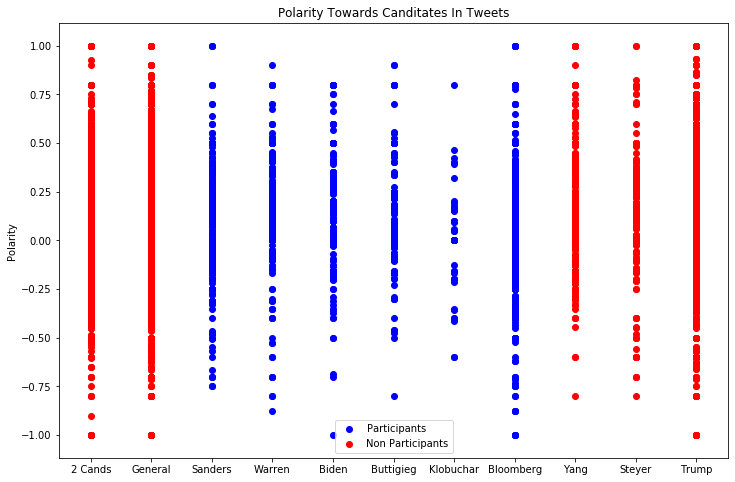
**Polarity and Subjectivity Per Candidate**

We look at what people are saying on twitter about a specific candidate and if that was correlated to what the candidate said during the debates. To start we find the subject of a given tweet we checked whether a tweet used a single candidate's name. We separated out all tweets that referred to more than one candidate and any tweet that didn’t mention a candidate. This left us with the following distributions:

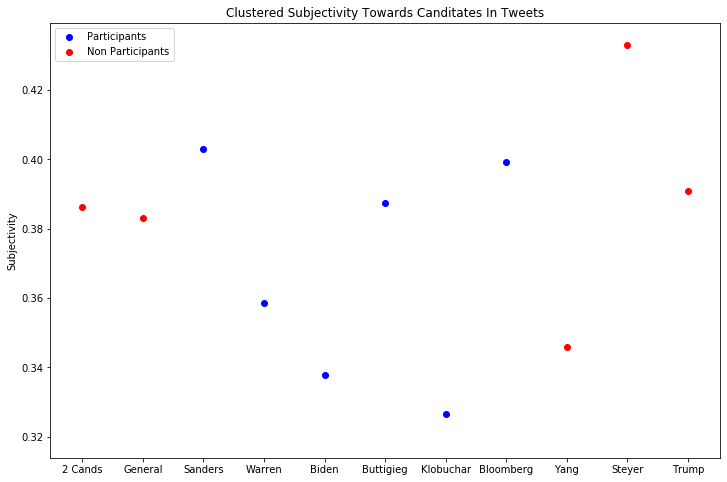
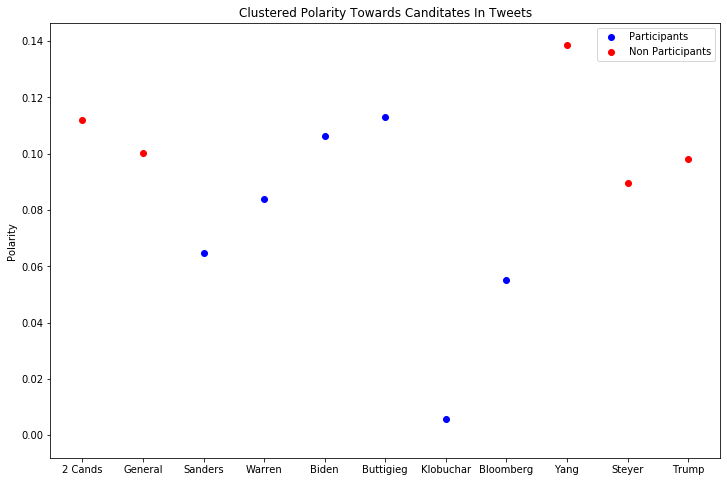
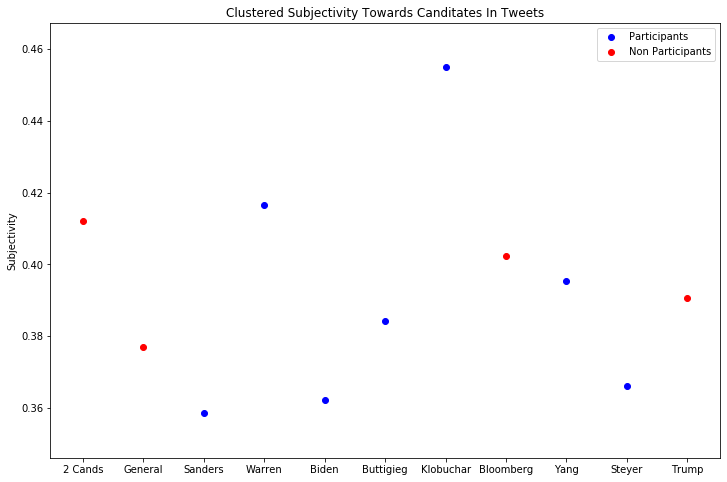
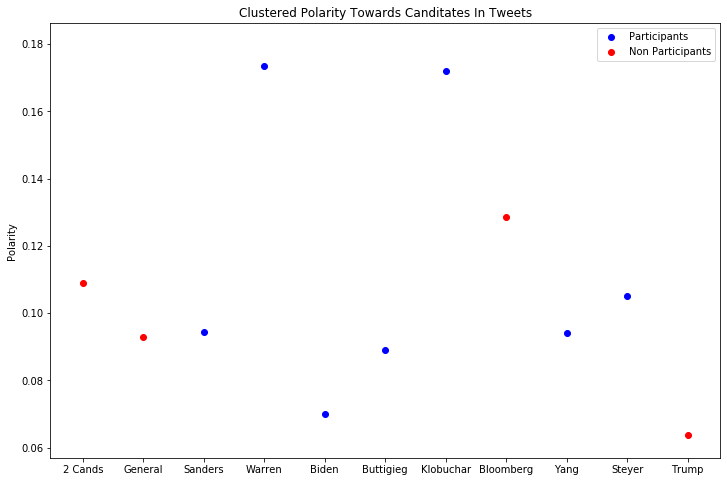
* First Debate - Two Candidates: 1326, No candidates: 5600, Sanders: 978, Warren: 60, Biden: 60, Buttigieg: 226, Klobuchar: 26, Bloomberg: 67, Yang: 79, Steyer: 152, Trump: 2211
* Second Debate - Two Candidates: 2966, No Candidates: 7283, Sanders: 405, Warren: 315, Biden: 272, Buttigieg: 162, Klobuchar: 67, Bloomberg: 803, Yang: 489, Steyer: 201, Trump: 2689

President Trump had the most tweets about him even though he was not participating in either debate. Interestingly, both Yang and Steyer had more tweets about them in the second debate even though they both only participated in the first. We then gathered the subjectivity and polarity. The following are plots from the first debate and second debate respectively:

****



The majority of the candidates lean slightly more positive and more objective than subjective. We felt this wasn’t very informative and difficult to gather much data so using this data we ran Lloyd’s Algorithm to cluster the data. We initialized a cluster at 0.0 for each candidate on the polarity data and 0.5 on the subjectivity data. After clustering we get the following plots:



From this we see that all candidates are talked about more positively than negatively. We can also see who did well and who didn’t in each debate. After gathering data for another debate we intend on plotting each of these debates together to further extrapolate trends. We will also be analyzing the debate transcripts in the same manner which may lead to more insights.