**Intermediate Report**

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

As describing below, I have preprocessed tweets and run it through bag of words to find frequency of words and also Jaccard similarity among pairs of candidates

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

What did not work well is trying to run it through the model without preprocessing

After removing punctuations and stops words the results make a lot more sense

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

We could try sentiment analysis to identify positive and negative words or calculate how close a word is with terms like good and bad.

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

I have tried both bags of words and k-means. However, k-means doesn’t return any meaning full context because tweets has so much noise. Next step is to try out k-means in terms of positive/negative features and not based on words. We could also try LSH

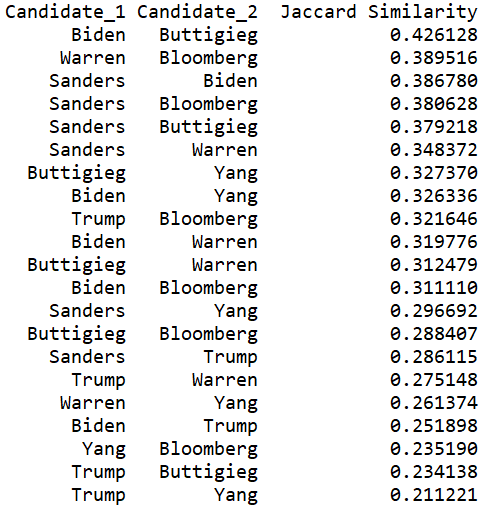
The below data is collected after the Democratic Debate happened in February 20th, 2020

**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:**

By using Jaccard Similarity is used to find similarity between pairs of candidates. We break it into n-grams, in this case, I 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 is generally do not support Trump and vice versa. Buttigieg and Biden’s similarity is quite higher than other pairs.