CISC/CMPE 251 - Project Report

Allison Christensen (10211533)

1 Introduction and Purpose

The following document describes a solution to an open-ended data analysis design project. The purpose of the assignment is to demonstrate an ability to use the skills we have covered in lecture to make reasonable design choices. We are then to use the observed results to suggest next steps to best understand the system. The provided dataset describes the word usage in U.S. presidential elections from 1992 to 2008. Each speech is associated with a candidate who either won or lost in a particular election campaign.

The main goals of the analytics are as follows:

1. To make the most accurate prediction of the likelihood of winning a US presidential election based on the words used in speeches.

2. To find some of the words that might be associated with winning speeches

3. To see if deceptive language has any role in winning US presidential elections.

In addition to these goals we may be interested in what these results collectively suggest about the average American voter and general strategies for good speeches.

2 Data

2.1 Description

The provided dataset was composed of 6 separate .csv files:

1. **Speeches.csv** contains descriptions of who and when for each speech.
2. The file **winners.csv** contains the information about whether each speech is associated with a winning or losing candidate (The target attribute).
3. The file **mostfreq1000docword.csv** is a document-word matrix. Each row corresponds to a speech, each column to one of the words described
4. **Mostfreq1000word.csv** contains the words described in mostfreq1000word.csv and their associated part-of speech (POS) tagging. POS or grammatical tagging refers to the process of marking up a word in a text as corresponding to a particular part of speech, based on both its definition and its context.
5. **Deceptionword.csv** contains similar data, but for a set of words for which changes in rates are associated with deception.
6. **Deceptiondocword.csv** contains the occurrences of words associated with deception.

The frequencies described in the data were collected and divided by the total length of each speech, thus converting the values to rates. This allows the speeches to be compared to one another.

2.2 Issues & Cleaning

The processing and preliminary analysis of the dataset was conducted using Pandas, a software library written for the Python programming language. It offers data structures and operations for reading in csv data and manipulating numerical tables.

The mostfreq1000word.csv presented some initial processing challenges. In terms of cleaning, I first ensured that there were no duplicated rows and then broke up the single attribute into two columns: 1. word and 2. POS tag for ease of processing moving forward.

I then removed the missing values and certain words I encountered that were composed of character symbols (quotation marks, hyphens, etc.). Some of these symbols were not labelled as symbols or characters by their POS tags which is what made this challenging. I am operating under the assumption that these symbols are not overly valuable to eventually predicting an election outcome.

In this step I also manipulated and joined the various tables such that each word aligned with its occurrence data. The table was formatted such that the outcome column (win/loss) appended the occurrence matrix. This can also be performed in KNIME, but for some quick statistical analysis and visualization, it was straightforward to implement in python. To avoid any duplication of effort, the cleaned table was saved to a new csv to later be read into KNIME.

2.3 Preliminary Data Exploration

As a first step in investigating the processed data, I obtained the average word occurrence for winning and losing speeches, respectively. The top 20 most frequently occurring words found in losing and winning speeches were as follows:

	0	1	POS_tag	word
0	0.045227	0.040089	AT	the
1	0.034644	0.031822	CC	and
3	0.025013	0.020689	IN	of
2	0.021034	0.022585	TO	to
4	0.017429	0.017548	AT	a
5	0.017005	0.015834	IN	in
7	0.012510	0.011412	PPSS	i
6	0.011811	0.015332	PPSS	we
8	0.010242	0.010548	IN	for
11	0.009899	0.008124	IN	to
9	0.009494	0.008907	BEZ	is
10	0.009439	0.009031	PPS	our
16	0.008122	0.005267	MD	will
14	0.006989	0.006250	HV	have
12	0.006396	0.009004	CS	that
13	0.005996	0.008179	DT	this
18	0.005320	0.005315	IN	on
15	0.005314	0.006000	BER	are
21	0.005026	0.004389	BE	be
22	0.004961	0.004308	IN	with

Figure 1 – High frequency words from losing speeches.

	0	1	POS_tag	word
0	0.045227	0.040089	AT	the
1	0.034644	0.031822	CC	and
2	0.021034	0.022585	TO	to
3	0.025013	0.020689	IN	of
4	0.017429	0.017548	AT	a
5	0.017005	0.015834	IN	in
6	0.011811	0.015332	PPSS	we
7	0.012510	0.011412	PPSS	i
8	0.010242	0.010548	IN	for
10	0.009439	0.009031	PPS	our
12	0.006396	0.009004	CS	that
9	0.009494	0.008907	BEZ	is
13	0.005996	0.008179	DT	this
11	0.009899	0.008124	IN	to
14	0.006989	0.006250	HV	have
15	0.005314	0.006000	BER	are
17	0.003267	0.005686	PPSS	you
18	0.005320	0.005315	IN	on
16	0.008122	0.005267	MD	will
19	0.004408	0.005031	WPS	that

Figure 2 - High frequency words from winning speeches

The most frequently recorded word was “the” in both the winning and losing speeches. The frequencies of “the” in the two classes was relatively close. As we are looking to make a prediction one way or the other, a more useful approach may be to find a list of words that appear very frequently in the winning speeches but not in the losing speeches, or vice versa. I then created a new column for the “Difference in Frequency” of the word usage. This metric is obtained by subtracting the losing frequency of a word by the winning frequency and keeping the absolute value. The new top 20 list of words with the greatest difference in average frequency was as follows:

	0	1	POS_tag	word	Frequency_Diff
0	0.045227	0.040089	AT	the	0.005138
3	0.025013	0.020689	IN	of	0.004324
6	0.011811	0.015332	PPSS	we	0.003521
16	0.008122	0.005267	MD	will	0.002855
1	0.034644	0.031822	CC	and	0.002822
40	0.000953	0.003574	DT	that's	0.002621
12	0.006396	0.009004	CS	that	0.002608
17	0.003267	0.005686	PPSS	you	0.002420
13	0.005996	0.008179	DT	this	0.002182
11	0.009899	0.008124	IN	to	0.001775
49	0.003712	0.002097	PPS	he	0.001615
2	0.021034	0.022585	TO	to	0.001551
55	0.001140	0.002671	CS	because	0.001531
32	0.002304	0.003726	WDT	what	0.001423
87	0.000391	0.001777	PPSS	we've	0.001386
322	0.001423	0.000105	NP	obama	0.001318
88	0.000374	0.001644	VBD	got	0.001270
44	0.003729	0.002564	PP$	my	0.001165
99	0.002077	0.000958	NN	government	0.001119
64	0.001261	0.002364	VB	want	0.001103

Figure 3 - Words with highest frequency difference

What we may hope to see, ideally, is a list with words of very different frequencies. Meaning, we want something in one that is not often in the other. Instead, the frequency differences are not very high and many of the words in this new list are similar to those in the previously seen lists.

The following histogram was produced to compare the difference in spread in average word frequencies between winning and losing speeches:

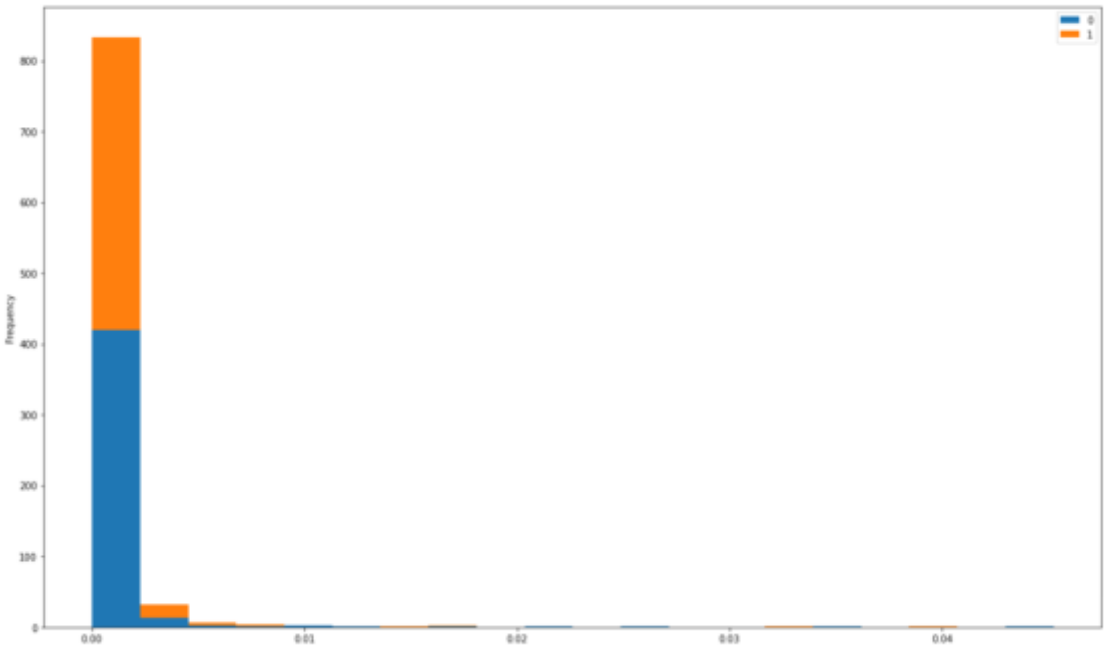


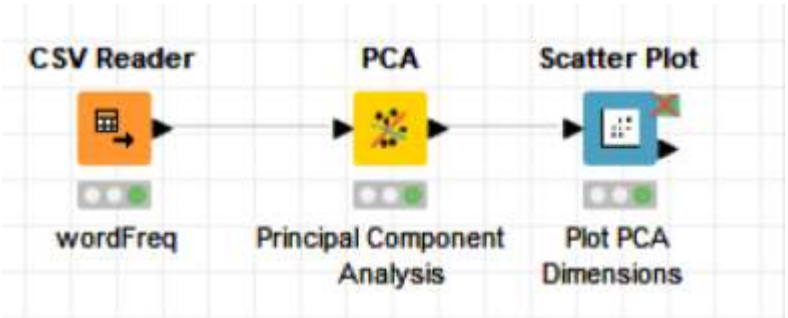
Figure 4 - Histogram of word frequencies in winning(1) and losing(0) speeches

Based on the above, losing speeches appear to have a wider spread of word frequencies compared to winning speeches. This may indicate that winning speeches tend to utilize repetition more. It may also be possible that frequent repetition of certain words could connect to a particular political stance or a slogan that resonates more with voters.

I considered searching for any pairwise correlation in this preliminary investigation, but it did not appear feasible given the large number of attributes. It would be interesting to evaluate frequencies of words occurring together, given the above spread.

2.4 Clustering

Clustering allows us to assess how difficult the prediction problem may be and discover any exceptional behavior of the system. Using the principal component analysis (PCA) node in KNIME, the input data was projected from its original feature space into a space of lower dimension (2D) with a minimum of information loss. The simple clustering workflow is as follows:



The result is a scatter plot, based on the word attributes, as seen below:

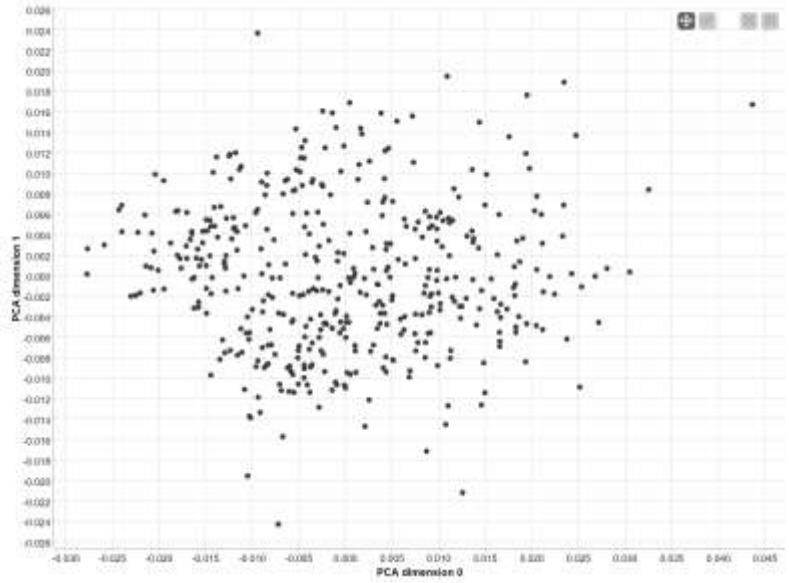


Figure 5 - PCA scatter plot depiction

Moving forward with the data as is, we can conclude that the prediction problem is not going to be particularly 'easy'. This is not surprising given the observed overlaps in high frequency words between winning and losing speeches. It may simplify the prediction problem and yield tighter clusters should we explore subsets of possible markers and visualize, for example, the variation in nouns versus verbs or adjectives.

3 Method

3.1 Predictor Selection

The basic prediction problem is ultimately a 2-class prediction problem: Given a speech, will the candidate win or lose?

With 1000 words in the dataset, we are dealing with many attributes. A Bayesian Predictor would therefore not be appropriate. Similarly, KNN would be an expensive process, predicting based on hundreds of words. Moreover, neither of these approaches offer a deeper understanding and it may be valuable for us to not only know the 'what', but also the 'why'. Neural Networks have potential for this dataset; however, these can be difficult to interpret and they can take a lot of time in the training phase.

Predictor	Yes or No
K Nearest Neighbor	✗

Bayesian Predictor	✗
Neural Network	✗
Support Vector Machine	✓
Random Forest	✓

Of the techniques discussed in this course, two predictors seem appropriate for this problem: support vector machines and random forests - we just need to be clear about what we are predicting.

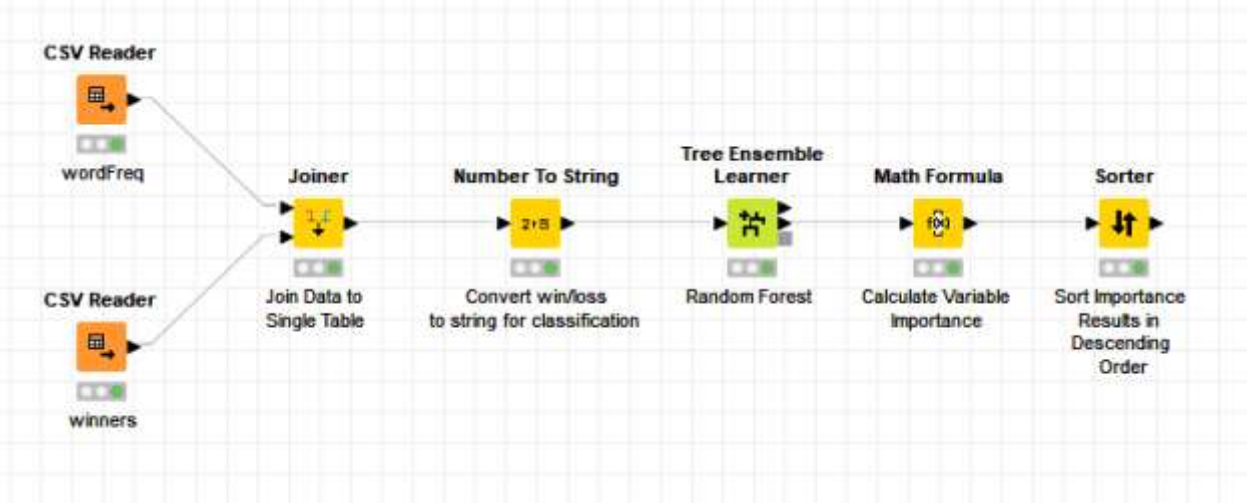
The Random Forest technique is intrinsically suited for multiclass problems, while SVM is intrinsically two-class. Random Forests work well with a mixture of numerical and categorical attributes and when features are on different scales. Roughly Speaking, with Random Forest, you can use data as is. SVM maximizes the “margin” and thus relies on the concept of “distance” between different points. As a result, further min-max or other scaling becomes important as a preprocessing step.

Based on the nature of the prediction problem SVM appears to be the best choice for our Win/Lose prediction (Goal 1). Random Forest, however, would be useful for establishing the influence of attributes on the success of a campaign (Goal 2).

3.2 Attribute Importance

A set of attributes together can have more predictive power than the sum of their individual predictive power. Random Forests are often used for attribute selection because the tree-based strategy naturally ranks how well a feature improves the purity of the node. Nodes with the greatest decrease in impurity happen at the start of the trees, while nodes with the least decrease in impurity occur at the end of trees. Thus, by pruning trees below a particular node, we can create a subset of the most important features.

In KNIME, the Tree Ensemble node offers a second output which provides details about attribute importance. It allows us to view and assess how often a word was used for building a decision tree and at what level it was found. As a measure of the importance, we can divide the splits with its candidate at each level and sum the respective values at each level:



The above workflow produced a list of words, each with a measurement of importance. The top 20 most important words in predicting a win or loss outcome, according to the above Tree Ensemble Learner, were as follows:

Variable Importance	Words
2.083333333333333	greater
1.9305555555555556	this
1.6428571428571428	spending
1.6190476190476188	obama
1.5	laughter
1.4	whatever
1.1818181818181819	i
1.1428571428571428	call
1.125	cost
1.0909090909090908	choices
1.0833333333333333	bring
1.0357142857142856	achieve
1.0	because
1.0	college
1.0	raise
1.0	balanced
0.8333333333333333	promised
0.8	wealth
0.7916666666666666	got
0.75	choice

Figure 6 - words with a high measurement of importance

This list is valuable in terms of making a more accurate prediction, however, it is important to note that we are not only interested in a list of attributes- we also want to know what it means. In other words, domain knowledge matters. Random Forest attribute selection is usually good for a first check because it does not require much tuning. However, there may be some benefit to experimenting with alternative feature selection strategies in future iterations.

Many classification experiments exploit a number of feature selection strategies. For example:

1	Using the topmost frequent 500 words, with no other filtering
2	Using the topmost frequent 500 words, with nouns only
3	Including only words which in majority of cases occur with the POS tab VB (verb); action words
4	Using only function words (AB, DT, HA, SN, etc. POS tags)
5	Using only the topmost words whose frequency is different in the two classes

Moving forward, it may be interesting to explore each selection scheme by simply making use of the information gain to find the most significant attributes. Information gain quantifies the information value in bits of knowing whether a feature is present or not given a certain classification task. A potential weakness of using this metric for feature selection would be that it does not provide any insight into the joint value of a set of features.

3.3 Deception Words

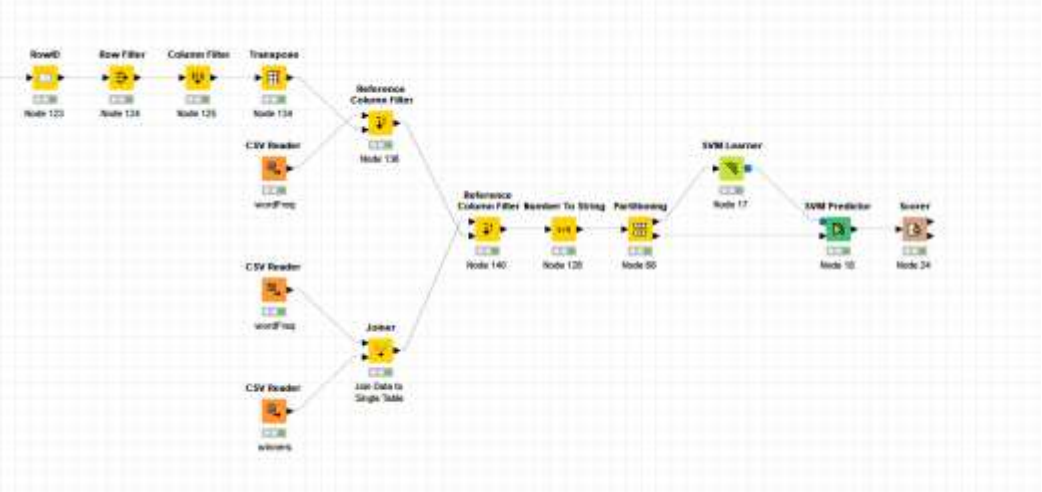
As the deception word data was provided in the form of word occurrences, these values were first turned into frequencies. This was done by dividing every word occurrence by the total word count in each speech. The word count was obtained by finding a word in deceptive doc that also occurred in frequencies doc and dividing by that frequency.

The goal of analyzing this data is to answer the following questions: Does deceptive language really influence election outcome and if so, does it help a candidate to win or lose?

Passing the deception word data through our random forest workflow demonstrated that deception words may have some small importance, but it was determined that they are not more present in winning speeches. In reviewing the deception word document, we can observe that this list consists mainly of words with a more negative connotation. There appears to be a high number of contraction words in this document as well.

3.4 Classifier Design

Using results obtained in the above analysis, only selected attributes whose importance is greater than 1.0 were used in the win/lose classifier design. The workflow is as follows:



As previously discussed, an SVM learner and predictor were selected for this prediction problem. The word frequency and outcome data is read in and attributes are filtered out based on the output of the Tree Ensemble Learner. The partitioning node splits the input table into two partitions: train and test data. The size of the training data set is 70% the size of the whole dataset. The data is linearly sampled to better compare the performance between runs with varying parameters.

As each candidate gave a different number of speeches, I ran into the issue of class imbalance. To work around this problem, the rows in the training data are resampled. The idea is to alter proportions of the classes of training data to obtain a classifier that can effectively predict the minority class. I integrated a Synthetic Minority Oversampling Technique (SMOTE) node into the workflow to synthetically oversample the minority class. This was chosen over other techniques as under sampling the majority class could lead to a loss of important information while oversampling exact copies of the minority class data could lead to overfitting concerns.

As a rule of thumb, it is best practice to use linear SVMs (logistic regression) for linear problems and nonlinear kernels such as the Radial Basis Function (RBF) kernel for non-linear problems. Based on the clustering results in section 2, the dataset does not appear to be perfectly linearly separable therefore the RBF Function was used in the SVM learner configuration.

SVM Parameter Tuning

The penalty parameter, C, tells the SVM optimization how much we want to avoid misclassifying each training example. Larger values of C correspond to a smaller margin hyperplane. We can view the effects of varying C in the below ROC Curve for this model:

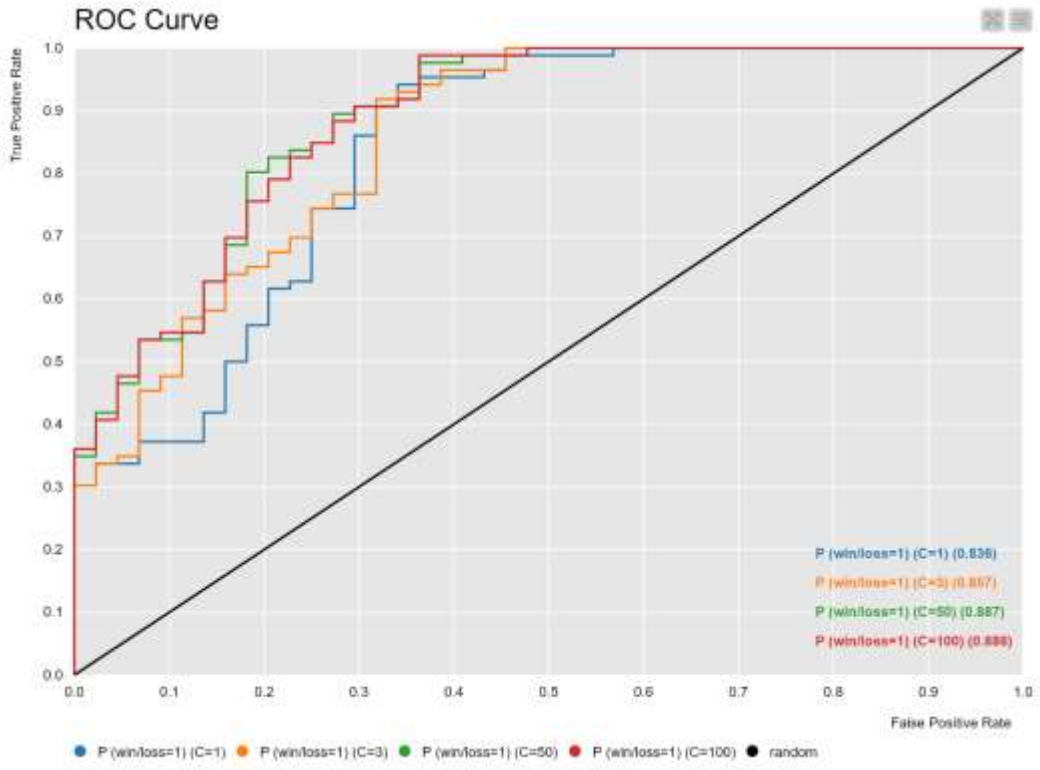


Figure 7 - ROC curve of varying C values in SVM classifier

Overall, the AUC values appear to indicate this is a fairly good classifier. This curve also demonstrates the tradeoff between sensitivity and specificity in our model.

A cross validation setup with a parameter optimization loop (available via the KNIME optimization extension) was integrated into the KNIME workflow and was used over the dataset to get the best C and sigma values for the data.

4 Discussion

4.1 Predictor Performance

After tuning, the performance of the SVM predictor was output as follows:

Correctly classified:	86
Incorrectly classified	44
Accuracy:	66.154%
Error	33.846%

The reported accuracy of 66.154% indicates that this is a reasonably performing predictor. A better approach would be to build more models to provide some indication of how well this predictor compares to some baseline.

To improve this performance, we may consider experimenting with some of the different approaches to attribute selection as described later in section 3.2. The accuracy of the SVM classifier should also be assessed on different sized training sets.

5 Conclusions

The payoff of this analysis, in part, is the predictor itself which allows us to predict a winning or losing outcome, given a campaign speech, with a degree of accuracy. The other part lies in the insights and understanding gained through examining our most valuable attributes.

Upon further examination of the top 50 words with high importance, obtained in section 3.2, I was able to observe the following, recurring themes and connotations:

- **Economics:** spending, cost, raise, wealth, buy, rates,…
- **Joy/Freedom:** greater, laughter, choices, achieve,…
- **Unity:** keep, members, nation's,…
- **Fear/Aggression:** fight, risk, terror,…

Note that these categories are only based on observation and may be subject to some bias. However, assuming my approach is reasonable, the results appear to demonstrate that words associated with finance, unity, and fear resonate with voters and are connected to election results. In preparing speeches, candidates may want to consider that winning outcomes have a stronger relationship to words with a more positive connotation than words with a more negative connotation.

With any conclusions drawn, it is important to consider the impact of time on the robustness of the results. Where more fair/just language might resonate more with voters during one election year, a voice of strength/authority might garner more support another year based on current events. Ultimately, there is more to consider beyond word choice and frequency alone when it comes to who wins or loses. As such, current results may be subject to change depending on what years are taken into consideration. These insights do, however, provide a better understanding of this system and depict a trend of generalized voter behavior.