

**Campaign Finance**

*Using Data Analysis and Machine Learning*

*to study finances in*

*US political campaigns*

IST 707 Final Project

Michael Justis

Timothy Fajembola

Nick Yurashku

**Introduction:**

"Elections, by their nature, deviate from the concept of "Black Swan" events\*. According to Nassim Nicholas Taleb, the proponent of the Black Swan theory, an event with a probability of 50/50 does not fit within this classification (La Roche, 2016). Consequently, we may contend that if elections inherently avoid categorization as Black Swan events, they could conceivably be predictable - a state perhaps better designated as 'Gray Swan' phenomena.

This proposition implies that in the absence of stationarity and ergodicity, a system may yet be predictable, contingent upon the application of suitable features in machine learning methodologies. Additionally, if significant factors like campaign spending are included within these features, it could empower a data scientist to ascertain the 'optimal' or 'ideal bet size' that a campaign manager should dedicate to an election campaign to ensure victory.

In this study, we explore the influence of campaign finance on election outcomes through the application of machine learning algorithms. Our analysis transcends the boundaries of simple exploratory data analysis. We critically assess which models demonstrate the highest efficacy for these predictive tasks. Ultimately, we employ the predicted probability outcomes from the best-performing algorithms to establish the Kelly criterion, thereby determining the optimal campaign budget allocation

**Analysis and Models:**

**About the Data:**

The data presented in the data set itself has a lot of dimensionality but does give incredible insight on how political campaigns allocate their money. Down below in Table 1.1, it can be seen the type of data that was provided.

*Table 1.1*

| **Tag** | **Field Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| lin\_ima | Link to list of reports for the committee | URI | List of all disclosure filings for this committee |
| can\_id | Candidate Id | Character |  |
| can\_nam | Candidate Name | Character | Name of candidate |
| can\_off | Candidate Office | Text | office abbreviation |
| can\_off\_sta | Candidate Office State | Character | Postal abbreviation for state |
| can\_off\_dis | Candidate Office District | Number | District number for House candidates |
| can\_par\_aff | Candidate Party Affiliation | Character | Party abbreviation |
|
|
| can\_inc\_cha\_ope\_sea | Candidate Incumbent Challenger Open Seat | Character | Incumbent or challenger or open seat indicator |
| can\_str1 | Candidate Street 1 | Text | Street address |
| can\_str2 | Candidate Street 2 | Text |  |
| can\_cit | Candidate City | Text | City of mailing address |
| can\_sta | Candidate State | Text | State of mailing address |
| can\_zip | Candidate Zip | Number | Zip code of mailing address |
| ind\_ite\_con | Individual Itemized Contribution | Currency | Sum of itemized contributions from individuals |
| ind\_uni\_con | Individual Unitemized Contribution | Currency | Sum of unitemized contributions from individuals |
| ind\_con | Individual Contribution | Currency | Total contributions from individuals |
| par\_com\_con | Party Committee Contribution | Currency | Contributions from party committees |
| oth\_com\_con | Other Committee Contribution | Currency | Contributions from other committees |
| can\_con | Candidate Contribution | Currency | Contributions from the candidate him(her)self |
| tot\_con | Total Contribution | Currency | Total contributions |
| tra\_fro\_oth\_aut\_com | Transfer from other Authorized Committee | Currency |  |
| can\_loa | Candidate Loan | Currency | Loans received from the candidate |
| oth\_loa | Other Loan | Currency | Other loans |
| tot\_loa | Total Loan | Currency |  |
| off\_to\_ope\_exp | Offsets to Operating Expenditure | Currency |  |
| off\_to\_fun | Offsets to Fundraising | Currency |  |
| off\_to\_leg\_acc | Offsets to Legal Accounting | Currency |  |
| oth\_rec | Other Receipts | Currency |  |
| tot\_rec | Total Receipts | Currency |  |
| ope\_exp | Operating Expenditure | Currency |  |
| exe\_leg\_acc\_dis | Exempt Legal Accounting Disbursement | Currency |  |
| fun\_dis | Fundraising Disbursement | Currency |  |
| tra\_to\_oth\_aut\_com | Transfer to Other Authorized Committee | Currency |  |
| can\_loa\_rep | Candidate Loan Repayment | Currency | Loan repayments to the candidate |
| oth\_loa\_rep | Other Loan Repayment | Currency | Loan repayments to banks or others |
| tot\_loa\_rep | Total Loan Repayment | Currency |  |
| ind\_ref | Individual Refund | Currency | Contribution refunds made to individuals |
| par\_com\_ref | Party Committee Refund | Currency | Contribution refunds made to parties |
| oth\_com\_ref | Other Committee Refund | Currency | Contribution refunds made to other committees |
| tot\_con\_ref | Total Contribution Refund | Currency |  |
| oth\_dis | Other Disbursement | Currency |  |
| tot\_dis | Total Disbursement | Currency |  |
| cas\_on\_han\_beg\_of\_per | Cash on Hand Beginning of Period | Currency |  |
| cas\_on\_han\_clo\_of\_per | Cash on Hand Close of Period | Currency |  |
| net\_con | Net Contribution | Currency |  |
| net\_ope\_exp | Net Operating Expenditure | Currency |  |
| deb\_owe\_by\_com | Debt Owed by Committee | Currency |  |
| deb\_owe\_to\_com | Debt Owed to Committee | Currency |  |
| cov\_sta\_dat | Coverage Start Date | Date | MM/DD/YYYY |
| cov\_end\_dat | Coverage End Date | Date | MM/DD/YYYY |

As displayed in the table above, there are a total of 51 attributes and 1800 observations. In order to reduce the dimensionality of the data set, some columns were removed in order to have a better understanding of the relationship between spending and winners since that is the goal of this study.

After the reduction in dimensionality, that left the data set with just 22 attributes instead of 51 - which is a 57% reduction. Below in Table 1.2, the remaining data columns were left alone for further analysis:

*Table 1.2*

'data.frame': 1814 obs. of 22 variables:

$ X : int 1 2 3 4 5 6 7 8 9 10 ...

$ can\_off : chr "H" "H" "H" "H" ...

$ can\_off\_sta : chr "GA" "PA" "FL" "MT" ...

$ can\_par\_aff : chr "REP" "DEM" "REP" "REP" ...

$ can\_inc\_cha\_ope\_sea: chr "INCUMBENT" "CHALLENGER" "OPEN" "INCUMBENT" ...

$ ind\_ite\_con : num 554305 1042280 529030 2479616 746234 ...

$ ind\_uni\_con : num 46970 72431 13075 1837715 150890 ...

$ ind\_con : num 601275 1114711 542105 4317332 897124 ...

$ par\_com\_con : num 0 0 0 3545 0 ...

$ oth\_com\_con : num 473675 302834 106050 660039 308740 ...

$ can\_con : num 0 0 2700 0 0 0 0 0 0 0 ...

$ tot\_con : num 1074950 1417545 650855 4980915 1205864 ...

$ tra\_fro\_oth\_aut\_com: num 17710 0 0 136894 0 ...

$ can\_loa : num 0 0 60000 0 0 0 0 0 0 0 ...

$ oth\_loa : num 0 0 0 0 0 0 0 0 0 0 ...

$ tot\_loa : num 0 0 60000 0 0 0 0 0 0 0 ...

$ oth\_rec : num 0 0 0 55910 0 ...

$ ope\_exp : num 908519 1300558 656643 5073110 953437 ...

$ tra\_to\_oth\_aut\_com : num 0 0 0 0 0 ...

$ net\_con : num 1074950 1406719 650855 4938944 1197677 ...

$ net\_ope\_exp : num 907156 1298832 656210 5055942 949489 ...

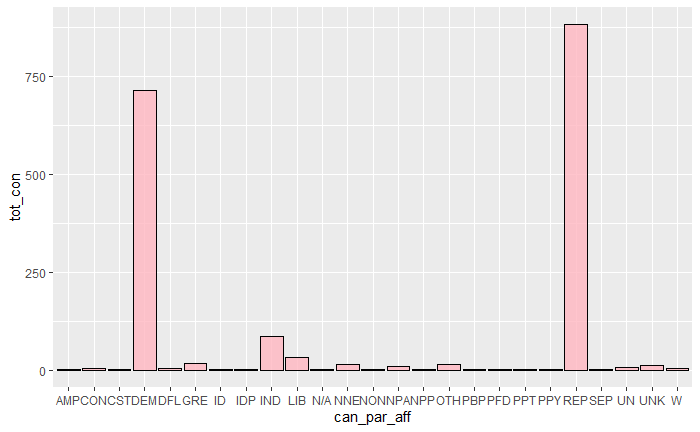
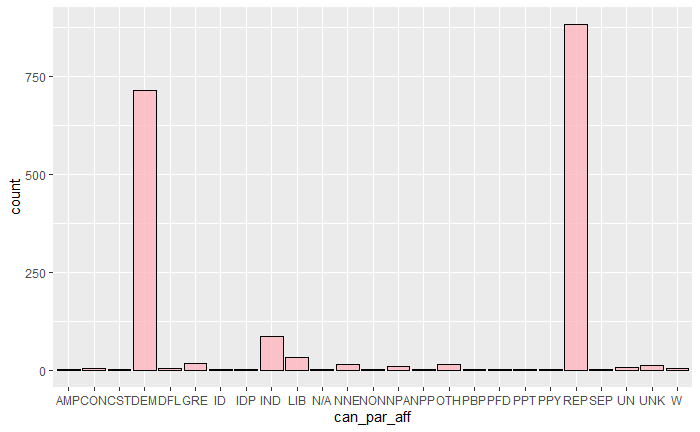
$ winner : chr "Y" "Y" "Y" "Y" ...

In addition to reducing dimensionality, some attributes also had their data types changed from character data types to numerical. This can be seen in the attached RMD code.

**Exploratory Data Analysis and Data Visualization:**

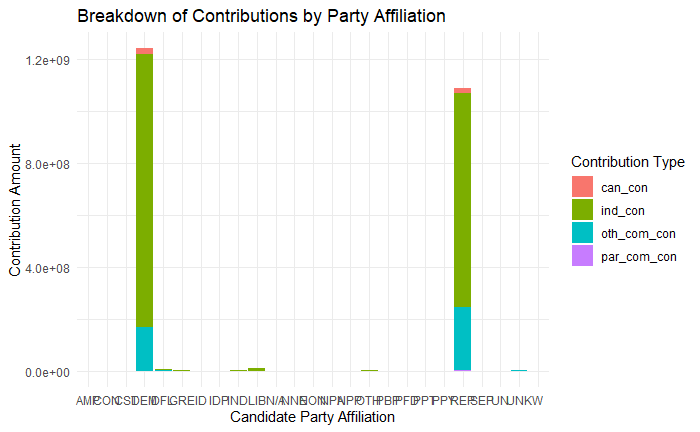
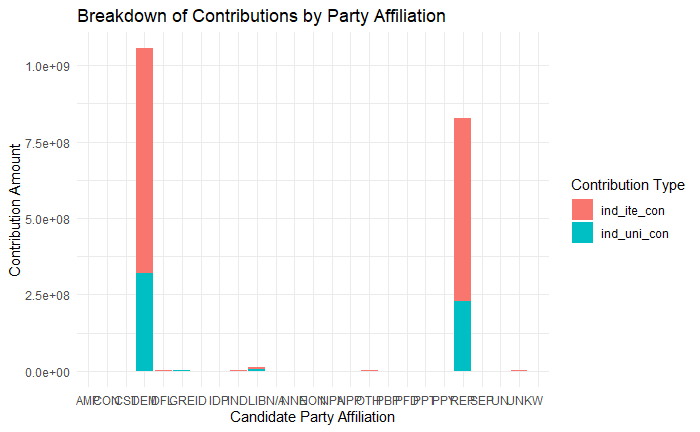
Preliminary data analysis of the dataset reveals intriguing insights about the candidates and patterns of political spending in the United States:

*Political Parties in Data Set* *Who gets the most contributions?*

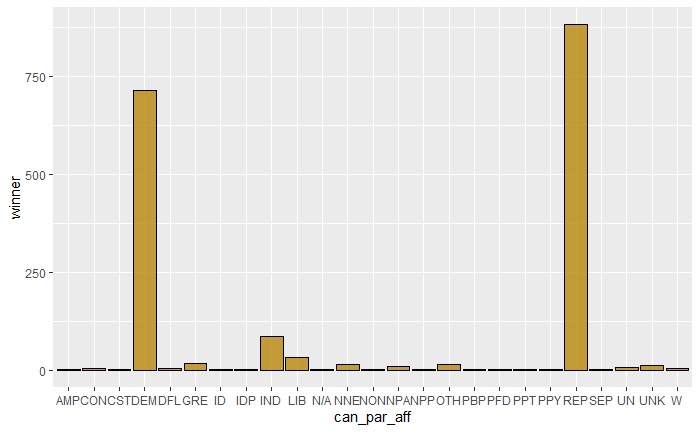
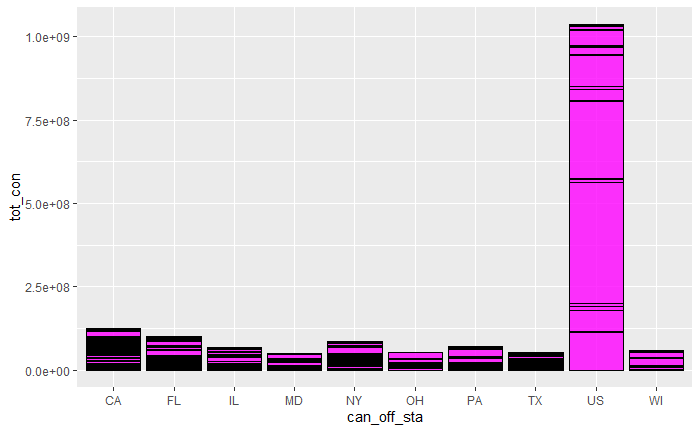


The dataset comprises various political parties, yet the majority are Republicans, Democrats, or Independents. A similar trend extends to campaign contributions, with Republicans leading, followed by Democrats, and Independents in third place. These findings imply that any parties other than these main three are likely to face substantial competition.

Subsequent analysis uncovers the primary sources of contributions for each party. Potential sources include candidate contributions, individual contributions, Super PACs, and party contributions. Predominantly, funds originate from Super PACs and independent donors. The majority of these independent donations are itemized contributions, indicating that wealthy donors provide most of the support.



Further investigation into expense and win ratios offers a unique perspective. Nationwide elections in the U.S. tend to be the most expensive, as depicted in the left-hand graph. Excluding nationwide elections, the most costly states to campaign in are California, Florida, New York, Illinois, Pennsylvania, Texas, and Wisconsin. An examination of the win ratios depicted in the right-hand graph shows that Republicans maintain the highest success rate.



**Models:**

**Model 1: Association Rule Mining (ARM)**

To discover patterns and associations between attributes in the history of the campaign's dataset, a data mining algorithm called association rule mining was employed. The goal of using association rule mining on the bank's dataset is to uncover hidden patterns, correlations, and associations within the data. The Apriori algorithm, implemented in R Studio, was used to determine the likelihood of one item occurring given the presence of another. This process is discussed in detail below.

The association rule mining algorithm (Apriori) begins by generating an itemset of the most frequent subsets, which enhances the algorithm's efficiency. This is illustrated in the "Results" section. After examining the most frequent rule, the algorithm employs support and confidence measures to ascertain the rule's credibility. Support is the measure of how many attributes contain the variable sets, while confidence is the proportion of attributes containing both variables.

**Model 2: Naive Bayes**

The Naïve Bayes algorithm is a probabilistic classifier that relies on Bayes' theorem and makes the assumption of independence among predictors. Despite its simplicity, it is widely used due to its surprising efficacy, especially in text mining and spam filtering tasks. In this analysis, it was sought to find probabilistic relationships between the campaign results and spending, using the Naïve Bayes algorithm to predict the winners and losers. The model was trained using 70% of the data and the remaining was reserved to the test data. The model was implemented using the naiveBayes() function from the 'caret' package in R.

**Model 3: K-means Nearest Neighbors (kNN)**

The kNN algorithm operates on the principle of proximity, classifying unseen samples based on the labels of their nearest neighbors in the feature space. Despite the potential computational burden, the popularity and simplicity of the kNN algorithm warranted its inclusion in this experiment. This analysis used a k-fold classification strategy to find the optimal amount of k-folds which came out to 11. The dataset also had to be changed to be fully numeric using hot encoding which added a lot of dimensionality to the data set but still gave promising results.

**Model 4: Support Vector Machine (SVM)**

The SVM is a more complex model that employs statistical learning principles to find the optimal hyperplane that best separates different classes. The strength of SVM lies in its capacity to handle high-dimensional data effectively, making it well-suited for classification tasks with a lot of numeric values.

**Model 5: Decision Trees**

The decision tree algorithm, by its nature, is inherently interpretable thanks to its tree-like structure of decision rules. When trained on the provided dataset, this model will attempt to classify pixels based on a series of binary decisions. The goal of this classification process is to correctly identify the elected winners represented in the data set, thus categorizing new data into appropriate classes.

The decision tree model was implemented using the rpart() function from the 'caret' package in R.

**Model 6: Random Forest**

The Random Forest algorithm is one of the most robust and complex models utilized in this analysis. Simply put, a Random Forest is an ensemble of decision trees, with each tree's prediction contributing to the final decision. This approach helps to overcome the overfitting problem common in individual decision trees.

**Kelly Criterion:**

The Kelly Criterion is a mathematical formula conceived by mathematician John Kelly while at Bell Labs. This formula is commonly employed in any context requiring the allocation of financial wagers based on probabilities. For instance, a political analyst who has estimated the likelihood of a candidate's victory can utilize the Kelly Criterion to determine the optimal allocation of campaign funds. This aids in strategically distributing resources, thus mitigating the risk of exhausting campaign finances and the subsequent need for additional capital.

The Kelly Fraction can be calculated by the following formula:

KF = (bp - q) / b

b - the odds of win on a particular wager

p - probability of winning

q - the probability of losing

**Results:**

**Model 1: Association Rule Mining (ARM)**

Filtering the rules for a support of .1 and confidence of .8, the following top 3 rules for lift were:

| **lhs** | **rhs** | **support** | **confidence** |
| --- | --- | --- | --- |
| {par\_com\_con=None, ope\_exp=low, net\_con=low} | {net\_ope\_exp=low} | 0.1085998 | 1 |
| {par\_com\_con=None, oth\_com\_con=None, ope\_exp=low, net\_con=low} | {net\_ope\_exp=low} | 0.1047409 | 1 |
| {par\_com\_con=None, oth\_rec=None, ope\_exp=low, net\_con=low} | {net\_ope\_exp=low} | 0.1014333 | 1 |

For all the lift rules, lift is only at 1 which shows that these rules actually are independent from one another. In this scenario, this means that x occurs as often as y. So these rules will not tell the observer anything interesting.

Confidence is another story and actually reveals some information. As a reminder, the confidence of an association rules measures how often Y is found in transactions that contain X.

The following rules for top 3 rules for confidence were:

| **lhs** | **rhs** | **support** | **confidence** | **coverage** | **lift** | **count** |
| --- | --- | --- | --- | --- | --- | --- |
| {can\_loa=low} | {winner=N} | 0.1174201 | 1 | 0.1174201 | 1.350707 | 213 |
| {ind\_ite\_con=None} | {winner=N} | 0.1345094 | 1 | 0.1345094 | 1.350707 | 244 |
| {net\_ope\_exp=low} | {winner=N} | 0.1493936 | 1 | 0.1493936 | 1.350707 | 271 |

1. *{can\_loa=low} => {winner=N}*

This rule suggests that if the loan amount taken by the candidate is low (can\_loa=low), then it's likely the candidate does not win the election (winner=N). The confidence of this rule is 0.1174201, which means this rule is correct about 11.7% of the time. The Lift of this rule is 1.350707, suggesting that the rule is 1.35 times more likely to be correct than a random assignment.

2. *{ind\_ite\_con=None} => {winner=N}*

This rule suggests that if there are no itemized individual contributions (ind\_ite\_con=None), then it's likely the candidate does not win the election (winner=N). The confidence of this rule is 0.1345094, which means this rule is correct about 13.4% of the time. The Lift of this rule is 1.350707, suggesting that the rule is 1.35 times more likely to be correct than a random assignment.

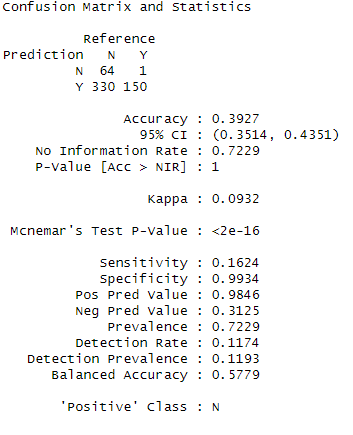
3. *{net\_ope\_exp=low} => {winner=N}*

This rule suggests that if the net operational expenditure is low (net\_ope\_exp=low), then it's likely the candidate does not win the election (winner=N). The confidence of this rule is 0.1493936, which means this rule is correct about 14.9% of the time. The Lift of this rule is 1.350707, suggesting that the rule is 1.35 times more likely to be correct than a random assignment.

These rules suggest a pattern in which a low spending candidate or low contribution receiving candidate often does not lead to not winning the election. The higher the confidence, the more frequently the rule has been found to be true. The Lift values greater than 1 suggest these rules are significant, as they are more likely than chance.

**Model 2: Naive Bayes**

Running the Naive Bayes algorithm gave the following results:



The confusion matrix that the naive bayes algorithm output and the statistics provided an overview of the performance of the model.

The matrix shows the following:

*True Negative (N, N)*: The model correctly predicted 64 times that the candidate would not win.

*False Negative (Y, N)*: The model incorrectly predicted 1 time that a winner would not win.

*False Positive (N, Y)*: The model incorrectly predicted 330 times that a losing candidate would win.

*True Positive (Y, Y)*: The model correctly predicted 150 times that the candidate would win.

*Accuracy* is the proportion of true results. Here, the accuracy is 0.3927, or about 39.27%. This means that the model correctly predicts the outcome about 39.27% of the time.

*The Kappa statistic* is a measure of how much better the predictions of the model are compared to predictions made by chance. Here, the kappa is 0.0932, which is quite low - please note that a perfect model has a kappa of 1. This is indicating that the model is not much better than a random model.

*Sensitivity* is the proportion of actual positives that are correctly identified. In this case, the sensitivity is 0.1624, meaning that the model identifies about 16.24% of the winning candidates correctly.

*Specificity* measures the proportion of actual negatives that are correctly identified. Here, the specificity is 0.9934, suggesting the model is very good at identifying candidates that won't win.

*The Positive Predictive Value* is the proportion of positive cases that were correctly identified by the model. In this case, it's 0.9846 or 98.46% of candidates predicted not to win, did indeed not win. However, the *Negative Predictive Value* is low (0.3125 or 31.25%), meaning that of all the candidates that were predicted to win, only 31.25% actually won.

*The No Information Rate* which is the accuracy that can be achieved without a model by always guessing the most frequent category. In this case, guessing the candidate does not win all the time would be right 72.29% of the time.

*The p-value* associated with the accuracy being greater than the no information rate is 1, which indicates no evidence of the model doing better than the no information rate.

Overall, this model seems to be better at predicting losers than predicting winners. The low accuracy, kappa, and high p-value suggest that there might be a need to improve this model or use a different model.

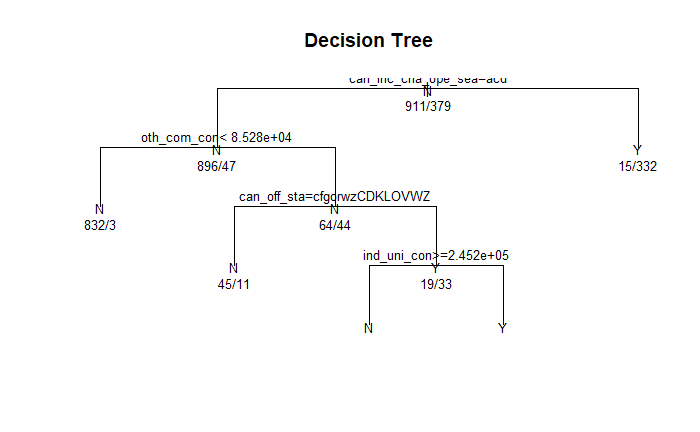
**Model 3: K-means Nearest Neighbors (kNN)**

Upon scaling the data and applying one-hot encoding to transform it into a purely numeric format, a k-fold validation strategy determined the optimal k value to be 11. Using this configuration of 11 clusters, the model yielded a remarkable output accuracy of 0.946789. This accuracy metric is computed as the ratio of correct predictions to the total number of predictions made. An accuracy of 0.946789 implies that the model accurately predicted the outcome approximately 94.68% of the time. While this is a generally impressive result, it may not be satisfactory if the cost associated with false negatives is substantial. For instance, in situations where financial stakes and campaign contributions are significant, a high false negative rate could have serious implications, despite being tolerable in other contexts.

**Model 4: Support Vector Machine (SVM)**

The SVM model displayed the most inferior performance among all the models, attributable to its inherent sensitivity to parameter selection. For example, our dataset displayed a significant disparity between the winning and losing candidates, affecting the model's performance. Consequently, the model yielded a disappointingly low mean accuracy of .0018, translating to a mere 0.18% predictive accuracy. Despite attempts to optimize the model, such as switching the kernel from "radial" to "linear" or modifying the proportions of the training and testing sets, only negligible enhancements in mean accuracy were observed. Therefore, we can conclusively assert that the SVM model was not ideally suited for this particular experiment.

**Model 5: Decision Trees**

The decision tree algorithm did an amazing job as this algorithm works well with both numerical and categorical data. Below in the graph, it can be seen how the algorithm decided to split the data for decision making: 

With stellar results, the algorithm predicted with high accuracy. The output showed the following:

The root of the tree (node 1) includes all 1290 observations. The base prediction (yval) is 'N', because the majority (about 70.6%) of the candidates did not win. The first split (node 2) is based on the variable 'can\_inc\_cha\_ope\_sea (this variable shows if the candidate is incumbent or not)'. If a candidate is an 'INCUMBENT', the model predicts a win (node 3) with a very high probability (95.7%). Showing that being an incumbent politician means that you have the upper hand.

Furthermore, if a candidate is a 'CHALLENGER', 'OPEN', or 'UNKNOWN', the model makes further splits based on the 'oth\_com\_con' and 'can\_off\_sta' variables (nodes 4 and 5). Meaning that there is a distinction if the candidate is receiving funds from super pacs (this probably gives a candidate a lot of power). For 'oth\_com\_con' < 85283.5 (node 4), the model predicts 'N' (not win) with a very high probability (99.6%).

For 'oth\_com\_con' >= 85283.5 (node 5), the model makes further decisions based on the 'can\_off\_sta' variable. If 'can\_off\_sta' belongs to one of these states: 'AR,CA,CO,IA,IN,MD,MN,MT,NC,NY,OH,PA,US,UT,WV', the model predicts 'N' (node 10), and if 'can\_off\_sta' belongs to the following states: 'AZ,DE,FL,GA,IL,KS,KY,LA,MI,NE,NH,NJ,NV,TN,TX,VA,WA,WI,WY', the model makes further decisions based on the 'ind\_uni\_con' variable (node 11).

If 'ind\_uni\_con' >= 245244.4 (node 22), the model predicts 'N', and if 'ind\_uni\_con' < 245244.4 (node 23), the model predicts 'Y' (win) with a high probability (74.4%).

The accuracy of this model on the test set is about 91.9%, which means the model correctly predicted the winner for about 91.9% of the cases in the test set.

**Model 6: Random Forest**

The Random Forest Model also showed very promising results just like the previous model. The output shows a lot of details that would be too large to paste in this report so please reference the attached R code. The explanation of the output is as follows:

*Type of random forest: classification* - meaning that a classification model was used and the target variable, ‘winner’, is categorical.

*Number of trees: 500*. This is the number of trees that were used to vote for the class in classification or averaged in regression.

*No. of variables tried at each split*: 3 - At each split in each decision tree, 3 variables were randomly selected as candidates for splitting.

*OOB estimate of error rate*: 4.73% - This is the Out-of-bag error estimate, which is a method of measuring the prediction error of random forests. This error rate suggests that the model is quite good, as the error rate is very low.

The confusion matrix provides a summary of the model's performance. It's a 2x2 matrix: "Y" and "N". The model has classified 882 instances correctly as 'N' and 29 incorrectly as 'Y'. Similarly, it has classified 347 instances correctly as 'Y' and 32 incorrectly as 'N'. The class error is calculated as the number of wrong predictions divided by the total predictions for each class.

Accuracy: 0.964401294498382 - This is the overall accuracy of the model on the test data. It is very high, which indicates that the model did a great job in classifying the winner.

**Kelly Criterion:**

Before utilizing the Kelly Criterion to compute the optimal allocation of campaign funds per candidate, it's crucial to note that the probabilities derived from our decision tree model may not perfectly mirror real-world probabilities. Election prediction, being more of an art than an exact science, involves numerous variables beyond those included in our dataset. The Kelly Criterion operates under the assumption of exact, error-free probabilities akin to those in a dice game, where outcomes strictly adhere to mathematical properties. In the realm of politics, variables such as the prevailing political climate, candidates' policy proposals, public opinion, and other influential factors might not be accurately represented in our data.

Thus, we recommend using a fraction of the calculated Kelly value based on the degree of risk a campaign manager is willing to assume. Given that the decision tree model assigns specific probabilities to each node (or scenario), the Kelly Fraction can be utilized. The tree outlines the following nodes and their associated probabilities:

**Node 3:** p = 0.956772334, at the top of the tree checks to see if the candidate is a challenger or incumbent. If the candidate is an incumbent then the probability of them remaining in office is a very high at 95%. According to Kelly, this would be a candidate worth supporting with a high allocation of funds:

b = 1 / 0.956772334 = 1.045228171

q = 1 - p = 1 - 0.956772334 = 0.043227666

Kelly fraction = ((0.956772334 \* 1.045228171) - 0.043227666) / 1.045228171 = 0.871.

This fraction is saying that an optimal bet of 87% of the campaign funds can be wagered for this election.

**Node 4**: p = 0.003592814, if oth\_com\_con (funds from super Pacs) < $85,283.50, the candidate not receiving super PAC support is a bad sign and the kelly formula comes out negative, meaning do not wager on this candidate:

b = 1 / 0.003592814 = 278.2773365

q = 1 - p = 1 - 0.003592814 = 0.996407186

Kelly fraction = ((0.003592814 \* 278.2773365) - 0.996407186) / 278.2773365 = -0.996

**Node 10**: p = 0.196428571, now if the candidate did indeed receive more than $85K from super PACS, then further split asks if the candidate belongs to one of these states: AR,CA,CO,IA,IN,MD,MN,MT,NC,NY,OH,PA,US,UT,WV. The odds of the candidate winning drops and so does the kelly formula:

b = 1 / 0.196428571 = 5.090909091

q = 1 - p = 1 - 0.196428571 = 0.803571429

Kelly fraction = ((0.196428571 \* 5.090909091) - 0.803571429) / 5.090909091 = -0.032

**Node 22**: p = 0.307692308. If the candidate instead is at one of these states: AZ,DE,FL,GA,IL,KS,KY,LA,MI,NE,NH,NJ,NV,TN,TX,VA,WA,WI,WY. Then the odds improve but the Kelly formula still is not positive because further splitting is needed:

b = 1 / 0.307692308 = 3.25

q = 1 - p = 1 - 0.307692308 = 0.692307692

Kelly fraction = ((0.307692308 \* 3.25) - 0.692307692) / 3.25 = 0

**Node 23**: p = 0.743589744. The last node is the determination on how much contributions from individual donors did the candidate get. If the amount is below $245k, then the candidates win has low probability. Otherwise, there is a 74% chance that candidate wins and the Kelly bet would be:

b = 1 / 0.743589744 = 1.344827586

q = 1 - p = 1 - 0.743589744 = 0.256410256

Kelly fraction = ((0.743589744 \* 1.344827586) - 0.256410256) / 1.344827586 = 0.3

This bet says that a wager of 30% of the funds can be put to this candidate. Which is a lot lower than from what we saw at Node 3. This is because this is what the campaign manager should be willing to spend given the probabilities on hand.

**Conclusions:**

This report demonstrates that machine learning algorithms can distill the rich history of U.S. election finances and outcomes into quantifiable probabilities. These algorithms, including Naive Bayes, kNN, Support Vector Machines, Decision Trees, and Random Forest models, can offer valuable insights to a range of stakeholders, from political analysts and budget analysts to candidates and campaign managers.

The primary goal of this report was to highlight the presence of predictability in election results and illustrate how these probabilities can be utilized in well-established formulas, such as the Kelly Criterion. Renowned for its use in optimizing betting strategies in probability-based games, the Kelly Criterion can determine the ideal betting amount that maximizes returns without the risk of going bust.

By integrating the Kelly Criterion with the outcomes of machine learning algorithms, we can ascertain, at any point in a decision tree, the appropriate allocation of funds a political party should commit to a specific candidate. However, it's crucial to note that while this study can be replicated with more robust data or refined algorithms, the raw output from the Kelly Criterion should not be used directly for budgeting decisions.

The Kelly Criterion presumes static probabilities, which do not truly reflect the dynamism of real-world election scenarios. Thus, we recommend utilizing a fraction of the calculated Kelly amount. Future studies could benefit from exploring additional algorithms, incorporating Monte Carlo simulations to estimate the variable probabilities a candidate may encounter.

In conclusion, this report refutes the notion of election cycles being "Black Swan" events and underscores the valuable insights that machine learning algorithms can offer to enhance election strategies.

**References:**

La Roche ,Julia (2016, November 11). Nassim Taleb: Donald Trump's election win was no 'black swan'. Yahoo! News. Retrieved from <https://www.yahoo.com/news/nassim-taleb-donald-trumps-election-win-was-no-black-swan-191857463.html?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAAHqhinPCFe6PvoKz23ElGdsY98NM2Iy5wwq5-6g5GBTeJS76mY4DRbe-w0oa20lOgSQSDsTjm014irASGWvOlHqysHdPV3-s-BXhJsLOm75hmj9dDQMtWv7P5JmE1maNph5FQssfu9tFxYGEuU4jWRKJlG4svgmpkI8VZUAbahYv>

Taleb, N. N. (2007). The Black Swan: The Impact of the Highly Improbable. Random House.