CMPE 251 – FINAL PROJECT

Analysis of Presidential Speech Characteristics Including use of Deceptive Words

Travis Cossarini
20047244

CMPE 251 Project
Travis Cossarini – 20047244

Table of Contents

Introduction

Elections in democratic societies represent one of the most important opportunities for the general population to influence the policy of their country. The presidential election in the United States is often considered the epitome of this statement as the winner of this election is considered by many to be the leader of the Free World. The 2012 presidential election reportedly cost $7 Billion dollars, further highlighting the importance of occupying the Oval Office [1]. Speeches and other forms of public outreach are clearly extremely important in determining the outcome of a given election. Thus, it is no surprise that identifying trends leading to victory in speeches is a pertinent issue in politics.

Through analysis of Presidential speech characteristics from 1992-2012 this paper aims to discern these trends and be able to develop strategies and predictions for future elections based on these characteristics. Specifically focusing on the effect of the use of deceptive language in speeches.

The analysis and predictions in this report will be done using Python and associated libraries. All code can be seen in the Github linked in the Appendix.

Data

The dataset analyzed in this report consists of 431 speeches with the 1000 most common words used in these speeches being reflected as usage rates (number of times that word was used in reference to the length of the speech). Additionally, 76 deceptive words have been identified, with the corresponding number of times each word was used in a speech. The candidates name, as well as whether they won or lost the election is also in the dataset.

Data Exploration

The first step taken in exploring the data is investigating any issues that my bias or impede prediction during the later stages. The dataset is clearly biased towards winning candidates, with 76% percent of speeches being given by the winning candidate. While this could be explained by the winner reaching more voters through more speeches, there are other factors such as the 54 speeches given by incumbent Presidents during non-election years. With this considered, winning candidates still represent 62% of the remaining 359 speeches. The number of speeches given by winner and loser is shown in the graph below [Figure 1]. Winning candidates on average give 15.5 more speeches in election years than their losing counterparts. This bias towards winning candidates is an important factor to consider in the construction of predictors. Another possible issue is that elections in more recent years have many more speeches given, which can also bias the prediction [Figure 1].

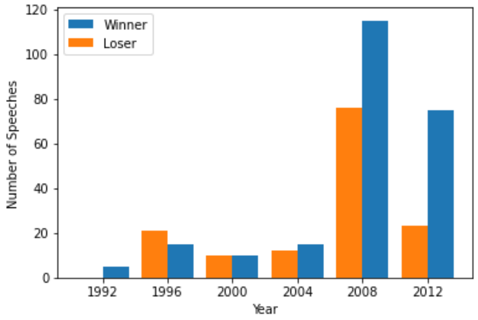

Figure 1: Number of Speeches Given by Winning and Losing Candidates

Language of Speeches

Examining the usage of different words could also be important in determining the winner of a given election. The word usage rates given in the dataset correspond to part of speech tagged (Stanford tags) words, these tags represent the context (surrounding words) in which the word was used in the speech [2]. By analyzing the correlation of each of these tags to winning the election, some insight can be gained. It is clear from the left figure of Figure 2 that the majority of words are not highly correlated with winning or losing. This implies that removing tags within a radius of zero could improve results. The right subfigure represents the correlations plotted against the length of the root word (For example at_CG would have a length of 2). The dataset clearly contains many more short words than long ones, with an average length of 5.6. Unfortunately, this clustering does not reveal any trends or identifiable grouping, indicating that prediction may be more complicated than simply using an attribute like word length.

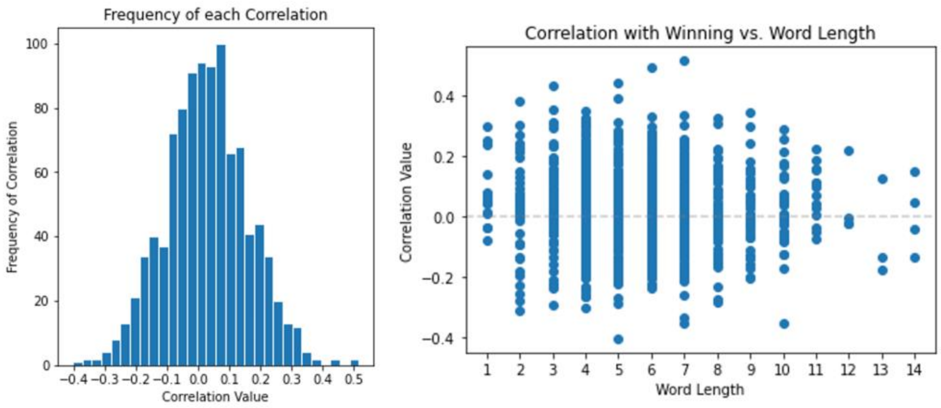

Figure 2: Correlations of Word Rates with Winning

Further investigating the relationship between correlation and word length, Figure 3 depicts the relationship between word length and correlation with winning. The average positive and negative correlation are relatively equally distributed indicating that word length is not important to determining the winner of an election. The positive bias of the overall correlation is most likely due to the inherent winner bias discussed above.

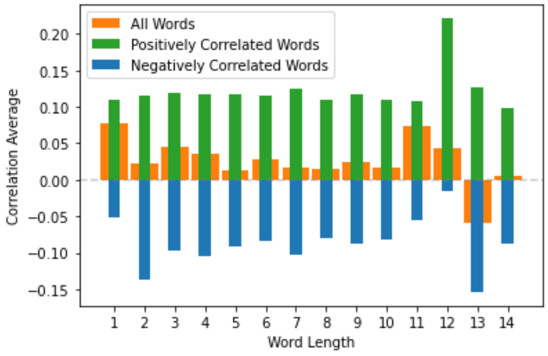

Figure 3: Average Correlation with Winning of Each Word Length

A more granular analysis of the correlated words, specifically the positive and negative extrema reveals an interesting anomaly, many of the most strongly correlated words correspond to "edge cases" in word use. For example, use of "Obama" is the most negatively correlated word in the entire dataset. This intuitively makes sense as in the 2012 and 2008 elections (and 2010 speeches) Obama was victorious. Obviously, he would rarely speak of himself in the third person during his speeches, so his losing opponents were the only candidates using this tag, hence the extremely strong negative correlation. A similar phenomenon exists for the term "President" though with a weaker correlation. This could indicate that words which occur in more election cycles would be have more actual predictive power, despite having lower correlations.

The other segment of the dataset includes the number of times different deceptive words were used. This data will be normalized to increase the performance of predictors, numerical counts with no reference to either the length of the speech or rate of use versus other speeches are unlikely to have any predictive power. Interestingly, by summing the number of deceptive words used in each speech it can be seen that higher use of any type of deceptive word is positively correlated with winning, with a correlation of ~0.22. This data is also biased towards winners for same reasons outlined above. Analysis for deceptive words is displayed below in Figure 4.

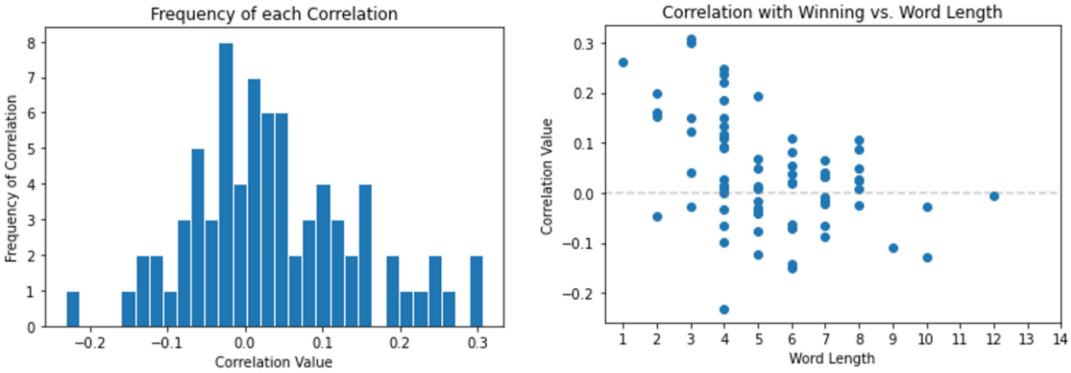

Figure 4: Correlations of Deceptive Word Use with Winning

Data Preparation

Based on the analysis above, to gain the most accurate predictions the data will need to be cleaned. A common practice in natural language processing is the removal of stop words, which are extremely common words with little significance to the meaning of a sentence. Using a Python library, all stop words were removed from the 1000-word dataset comprising input set 2, which contains 848 words [3].

In all input datasets, input values were min-max normalized to the interval [0,1] to increase predictor accuracy. Normalisation of the Raw Words data results in an average 4% increase in accuracy across all predictors discussed in this report. Z-score normalisation was also tried, though it made on appreciable difference in the accuracy of predictions.

Words that were used in specific elections, such as "Obama" were removed for input set 3. This was done through removing words that appeared in fewer than 5 election cycles (including non-election years). This removed 15 words, including "Obama" and "Romney". While this strategy removes problematic words like these, it also removes new issues, like “solar” from the dataset, making it insensitive to more recent social issues.

The final set of attributes used is the usage rates of the actual POS tags. As can be seen in Figure 5, each tag has a unique correlation and this serves as a form of dimensionality reduction as there are only 65 unique tags.

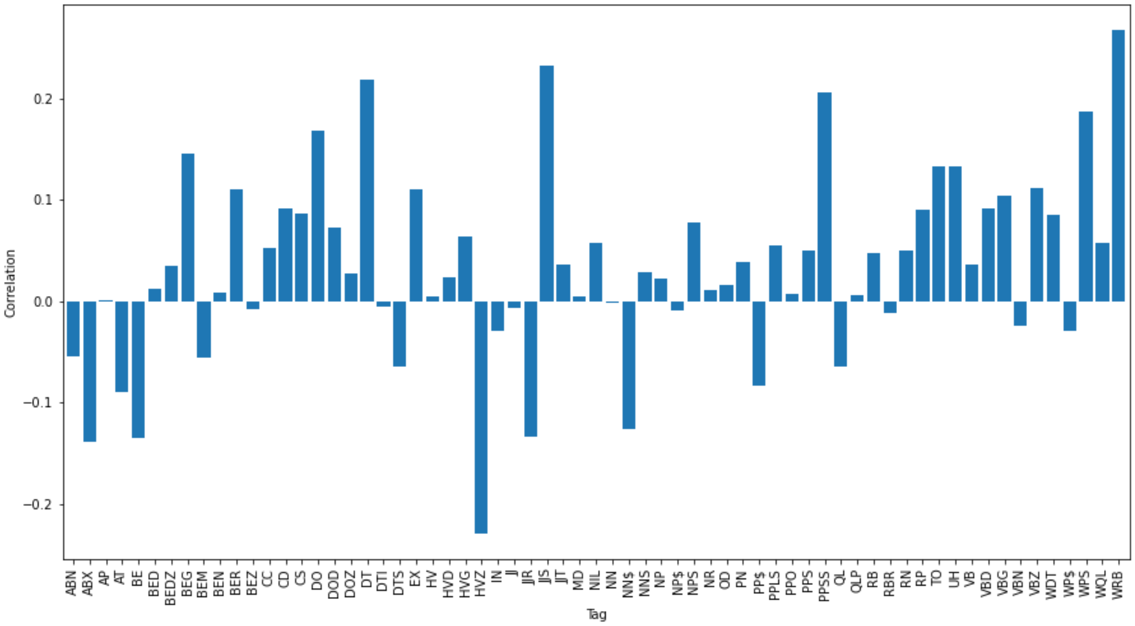

Figure 5: Correlations with Winning of Stanford Tags

Models were also tested on high and low correlation subsets of the 1000-words and deception words, though these input sets yielded low accuracies and as such will not be discussed.

Datasets

Dataset Title	Description
Full Words	All words and their usage rates normalised to [0,1]
Stanford Tags	All 65 Stanford tags in the dataset and their usage rates, normalised to [0,1]
Deception Normalized	The usage of deceptive words in each speech normalised to [0,1]
No Stop Words	Full Words with all stop words removed, normalised to [0,1]
Words in >5 Campaigns	Words that were used in at least one speech in more than 5 campaigns, normalised to [0,1]
Combined High Correlation	All words in the Full Words set and Normalised deception set with correlation magnitudes >0.05.
Total Combined	All 1000 words as well as the normalised deception counts.

Dimensionality Reduction

All the above datasets were transformed using principle component analysis to accomplish dimensionality reduction with the goal of increasing predictor performance by reducing the number of attributes considered. The PCA model was set to retain 99% of the explained variance in the original set. This resulted in a reduced feature space of 299 features for the 1000 words set. When applied to predictors, this reduction from 1000 features does not improve accuracy by a meaningful amount, as can be seen in the appendix accuracy table.

Since PCA was not successful in improving accuracy of predictors, Singular Value Decomposition was applied, this allowed for more flexibility in determining the dimension of the output feature space. Using SVD to reduce to two dimensions yields the following scatter plot, though there are no identifiable clusters within the plot.

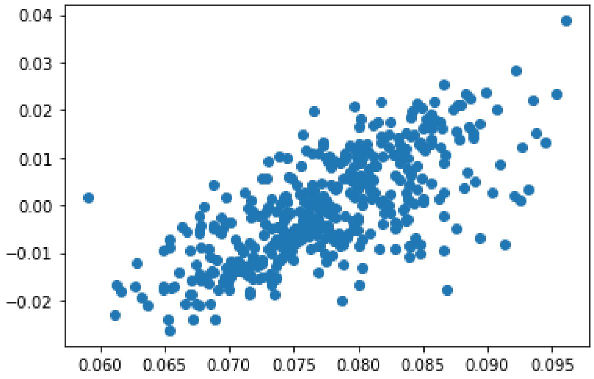

Figure 6: SVD 2 Dimension

A dimension of five was chosen for SVD as it offered the best balance between descriptive power on the original set and reduction in the size of the feature space. The dimension was chosen through testing the accuracy of a naïve Bayes classifier on several SVD sets with different dimensions, displayed below:

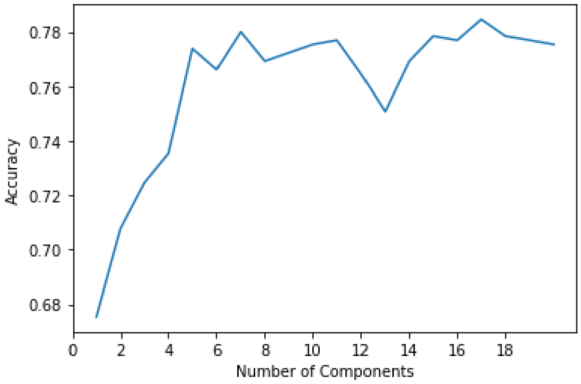

Figure 7: Comparison of Different SVD Dimensions on NB Classifier

Finally, Linear Discriminant Analysis was applied to each dataset, reducing the dimension to 1 single attribute [4]. LDA is a more sophisticated version of SVD that retains as much of the class discriminatory information as possible, leading to more accurate predictions.

Linear Discriminant Analysis was applied to transform the data to two dimensions. LDA is a more sophisticated version of SVD that retains more of the class discriminatory information when compared to SVD. LDA was applied based on the candidate that gave the speech, yielding the following scatter plot, with each candidate shown as their own color. Clearly, through LDA there are identifiable trends between candidates that may allow for the prediction of an election, this idea will be explored later in this report.

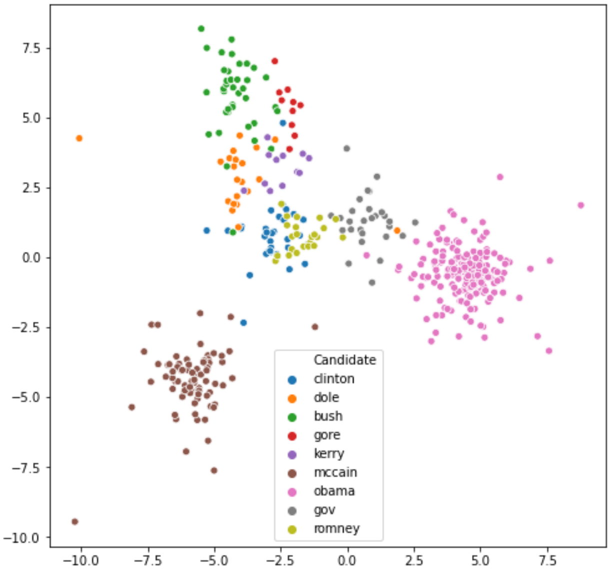

Figure 8: LDA Clustering by Candidate

Predictors

To evaluate each model, 5-fold cross validation was used to resist overfitting. Predictor results are displayed across all datasets, and their reduced dimension counterparts. Further analysis is done on the top performing set. All predictors are used to classify the outcome of the election as either a win or loss, no regressors are used. Predictors used are from the Python library Scikit-Learn [5].

KNN

Initially, simple predictors were used to ascertain any issues with the data and check the possibility that a computationally simplistic solution exists, as well as identifying any underlying issues in the dataset. K was chosen to be 4 which provided the maximum accuracy, as shown in Figure 9.

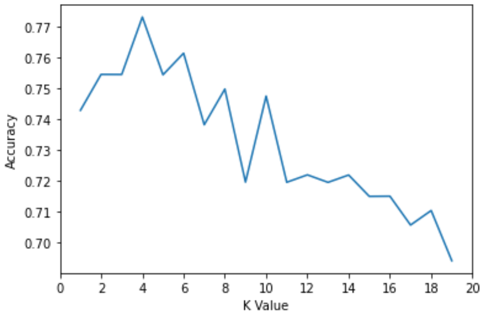

Figure 9: Comparison of K Value vs. Accuracy

Accuracy Scores K = 4	Full Words	No Stop Words	Stanford Tags	Deception Words	Combined High Correlation	Words in >5 Speeches	Total Combined
Normal	86.3%	83.8%	78.9%	65.5%	87.7%	86.2%	65.5%
SVD	78.9%	74.0%	72.8%	77.4%	85.2%	78.6%	69.7%

Due to the size of the dataset KNN was slower to compute than the predictions of other models, but there was no training time involved. This model provides the first indication that SVD is losing too much information during the reduction to be effective at predicting the outcome of an election.

Metric	Accuracy
Accuracy	87.7
F1 Score	0.913
Recall	0.969
Precision	0.862

Clearly, the accuracy deficit of KNN is due to low precision, indicating that the classifier is generating too many false positives. Perhaps this is indicative of the positive bias (76%) of the dataset impacting the output of KNN. This theory is further reinforced by the high recall, indicating that there are very few false negatives.

Naïve Bayes

Naïve Bayes predictors do not usually perform well in situations with many inputs due to their underlying assumption of independence between attributes. However, NB handles the positive bias of the dataset better since it is not relying on the values of neighboring records. This predictor performed better than KNN, indicating that correlations between words may not have massive predictive power. NB performed particularly poorly on the dataset comprised of deceptive words, indicating that these word's covariances are much more important to predictions.

Accuracy Scores	Full Words	No Stop Words	Stanford Tags	Deception Words	Combined High Correlation	Words in >5 Speeches	Total Combined
Normal	86.3%	86.8%	77.1%	48.0%	86.0%	85.1%	83.2%
SVD	77.4%	74.6%	76.2%	79.7%	78.9%	77.2%	66.9%

Metric	Accuracy
Accuracy	86.3%
F1 Score	0.897
Recall	0.898
Precision	0.896

In support of the theory that NB would manage the winning bias of the dataset better, the recall and precision of this model are roughly in balance, indicating that the problem of false positives has been solved.

Ensemble Classifiers

Ensemble classifiers are the natural next step in predicting the outcome of elections. The ensemble classifiers compared are AdaBoost, gradient Boost, random forest, and bagging classifier. All these ensemble predictors are based on the random forest paradigm and use a decision tree as the basic unit. AdaBoost employs the AdaBoost boosting function whereby, as successive trees are trained, the weights of incorrectly classified instances are increased so that the classifiers focus on more difficult cases [6]. Gradient boosting employs a similar strategy with a negative gradient of the loss function being applied to reweight difficult tasks. The bagging classifier makes use of sampling with replacement from the data and building a predictor based on each of these subsets. Bagging is more resilient to overfitting than boosting.

These models were applied to the 5 input datasets to determine the optimal input with their accuracies displayed below:

		Full Words	No Stop Words	Stanford Tags	Deception Words	Combined High Correlation	Words in >5 Speeches
Bagging Classifier	Normal	88.9%	85.5%	82.2%	74.6%	88.5%	87.5%
	SVD	80.8%	79.4%	72.9%	74.5%	84.6%	80.5%
Gradient Boost	Normal	90.3%	89.5%	83.2%	75.2%	90.9%	91.2%
	SVD	80.3%	77.5%	73.7%	76.0%	84.2%	80.9%
Random Forest	Normal	88.6%	91.1%	83.2%	76.6%	82.5%	87.7%
	SVD	80.0%	80.3%	74.5%	74.5%	84.9%	79.2%
AdaBoost	Normal	91.1%	88.5%	82.5%	70.5%	90.9%	90.8%
	SVD	76.8%	75.7%	73.8%	79.1%	82.5%	76.9%
Mean Accuracy	Normal	89.7%	88.0%	82.8%	74.2%	90.0%	89.3%
	SVD	79.5%	78.2%	73.7%	76.0%	84.0%	79.4%

The results below were calculated using 100 basic estimators for each predictor on the combined high correlation dataset which yielded the highest average accuracy across the ensembles:

Metric	Bagging	Gradient Boost	AdaBoost	Random Forest
Accuracy	88.9%	90.9%	91.1%	88.6%
F1	0.914	0.932	0.930	0.923
Recall	0.919	0.944	0.921	0.948
Precision	0.909	0.920	0.941	0.901

Clearly, all the chosen predictors performed very well, especially given the relatively small number of estimators used. Interestingly, the AdaBoost classifier performed the best in terms of prediction accuracy, this implies that there are more difficult speeches to predict giving an advantage to the boosting function. Once again, the dimension reduction appears to be losing too much information to be able to make predictions, even in an ensemble.

Since the combined high correlation dataset produced the best results, this is the set used to optimise for the number of basic iterators used, which can be seen in Figure 10.

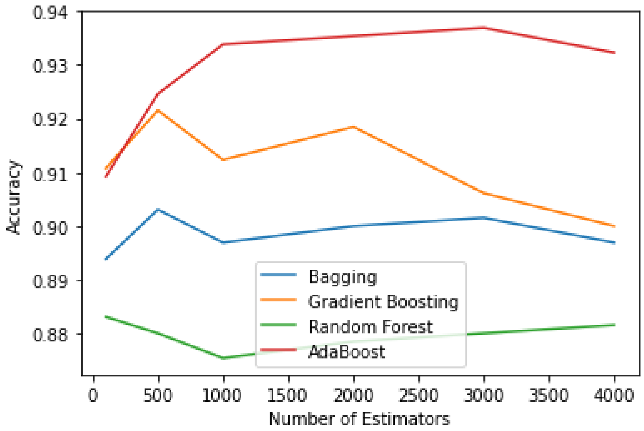

Figure 10: Ensemble Comparison on Combined High Correlation Data

Evidently, the best ensemble accuracy is reached by AdaBoost with 3000 basic iterators and an accuracy of 93.9%. This increase in the size of the ensemble also training time, but not beyond the bounds of what is possible on a small computer such as a laptop.

Multi-Layer Perceptron

The MLP construction consisted of 3 hidden layers, with 50, 25 and 25 neurons, respectively. 100 hidden neurons were used to optimize the structure as 100 is that square root of 1000, the average number of inputs across the sets. 50-25-25 arrangement was found to be optimal through repeated testing.

Accuracy Scores	Full Words	No Stop Words	Stanford Tags	Deception Words	Combined High Correlation	Words in >5 Speeches	Total Combined
Normal	90.9%	91.5%	66.3%	72.6%	91.1%	90.6%	77.8%
SVD	66.3%	66.3%	67.2%	79.4%	85.1%	66.3%	70.8%

The MLP performed very well, with scores nearing that of the ensemble predictors. One downside of this model is that it took much longer to train than other the other models discussed.

Metric	Accuracy
Accuracy	91.5%
F1 Score	0.934
Recall	0.937
Precision	0.930

Support Vector Machines

A support vector machine with an RBF kernel was also used in the hopes that it would excel in the lower dimensional SVD spaces. However, it underperformed both the MLP and ensemble classifiers. To see SVM results please refer to the accuracy table in the appendix.

LDA

Clearly, the SVD dimension reduction did not assist with prediction. However, using LDA as described above to reduce the dimension of the Combined High Correlation set to a single vector yields extremely good results. Using the LDA 1-dimensional reduced data on the MLP network produced an accuracy of

99.5%, while the AdaBoost predictor with 3000 basic estimators yielded an accuracy of 99.7%. These models were also tested on LDA 2-dimensional data, though the accuracies were not as high.

Conclusion

The word makeup of speeches appears to be highly associated with winning an election, using linear discriminant analysis and the application of a neural net, election's result could be predicted with 99.7% accuracy using sophisticated models.

Interestingly, there is little correlation with the length of words used in a speech and the chance of winning an election. It is far more effective to increase the amount of deceptive words used. Without any sort of data prep, the use of deceptive language had a positive correlation of 0.22 with winning an election. When deceptive words were introduced to the dataset for prediction, accuracy increased by as much as 5% for some models. This indicates that deceiving the general population can have a positive effect on your election chances.

Through attribute importance analysis on the highest accuracy AdaBoost model, the 5 most impactful words are: (according to Gini Importance)

Work	Chance	Union	Americans	Change
Pay	Country	Best	Work	Simple

This implies that the use of words emphasising collective responsibility and hope appear to be important, which depending on the reader's worldview, can be seen in a positive or cynical light.

One downside of LDA is that it produces "black box" attributes through dimension reduction, so it is impossible to gain insight on which attributes are important to the most accurate models.

Input Set Comparison

Below are the average accuracies across all predictors tested for the datasets described in the data preparation section. Dimension reduction is not included in these accuracies.

	Full Words	No Stop Words	Stanford Tags	Deception Words	Combined High Correlation	Words in >5 Speeches	Total Combined
Average Accuracy	89.3%	88.9%	78.7%	73.8%	90.3%	88.9%	85.0%

The combined, high correlation dataset provided the best results over all the predictors. Surprisingly, the dataset with stop words removed did not outperform the full data set, producing on average accuracies that are 0.4% worse. This implies that connecting words in certain situations (with certain Stanford tags) have importance to victory in an election. Another surprisingly, ineffective input set was the set of Stanford tags, underperforming full words by 11.6% across the predictors. The high correlation subset of the combined deception words and 1000 common words outperformed the regular combined data by 5% on average.

In all but one election year, the candidate that gave the most speeches won the election. This indicates that perhaps reaching the most people is an important, if simplistic, factor as well.

References

[1] J. K. Trotter, "The 2012 Presidential Election Cost $7 Billion," *The Atlantic*, Feb. 01, 2013. https://www.theatlantic.com/politics/archive/2013/02/2012-presidential-election-cost-7-billion-dollars/318783/ (accessed Nov. 25, 2020).

[2] "The Stanford Natural Language Processing Group." https://nlp.stanford.edu/software/tagger.shtml (accessed Nov. 26, 2020).

[3] "2. Accessing Text Corpora and Lexical Resources." https://www.nltk.org/book/ch02.html (accessed Nov. 26, 2020).

[4] "sklearn.discriminant_analysis.LinearDiscriminantAnalysis — scikit-learn 0.23.2 documentation." https://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html#sklearn.discriminant_analysis.LinearDiscriminantAnalysis.fit_transform (accessed Nov. 27, 2020).

[5] "scikit-learn: machine learning in Python — scikit-learn 0.23.2 documentation." https://scikit-learn.org/stable/index.html (accessed Nov. 26, 2020).

[6] R. Rojas, "AdaBoost and the Super Bowl of Classifiers A Tutorial Introduction to Adaptive Boosting," p. 6.

Appendix

Git

https://github.com/Tcoss7/CMPE251Project

Accuracies

	KNN	Naive Bayes	MLP	Decision Tree	Bagging Classifier	Gradient Boost	Random Forest	AdaBoost	SVM
Full Words	86.3%	86.3%	90.9%	79.7%	88.9%	90.3%	88.6%	91.1%	85.7%
No Stop Words	83.8%	86.8%	91.5%	80.0%	85.5%	89.5%	88.6%	88.5%	89.7%
Stan Tags	78.9%	77.1%	66.3%	77.5%	82.2%	83.2%	83.2%	82.5%	74.6%
Deception	65.5%	48.0%	72.6%	67.8%	74.6%	75.2%	76.6%	70.5%	73.1%

Combined	87.7%	86.0%	91.1%	80.3%	88.5%	90.9%	89.5%	90.9%	90.9%
Words in >5 Speeches	86.2%	85.1%	90.6%	80.5%	87.5%	91.2%	87.7%	90.8%	85.4%
Complete Combined	65.5%	83.2%	77.8%	80.0%	89.1%	91.1%	87.5%	90.9%	73.4%
LDA2 FW	89.4%	77.7%	90.5%	89.5%	90.3%	90.6%	90.6%	90.3%	85.2%
LDA2 CHC	99.5%	92.9%	99.7%	99.2%	99.2%	99.2%	99.5%	98.9%	99.5%
lda1 Full Words	98.9%	98.9%	98.9%	98.6%	98.6%	98.6%	98.6%	98.6%	98.8%
lda1 No Stop Words	99.4%	99.2%	99.2%	99.4%	99.4%	99.4%	99.4%	99.4%	99.4%
lda1 Stan Tags	88.3%	89.4%	88.8%	85.5%	85.5%	85.5%	85.5%	86.8%	88.6%
lda1 Deception	77.4%	79.7%	79.4%	74.5%	74.5%	76.0%	74.5%	79.1%	79.4%
lda1 Combined	99.7%	99.7%	99.7%	99.7%	99.7%	99.7%	99.7%	99.7%	99.7%
lda1 Words in >5 Speeches	98.9%	98.9%	98.9%	98.0%	98.0%	98.0%	98.0%	98.0%	98.6%

lda1 Complete Combined	98.8%	98.5%	98.6%	97.8%	97.8%	97.8%	97.8%	97.8%	98.6%
PCA Full Words	85.8%	63.8%	88.8%	71.5%	77.1%	78.9%	69.5%	76.6%	89.2%
PCA No Stop Words	84.9%	64.0%	86.5%	72.9%	80.0%	81.1%	70.6%	76.8%	89.4%
PCA Stan Tags	79.4%	74.8%	66.3%	70.2%	79.2%	78.5%	76.5%	78.8%	84.6%
PCA Deception	65.7%	63.7%	72.3%	64.0%	71.2%	68.9%	71.5%	68.9%	72.8%
PCA Combined	87.7%	67.8%	88.3%	79.5%	84.0%	84.8%	74.2%	82.9%	89.7%
PCA Words in >5 Speeches	85.8%	64.0%	88.6%	70.9%	77.5%	76.6%	72.0%	75.8%	89.4%
PCA Complete Combined	65.7%	63.8%	72.2%	63.8%	70.9%	70.3%	70.6%	69.2%	72.8%
svd5 Full Words	78.9%	77.4%	66.3%	78.0%	80.8%	80.3%	80.0%	76.8%	79.7%
svd5 No Stop Words	74.0%	74.6%	66.3%	71.5%	79.4%	77.5%	80.3%	75.7%	78.6%
svd5 Stan Tags	72.8%	76.2%	67.2%	70.5%	72.9%	73.7%	74.5%	73.8%	72.3%

svd5 Deception	77.4%	79.7%	79.4%	74.5%	74.5%	76.0%	74.5%	79.1%	79.4%
svd5 Combined	85.2%	78.9%	85.1%	81.4%	84.6%	84.2%	84.9%	82.5%	82.8%
svd5 Words in >5 Speeches	78.6%	77.2%	66.3%	77.7%	80.5%	80.9%	79.2%	76.9%	78.8%
svd5 Complete Combined	69.7%	66.9%	70.8%	67.2%	72.5%	71.2%	71.8%	71.1%	69.7%