Final Project

IST 736 – Text Mining

Drew Howell & Nash Delcamp

2019-06-14

Table of Contents

[Introduction 2](#_Toc11446224)

[Analysis and Models 2](#_Toc11446225)

[About the Data 2](#_Toc11446226)

[Models 4](#_Toc11446227)

[Methods 4](#_Toc11446228)

TF-IDF Vectorizer 6

K-means Clustering 6

Multinomial Naïve Bayes 7

Support Vector Machine 7

Neural Networks 8

Random Forests 8

Cross Validation 8

Model Evaluation 9

K-means Clustering 9

Multinomial Naïve Bayes 10

Support Vector Machine 10

Neural Network 10

Random Forest 10

Results 10

K-means 10

Political Party 11

Term Length 12

TIme Period 13

[Conclusions 1](#_Toc11446247)4

[References 1](#_Toc11446248)6

# Introduction

Many of the most regarded speeches in world history were delivered by a United States president. Reagan telling Gorbachev to tear down the Iron Curtain; Kennedy instructing us to ask how we may serve rather than how we may be served; every formal address to the public delivered by a US president is grammatically, syntactically, and phonetically engineered to an exhaustive degree with the end goal being to resonate with the target audience: us. It is an inarguably worthy cause, then, to use the skills gained in this course to “reverse-engineer” these speeches. Perhaps we can derive patterns within and between presidential speeches such that we may learn something new about our leaders, past and present. Such is the goal of the following analysis. We approached this project by starting with the available data and allowing exploratory analysis to define our most fruitful avenues for statistical modeling, and we did learn something new – several things, in fact.

# Analysis and Models

## About the Data

A total of 122 presidential speeches were scraped from The Miller Center – Famous Presidential Speches7. The top three ranked speeches were selected, however some presidents were not in office long enough to give three speeches. In these cases, all available presidential speeches were selected. In addition to the raw text of the speeches, demographic information for each president was scraped. These demographics include date of birth, date of death, education, careers, marriages, political parties, inauguration date, and final date of presidency. Speeches from all former presidents besides George Washington and John Adams – the first two presidents – as well as the current president were included. The time of the speeches ranges from March 1801 – January 2018.

All non-ascii characters were removed from each speech, then stopwords from the NLTK English stopwords were removed. All HTML headings and tags were then removed. Each speech was then vectorized using a unigram TF-IDF vectorizer. The tokens with the greatest TF-IDF value overall are presented below. These tokens include locations – Panama, Japanese, Greece, Iraq; economical forces – bank, harvest, coalition; and big actions – dedicated, advice, rebellion, and applause.

Figure 1 displays the tokens with the greatest TF-IDF values



Additional features were calculated based on the demographic data for each president. These variables include age at inauguration, age at death, years in office, and indicator variables for popular careers and political parties. Summary statistics for scraped and calculated data are presented below. Some particularly interesting features from this summary include 55% of all presidents were at one time a lawyer while only 17% were in the military, 45% of presidents were identified as Republicans while 38% were identified as Democrats. The average number of years in office was about 5 but was as low as 0 years and as high as 12 years – only once when Franklin Delano Roosevelt was elected to three terms.

Table 1 presents summary statistics of demographic data of all presidents

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Age  Inaugurated** | **Age Death** | **Years In  Office** | **Lawyer** | **Military** | **Teacher or  Professor** | **Republican** | **Democrat** |
| **count** | 42 | 37 | 41 | 42 | 42 | 42 | 42 | 42 |
| **mean** | 54.810 | 70.622 | 4.927 | 0.548 | 0.167 | 0.095 | 0.452 | 0.381 |
| **std** | 6.758 | 12.544 | 2.678 | 0.504 | 0.377 | 0.297 | 0.504 | 0.492 |
| **min** | 42 | 46 | 0 | 0 | 0 | 0 | 0 | 0 |
| **25%** | 51 | 63 | 3 | 0 | 0 | 0 | 0 | 0 |
| **50%** | 54 | 70 | 4 | 1 | 0 | 0 | 0 | 0 |
| **75%** | 57.75 | 79 | 8 | 1 | 0 | 0 | 1 | 1 |
| **max** | 70 | 94 | 12 | 1 | 1 | 1 | 1 | 1 |

All speeches were also given a sentiment score using Vader Sentiment. The sentiment of each speech is valuable because it could reflect the type of speaker a president is or the circumstances surrounding the speech. A normalized time series plot is presented below which illustrates the overall increase of both positive and negative sentiment over time. The positive sentiment ranges from 0.34 – 0.20, while the negative sentiment ranges from 0.15 – 0.08.

Figure 2 illustrates the positive upward trend in both positive and negative sentiment over time



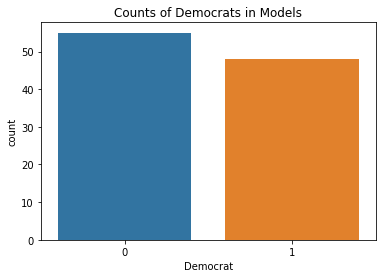
## Models

### Methods

Models were built using K-Means Clustering, Multinomial Naïve Bayes (MNB), Support Vector Machines (SVM), Neural Networks, and Random Forests. The predictive models were used to model political party, term length, and time period. All train/test splits were stratified so each class was distributed properly across the train and test sets. All cross validation was performed using the train set, then the best model was selected, then that model was used on all training data to predict results in the test set.

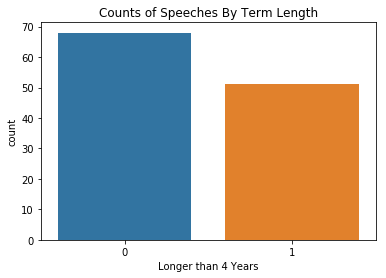
Political party was modeled using only presidents who were identified as either Democrat or Republican. From a historical standpoint, this proposes a difficult challenge: the Republican party today would be unrecognizable to a Republican in the 1800s, such as Abraham Lincoln. The focus of both parties has changed drastically over time. The distribution of political party used in modeling is presented below.

Figure 3 depicts the distribution of pollical party used in modeling



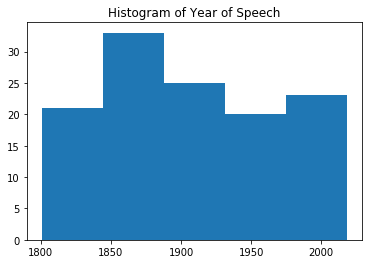
Term length was binned into two categories: Years in Office <= 4 years or Years in Office > 4 years. The current president, Donald Trump, was removed from the training and testing data since his term length is still unknown. Predictions of Trump’s term length were made based off each of his speeches. One difficult challenge in modeling term length is the fact that many presidents met untimely deaths due to assassination or illness, not lack of likeability or proficiency in scandals. These presidents include William Henry Harrison (0.08 years), James A. Garfield (0.54 years), Zachary Taylor (1.33 years), Gerald Ford (2.42 years), Warren G. Harding (2.42 years), Millard Fillmore (2.67 years), and John F. Kennedy (2.83 years)8. The distribution of term length used in modeling is presented below.

Figure 4 depicts the distribution of term length of presidents used in modeling



Time period was binned into five intervals: 1801 – 1844 (era1), 1844 – 1887 (era2), 1887 – 1931 (era3), 1931 – 1974 (era4), and 1974 – 2018 (era5). The goal of these models was to demonstrate the difference in the way presidents spoke during these different time periods. The distribution of the time periods is presented below.

Figure 5 depicts the distribution of time period



#### TF-IDF Vectorizer

All models were built based on TF-IDF vectorized values of each word in each speech. The vectorized values represent the product of the normalized term frequency and the inverse document frequency (idf). The idf is calculated as the logarithm of the number of documents divided by the number of documents that term occurs in. The TF-IDF vectorization therefore penalizes terms that occur in many speeches.

#### K-Means Clustering

K-means is a supervised learning algorithm that is especially useful for discovering natural similarities within a dataset. The algorithm uses a predefined number of groups (K) and equally distributes K centroids throughout the vector space. The observations of the dataset are clustered to the nearest centroid over several iterations and each is assigned a label denoting the cluster to which it belongs.

This is a particularly helpful classification method when it comes to generalizing the different characteristics of the data being analyzed. Such is explored in a later section.

Figure 6 shows an example of a K-means model classifying a dataset with a 2-D vector space

#### ../Desktop/Screen%20Shot%202019-06-14%20at%2010.49.27%20PM.png

#### Multinomial Naïve Bayes

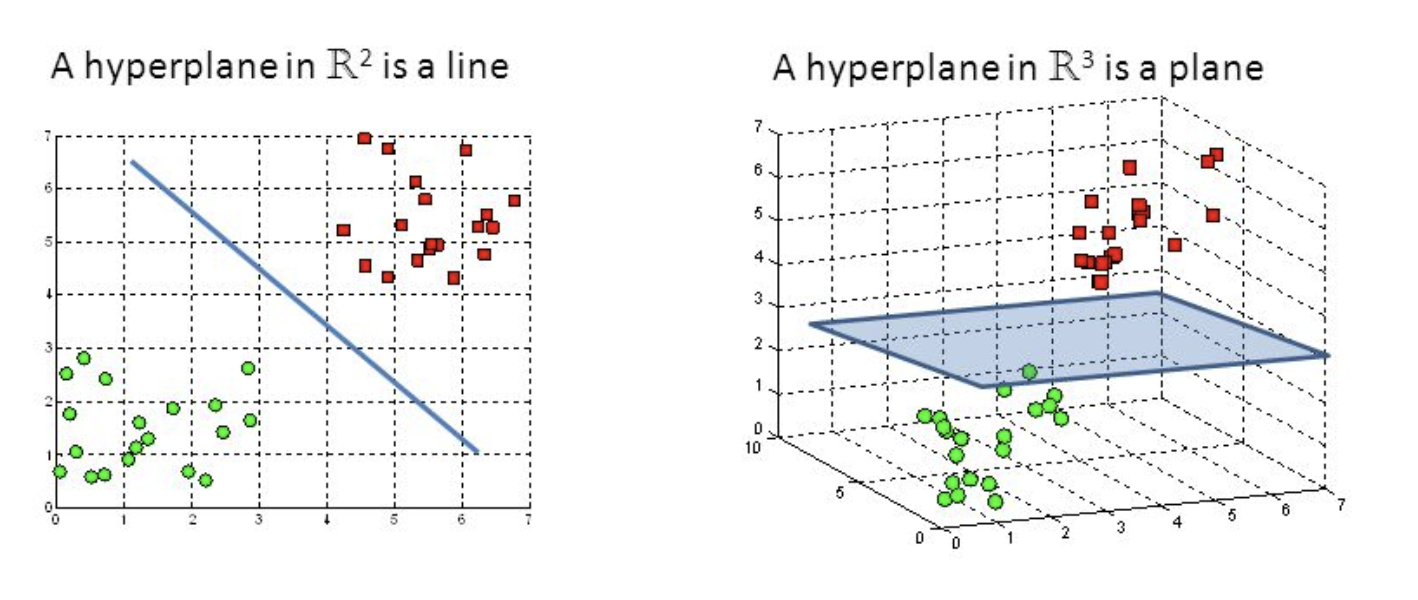
Naïve Bayes is a supervised learning method that requires a labeled response variable, like lie and sentiment. Naïve Bayes models are particularly popular for text classification1. This machine learning classifier utilizes Bayes rule, which states that the probability of Event A given Event B is equal to the probability of Event B given Event A times the probability of Event A divided by the probability of Event B. Or;

The Naïve Bayes classifier makes one particularly well-known assumption: All the occurrences of features are conditionally independent. This is almost always false – naïve – but it performs consistently well in most situations. A requirement of the model is that the predictor data must be nonnegative, so if one were to standardize the values of a vectorizer, the results must all be nonnegative. Naïve Bayes can model both numerical and categorical predictors if the posterior probabilities can be provided.

#### Support Vector Machines

Support Vector Machines can be used for both continuous and classification tasks. An SVM model finds a hyperplane in N-dimensional space to classify each point, where N is the number of features. In the case of the sentiment of movie reviews, there are 5 features. This hyperplane acts as a decision boundary to determine how to classify each point. As shown below, this can be visualized well in 2-dimensional (line) and 3-dimensional (plane) space but becomes difficult to visualize beyond three dimensions2.

Figure 7 illustrates the hyperplane decision boundary in 2 and 3-dimensional space.

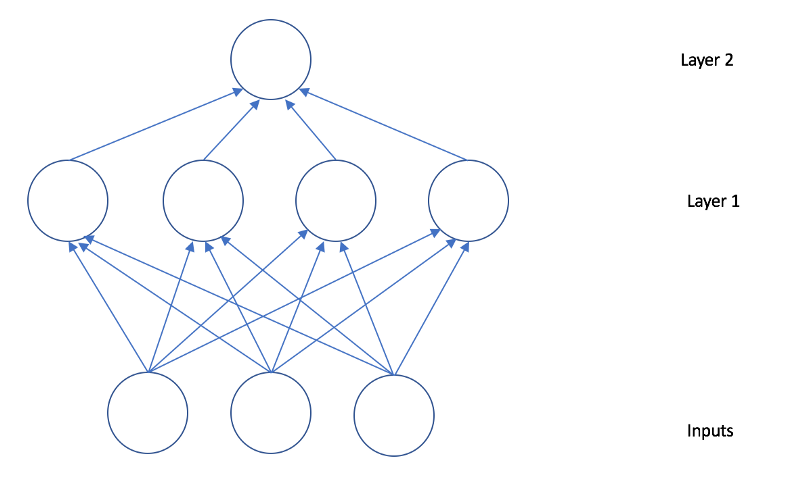


The C parameter in SVM models controls the margin of the hyperplane, inversely. In other words, C controls the penalty of the error term. The linear kernel was used in all models to focus on the value of n-grams and C.

#### Neural Networks

Neural Networks are often described as an algorithm that operates similarly to the human brain. Much like the human brain consists of neurons that communicate with one another, a Neural Network consists of nodes that pass information to one another to learn and make predictions. A Neural Network can consist of many different layers which add to the perplexity of the model. This method also utilizes some number of epochs which modify the weights assigned to each connection. A visualization of a two-layered Neural Network is presented below5.

Figure 8 illustrates an example of a Neural Network5

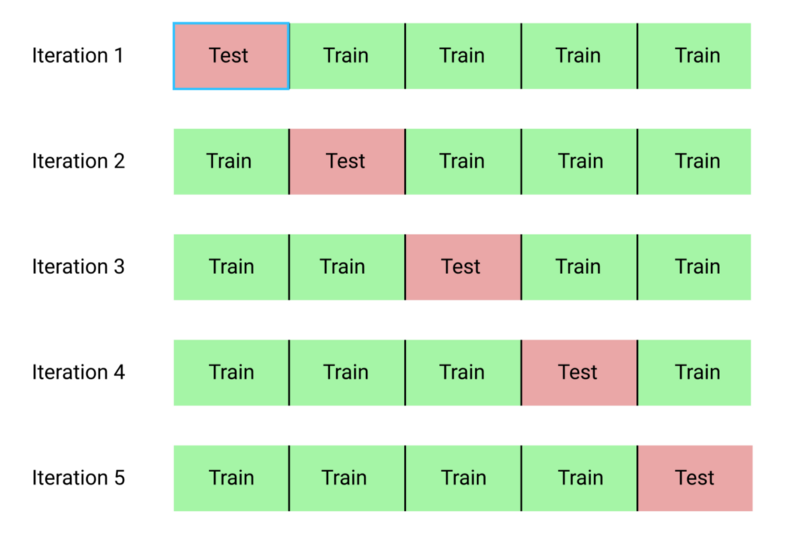


#### Random Forests

The Random Forest method is an ensemble method that creates many random, uncorrelated decision trees to make continuous or discrete predictions. Since decision trees can have high variance depending on training and testing data, using many of these trees improves the accuracy and reliability of this method. The Random Forest method also only uses a subset of all features in each decision tree, making the decision trees uncorrelated6.

#### Cross Validation

MNB, SVM, and Random Forest models were selected using three-fold cross validation. Cross validation is a method of validating a model such that some percentage of the dataset is the test set for one of the folds. In this case, for each of the three folds, the model is trained on 67% of the data then tested on 33% of the data. This is repeated with a different test set three times. The figure below depicts a 5-fold cross validation example. In this example, for each iteration, the model is trained on 80% of the data and 20% of the data are used as the test set.

Figure 9 displays an example of 5-fold cross validation3  


#### Model Evaluation

The models that were not cross validated were evaluated *via* confusion matrices, precision, and recall.

Confusion matrices are a visualization metric that display the number of true positives, true negatives, false positives, and false negatives that a model predicts. These are beneficial for visualizing accuracy as well as potential bias.

Precision4 is often referred to as the positive predictive value which is calculated as the number of true positives divided by the number of false positives; i.e.,

Recall4 is often referred to as sensitivity and can be interpreted as how complete the results are. Recall is calculated as the number of true positives divided by the number of total positives; i.e.,

The macro average of these metrics was also calculated. The macro average is preferable to the macro average since the classes are balanced well.

### K-Means Clustering

The K-means algorithm uses the entire dataset, rather than training on a percentage of the whole and vice versa. To fit the model, it was necessary to use a vectorized sparse matrix as the input. Aside from commanding the number of clusters, default settings on the skLearn KMeans function were used to run the model. The K values were set individually at 2,3, and 4.

### Multinomial Naïve Bayes

All MNB models were built using GridSearchCV with three-fold cross validation which tested the following values of alpha: .000001, .0001, .01, .1, .5, 1, and 2.

### Support Vector Machine

All Linear SVM models were built using GridSearchCV with three-fold cross validation which tested the following values of C: .3, 1, 4, 20, 50, 100, and 200.

### Neural Network

All Neural Network models used the adam optimizer, the sparse categorical crossentropy loss function, and were tuned by hand. The values of the dense rectified linear unit layer and the dense softmax activation layer were adjusted along with the number of epochs.

### Random Forest

All Random Forest models were build using GridSearchCV with three-fold cross validation which tested all combinations of max depth: 4, 10, 35, and 70 with max features: auto, log2, 0.15, None. Max features = 0.15 represents 15% of the number of features used; auto represents the square root of the number of features; log2 represents log2 of the number of features; and None represents the number of features.

# Results

## K-Means

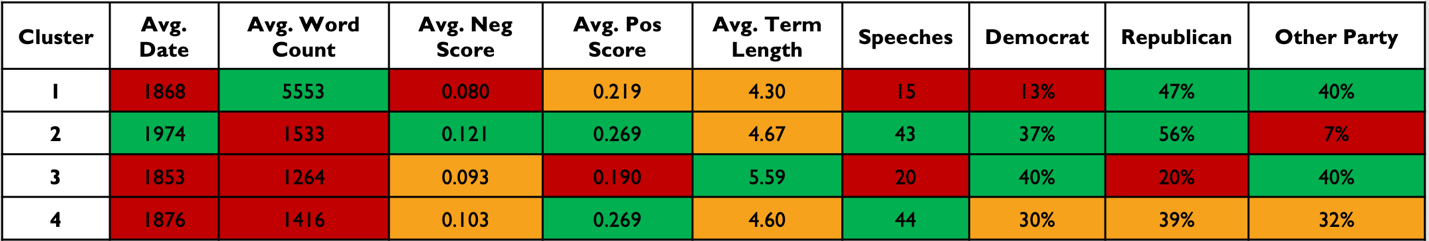
After extracting the characteristics of the two group and four group model outputs using the metadata for each president along with the general sentiment scores of each speech, the clusters for the 2 and four group models can are summarized in Tables 2 and 3, respectively.

Table 2 displays the statistical summaries for each cluster in the K = 2 model classifications.

## 

The two clusters shown above are noticeably distinct from each other in all metrics except term length. This is usually a helpful indicator that the term length variable is probably pretty independent of the other variables included in the analysis. We might classify cluster 1 and “New School” and cluster 2 as “Old School” because the average date of the speeches were over 100 years apart. Additionally, the New School cluster had comparatively shorter, more sentimental speeches and a more binary distribution of parties.

Table 3 displays the statistical summaries for each cluster in the K = 4 model classifications.



Date is once again a primary delimiter in the data, this time separating cluster 2 from the rest by about 100 years. Moreover, cluster 2 shares strikingly similar statistics to cluster 1 in the K=2 classifications. Cluster 1 shares the same period with 3 and 4, but is obviously an outlier in terms of average word count per speech. These first two differences are drastic to the extent that we can be confident in our assumption of how the K-means algorithm seeks to divide the vector space of the speeches dataset. The intuition that the text content of a modern presidential address is significantly different than that of older speeches from the 19th century is supported in these findings.

## Political Party

The base accuracy of predicting political party was 54.8%. Both the MNB and Random Forest models achieved 61.3% accuracy in predicting political party. These two models also have the greatest precision and recall. The Neural Network was less accurate (58.1%) and the SVM only achieved base accuracy. A table containing the best parameters of each model is presented below, along with confusion matrices of each model.

Table 4 displays the best model of each type in predicting political party

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Parameters | Test  Accuracy | Precision | Recall | F1- score |
| SVM | C = 100 | 0.548 | 0.53 | 0.52 | 0.48 |
| **MNB** | alpha = 10-6 | **0.613** | 0.62 | 0.59 | 0.58 |
| Neural Network | Dense relu, 50 Dense softmax, 2 epochs = 20 | 0.581 | 0.57 | 0.56 | 0.55 |
| **Random Forest** | max\_depth = 10  max\_features = log2 | **0.613** | 0.62 | 0.59 | 0.58 |

Figure 10 displays the confusion matrices of the best model of each type in predicting political party



The best model only classified 2 out of 31 more records than the base accuracy model. Predicting political party is not a simple task, especially since the roles of the political parties have changed and even switched at times.

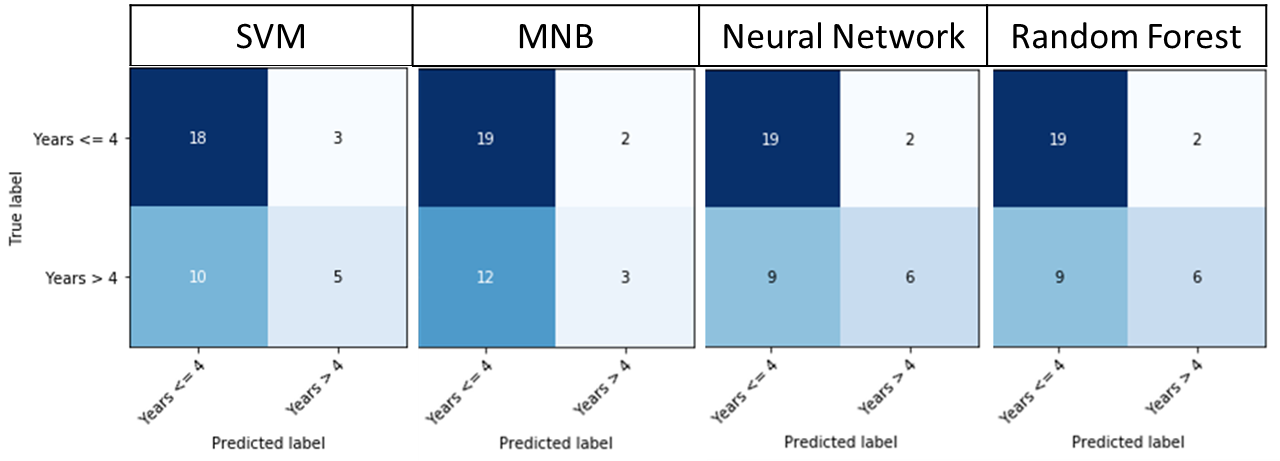
## Term Length

The base accuracy of predicting term length was 58.3%. Both the Neural Network and Random Forest models achieved 69.4% accuracy, 71% precision, and 65% recall. The SVM was less accurate (64%) while the MNB model performed worst (61%) but still better than base accuracy. A table containing the best parameters of each model is presented below, along with confusion matrices of each model.

Table 5 displays the best model of each type in predicting term length

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Parameters | Test  Accuracy | Precision | Recall | F1- score |
| SVM | C = 200 | 0.638 | 0.63 | 0.60 | 0.58 |
| MNB | alpha = 10-6 | 0.611 | 0.61 | 0.55 | 0.52 |
| **Neural Network** | Dense relu, 200 Dense softmax, 2 epochs = 40 | **0.694** | 0.71 | 0.65 | 0.65 |
| **Random Forest** | max\_depth = 35  max\_features = auto | **0.694** | 0.71 | 0.65 | 0.65 |

Figure 11 displays the confusion matrices of the best model of each type in predicting term length



The most accurate model predicted 4 out of 36 more records correctly than the base accuracy model. This is a stride in the right direction, but due to the untimely deaths of many presidents, the true relationship of re-electability and speeches may be lost.

The Neural Network predicts a different term length based on each of Donald Trump’s three selected speeches. The speeches and predictions of term length are presented below. It appears that the probability of re-election decreases over time. This is likely not the true trend since the model is not very accurate and there may not be enough data.

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Speech | Probability In Office <= 4 | Probability In Office > 4 |
| 1/20/2017 | Inaugural Address | 0.315 | 0.685 |
| 2/28/2017 | Address Joint Session Congress | 0.497 | 0.503 |
| 1/30/2018 | State Union Address | 0.536 | 0.464 |

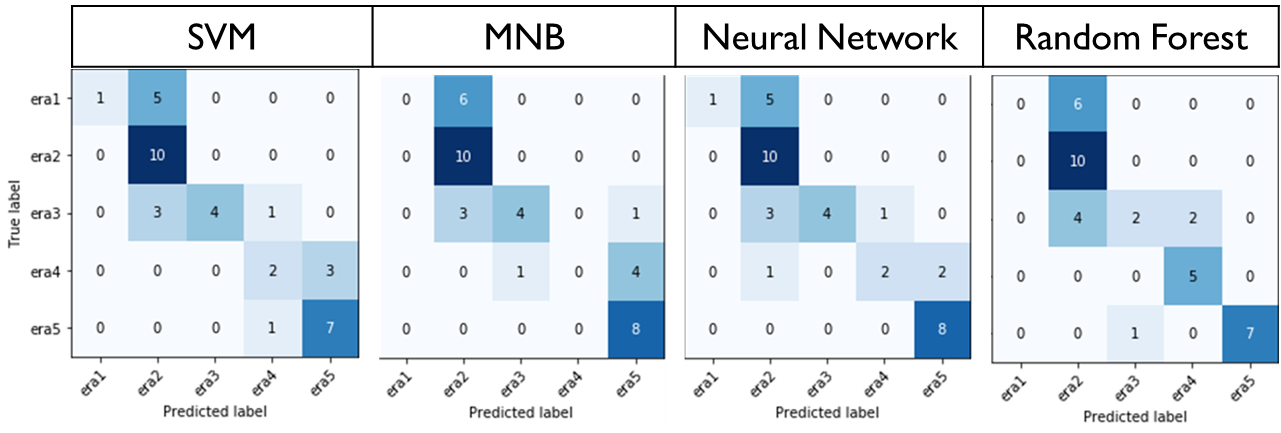
## Time Period

The base accuracy in predicting time period was 27% when predicting era2 (1844 – 1887) for all speeches. The Neural Network model was the most effective, predicting 68% of time periods correctly. This model had the greatest precision (80%) and nearly the best recall (61%) although the Random Forest model achieved 62% recall. The SVM and Random Forest models achieved 65% accuracy and the MNB model only achieved 60% accuracy. A table containing the best parameters of each model is presented below, along with confusion matrices of each model.

Table 6 displays the best model of each type in predicting time period

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Parameters | Test  Accuracy | Precision | Recall | F1- score |
| SVM | C = 200 | 0.649 | 0.75 | 0.59 | 0.58 |
| MNB | alpha = 10-6 | 0.595 | 0.39 | 0.50 | 0.41 |
| **Neural Network** | Dense relu, 800 Dense softmax, 6 epochs = 45 | **0.676** | 0.80 | 0.61 | 0.61 |
| Random Forest | max\_depth = 70  max\_features = None | 0.649 | 0.58 | 0.62 | 0.56 |

Figure 12 displays the confusion matrices of the best model of each type in predicting time period



The most accurate model predicted 15 out of 37 more records correctly than the base accuracy model. This is a very large improvement. The records misclassified were mostly from era2 (1844 – 1887) which is the class with the greatest number of records.

# Conclusions

The story we discovered in using text mining tools and methods to explore the most popular speeches of our past and present leaders is interesting, if not inspiring. In a political climate that is overwhelmingly characterized as extremely ideologically polarized, we found no observable connections between the rhetoric of a president as he addresses the public and the party to which he belonged or belongs. The speeches were consistently over twice as positive as they were negative. And lastly, the steadfastness of the individual voter was inversely supported by our inability to confidently predict a president’s ability to influence his chances of reelection based on his words. We are happy to conclude that the only meaningful way in which our leaders’ words hold power is on a case-by-case basis, considering the unique context of each speech. Though it would’ve been personally encouraging for our models to have predicted our metrics to 90+% accuracy, we believe it is better to prove through deconstruction that our democracy is not so easily manipulated as the cynics argue, at least based on our leaders’ publicly conveyed sentiments.

# References

1. <https://towardsdatascience.com/introduction-to-naive-bayes-classification-4cffabb1ae54>
2. <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>
3. <https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85>
4. <https://en.wikipedia.org/wiki/Precision_and_recall#F-measure>
5. <https://medium.com/machinevision/overview-of-neural-networks-b86ce02ea3d1>
6. <https://www.datascience.com/resources/notebooks/random-forest-intro>
7. <https://millercenter.org/the-presidency/presidential-speeches>
8. <https://time.com/3398873/10-shortest-stints-in-the-oval-office/>