Using a Naive Bayes Classifier to Predict 2020 Democratic Candidates Based on Question Responses

Aashish Ghimire, Chase Mortensen

Introduction

The purpose of this project was to apply a Naive Bayes classifier to a dataset containing information on the 2020 DNC debates. We often associate certain candidates with a phrase or words, and in this project to quantify if that is statistically significant and we can tell candidates based on part of their speech. This project could be useful for politicians seeking to align or distinguish themselves from other popular or notable politicians. We hoped to predict which candidate had given a specific response based on other responses. Additionally, we were interested in applying a cross-validation technique and divide the dataset based on debates to see which debates were most predictable and which were most atypical. Slides are here and the GitHub repository of the code can be found here.

Dataset

The dataset used in the project is the transcript of the democratic primary debate for 2020 presidential candidates. This is a political dataset, and mostly have qualitative values rather than quantitative. The dataset used for this class is obtained from Kaggle. It can be found here and it has been updated to include the most recent debate within a few days of debate. One aspect of this data is that all speeches are a response to a question by debate host or moderators - so they tend to be similar across candidates. The dataset was retrieved as a CSV file, and some cleaning was done to ensure that punctuations, cases, and formatting do not hugely influence the results. We used the most frequent words in the English language (known as stopwords in nltk package) to ignore them.

We also created a second small dataset to make predictions on that included responses from previous election cycles. The dataset includes 40 debate responses: 10 each from Donald Trump, Hillary Clinton, Barack Obama, and Mitt Romney.

Analysis technique

We implemented a Naive Bayes classifier model from the nltk package to analyze textual input from the 2020 election cycle DNC debates. This classifier was trained on all of the words spoken in each of the debates and would make predictions for a response by parsing the response into individual words, removing common words, and making predictions on the remaining words for the candidate most likely to have said each word. The response was then classified as the candidate that was classified most for each of the words within the response.

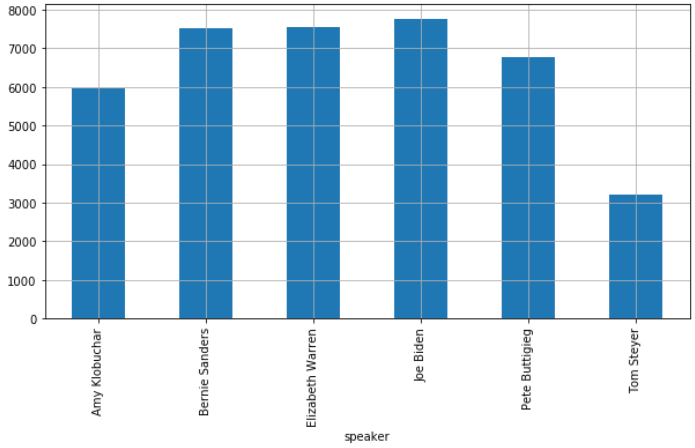
The Naive Bayes classifier allowed us to train and predict each of the words and taking into account the proportion of words each candidate spoke (for example, Steyer was predicted less because he had fewer words overall).

We tested the classifier using a modified k-folds cross-validation technique where data from each of the debates was used as a fold and the model was trained on the other debates. This allowed us to answer one of our initial questions of which debates were typical and which were atypical.

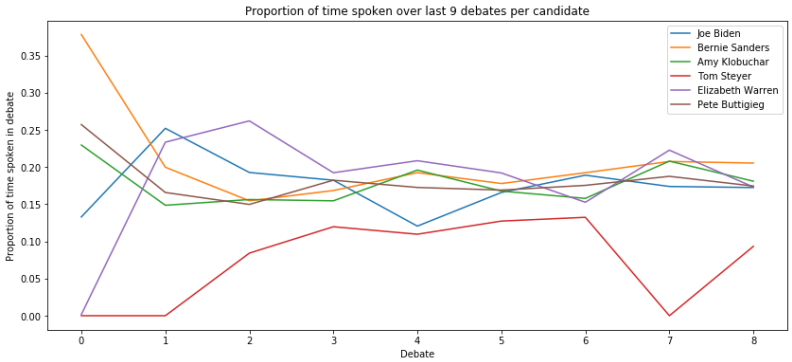
We also used a few other approaches - like using phrases (defined as two or three consecutive words) and taking that pair or tuple and using that to classify.

Results

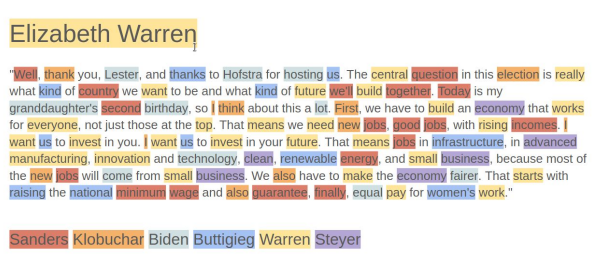
First, we looked at total speaking time, which gives us an indication of why Biden was predicted more often and candidates like Steyer were predicted less often. The graph to the right shows speaking time totals across all debates. The graph below shows the proportion of candidate speaking time across debates. The first debate was removed because there was missing data and the second debate was split into two nights and then recombined our analysis, which may explain the high variation in debate 0 below.

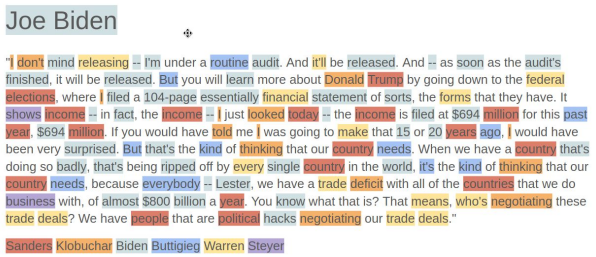


for



Besides predicting responses from candidates in this election cycle, we also tested responses from previous election cycles. Our model was biased toward Joe Biden since he was the most frequent speaker. We tested 10 responses each from Trump, Obama, Clinton, and Romney. Our model predicted Joe Biden on all responses but five: four Clinton responses and one Obama response were attributed to Elizabeth Warren. Here are two examples - one where a Clinton response was classified as Elizabeth Warren and another where a trump response was classified as Joe Biden.





While these response classifications are interesting, our classifier is predicting who is most likely to have said it *in a 2020 DNC debate* - not whose speech is most similar overall. Again, the model is more likely to predict Biden based on the amount of time/number of words he spoke overall. This could be improved by training on an equal number of words for each candidate.

Debate	Accuracy
2-25-2020	0.851
2-19-2020	0.920
2-7-2020 (150)	0.840
1-14-2020	0.775
12-29-2019	0.840
11-20-2019	0.865
10-15-2019	0.915
9-12-2019 (116)	0.966
7-31-2019 (41*)	1.0
7-30-2019	0.942
6-27-2019	0.921
6-26-2019 (34)	0.853
ALL	**0.879**
ALL (without discarding)	**0.716**

The table to the left shows our accuracies for the different debates when using cross-validation across the different debates after discarding responses of length greater than 30. The 7-31-2019 debate had perfect accuracy - however, this debate was the second night of a two-night debate that split the candidates up. Joe Biden was the only 'top candidate' (that we chose to analyze) that participated in the second night. There were also spikes in prediction accuracy for debates where Tom Steyer is not present, like 2-19-2020 and 9-12-2019. If we remove 7-31-2019 from consideration, 9-12-2019 was the most predictable debate - meaning candidates' answers from other debates resulted in a model that was highly accurate at predicting candidates in this debate. We discarded short responses because these responses weren't very interesting in general. Before removing these responses, our accuracy was 0.716. After removing them, accuracy improved to 0.879. We tested varying lengths and 30 words seemed to provide a good result - discarding more didn't improve the model more than a few percentage points.

The alternative approach - using a pair or tuple: We tried using two or three consecutive words for predicting the candidates. We observed that because of a relatively smaller dataset, this will produce a new and unique phrase during testing that is not present in the training dataset. As a result, the model tends to predict the candidates with the highest speech length. This approach of taking phrases hugely reduces false positive (because of less ambiguity) but at the cost of accuracy. While we did much better than a random choice, the overall accuracy was much less than using a single word. The result of this approach is listed in the table here.

len(phrase) \ test	by phrase	by answer
two words	0.63	0.26
three words	0.83	0.23

The Biden bias compounds when we take the whole answer and decrease the accuracy. While we did not have enough time to implement a hybrid of both approaches in this project, the use of both word-wise and a phase-wise classification would be able to achieve better accuracy and precision working together.

From this project, we could clearly see that each candidate has a signature word that they tend to use more and we can guess them based on a word or small speech with relatively high accuracy. This is evident even in a moderated debate where everyone is talking on a same topic. Similarly, we can use the same classification method to see the similarity between two candidates or find whose speeches are the most similar.