Buying Elections: Predicting the 2020 U.S. Senate General Elections using Artificial Intelligence

Joel Raymond Day
joelday.business@gmail.com

ABSTRACT

The best election forecasting models in recent decades are trained on a combination of polling data and a hand-selected bundle of supplementary features. While traditional polling processes are being replaced by methods that utilize social media information, the supplementary bundle of features remains necessary for optimal performance – this project provides that bundle. Results have demonstrated the Random Forest model, excluding polling data and knowledge of the candidates' state, can predict which candidates will obtain at least 50% vote share with an F1 score of 81%.

1 INTRODUCTION

Two interacting forces are driving the growth of the data industry. People are becoming increasingly willing to sacrifice privacy for convenience, and organizations are becoming increasingly incentivized to harvest and distribute data for financial gain. Consequently, data is stored away faster than we can consume it, creating an abundance of unused data hiding potentially useful insights. Election Forecasting methods have not gone untouched. Utilizing all this data will certainly increase the accuracy of forecasts as well as change how forecasts are made.

The top performing models in recent years combine polling data with a bundle of other features (e.g., economic variables, demographic features, candidate incumbency), but social media provides an alternative to traditional polling methods that is both cheaper and can reach a larger audience. Although filtering and tokenizing social media data has presented an abundance of problems, recent work has shown that predicting elections using twitter data is possible, but only when combined with a supplementary bundle of features – no different than polling data. If the resulting models are consistently improved when other features are added, polling and social media data must be unable to capture the whole story. This project's model attempts to explain the vote share variance polling and social media models can't. This project is the first step towards an automated U.S. election forecasting model.

2 MODEL FEATURES

Along with the age, incumbency, and party of the candidate, the following features will be used:

2.0.1 Contributions from Individuals and Multi-Candidate Committee Participation (FEC)

Strategic campaigning and the use of campaign funds can swing undecided voters and increase voter participation. A large budget enables more outreach and exposure both in person and online. Exposure theory proves that repeated exposure to a candidate will improve the impression of them (Oppenheimer et al., 1986). Only the contributions from individuals are used in the totals because only they can vote, and multi-candidate committees participation partially captures the support of PAC contributions. Contribution totals are calculated using data up to one month before the election.

2.0.2 Voter Education Inequality by State

Social demographics are also additive to election models (Myilvahanan et al., 2023). This project will focus on state education levels. For each state, the ratio of people who have a college degree to those who do not have a high school degree is calculated.

2.0.3 Economy Indicator - Unemployment

Both Kennedy et al. (2017) and Takashi (1981) warn about the poor performance of economic indicators, however, employment and inflation may be weak exceptions. Takashi found that employment was predictive, and Kennedy et al. found that inflation was predictive. I included unemployment because it varies more across the sample elections.

3 METHODOLOGY

I use data from the 2016 and 2018 elections to train the model and data from 2020 for testing. It is important to ensure the train/test split mimics the real-world forecasting period. The test data in this case should be more recent and independent from the training data to avoid bias.

Why did I choose this time span? Elections are held every other year (on even years), on each election year states are divided into the same 3 groups, and one of the groups does not participate - ensuring each state participates only 2 times every 3 election years (replacing 1 of their 2 senators at each election). The chosen time span captures one full 6-year senate cycle (3 elections) where all 100 senate seats are replaced.

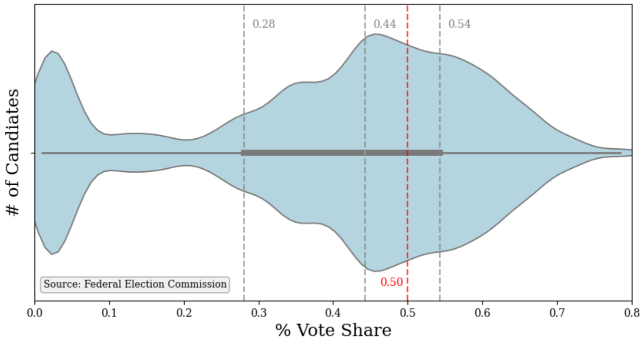
Google Collab was used to train and run python scripts for data aggregation, MySQL was used to store and query data for visualizations and deployment, and Jupyter-notebook was used to run the MySQL-related python scripts and interact with the local database. GitHub was used for storage and version control.

Regarding the dependent variable, vote share is preferred over binary values for the winner because it is more informative (Quek & Sances, 2015). A lot of people might be 60/40 or 80/20 in favor of a candidate, but casting a vote has the appearance of 100/0 - introducing bias to the model. In practice, a vote share of over 50% isn't needed because it is impossible to lose past at this threshold. This project is concerned with predicting who will win, so a 50% cutoff was chosen; In other words, (yes or no) will a candidate obtain 50% of their election's vote share?

Figure 3.1 demonstrates that (1) most elections are close and (2) the top two candidates get most of the votes. There is an equal amount of sample candidates between each of the three lines. The mean is 39%, despite a median of 44% due to the high concentration of vote share under 5%; this subgroup is comprised of primarily third place candidates and is a symptom of the two-party political system.

Figure 3.1

Distribution of Candidate Vote Shares in Quartiles



Source: Federal Election Commission

3.1 Data Acquisition

The data is pulled directly from the FEC, BLS, and Census Bureau's sites using their official APIs or web scraping from their websites. The results were web-scraped using the requests and BeautifulSoup libraries. Incumbency, party, vote share, and committee IDs were collected this way. Candidate age was then added manually through web searches. The finance data was pulled from the FEC's API (api.open.fec.gov). Employment data came from the Bureau of Labor Statistics' API (api.bls.gov), and the education data came from the Census Bureau's API (api.census.gov).

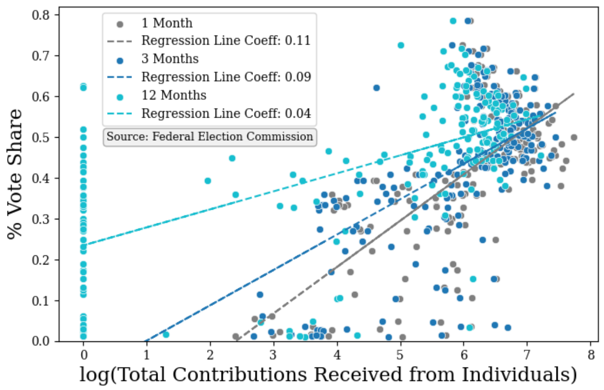
3.2 Exploratory Data Analysis

The following visuals demonstrate the relationship between independent and dependent variables, as well as inter-state differences. It is important to note that these visuals are derived from the sample data which thoughtfully excludes some candidates. The remaining candidates meet the following criteria: obtained more than 1% vote share, had their age and personal information easily accessible via google search, and has a Principal Campaign Committee with an official FEC ID (candidates are included even if their registered PCC reported no contributions).

Figure 3.2 visually demonstrates how the relationship between vote share and contributions received is positive and significant. Additionally, as the election approaches the relationship between these variables strengthens. Although they are all correlated with vote share, including all three lookback periods is not advised because they are all dependent on each other, so I use only the 1 month lookback period because it has the highest regression coefficient.

Figure 3.2

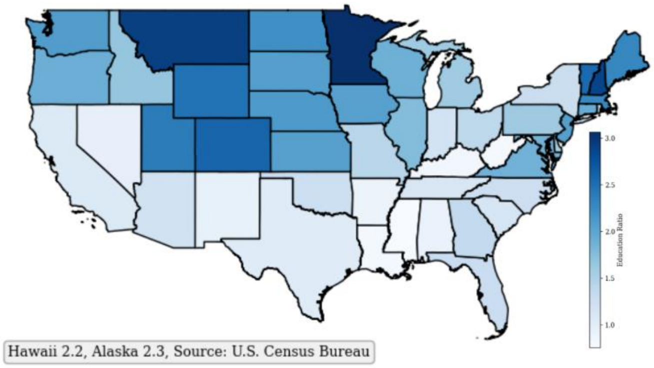
Vote Share vs. log (Total Contributions Received from Individuals 1, 3, & 12 Months Before the Election)



Source: Federal Election Commission

Figure 3.3

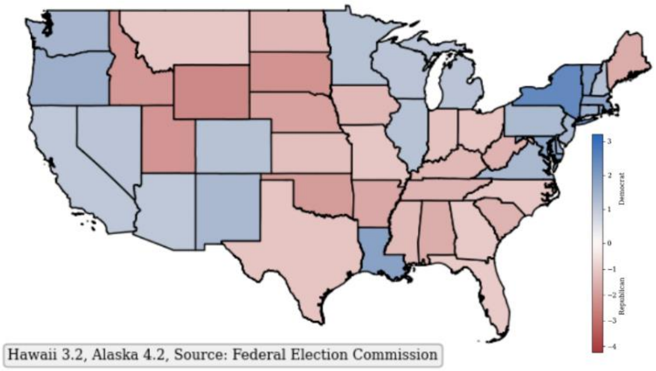
Ratio of Citizens with a Bachelor's Degree to Citizens with no High School Degree



Hawaii 2.2, Alaska 2.3, Source: U.S. Census Bureau

Figure 3.4

Ratio of Vote Share by Party



Hawaii 3.2, Alaska 4.2, Source: Federal Election Commission

Figures 3.3 and 3.4 show inter-state differences in education and party preferences respectively. Localized issues make close states more similar, creating regional differences in voting preferences and education accreditations that expand past state borders. The models are trained using these regional features and without knowledge of state borders. This generalizes the model and enables it to cast predictions in both state and federal elections.

3.3 Pre-Processing

The following pre-processing steps were performed in this order: change the data type of vote share to binary, calculate the log of contributions received, one-hot encode the only non-ordinal categorical variable (party), split the dataset into a test and a train, scale the numerical variables, and perform principal component analysis to remove multi-collinearity.

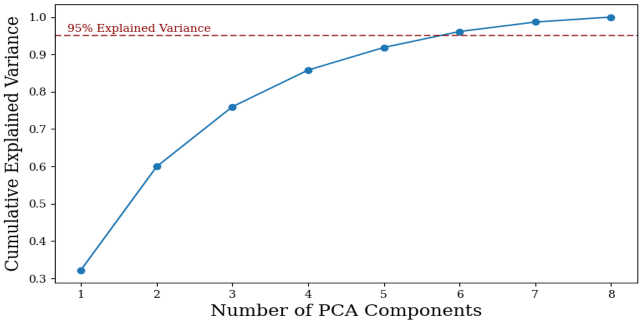
The pre-processing steps can be broken down into two parts, steps that introduce bias and steps that do not. It is imperative that there is no information leak from the test set into the training set. To prevent this, actions that introduce bias, such as scaling, must be performed after the test/train split – treating each data frame the same but separately. When scaling it is important to scale the test data using the mean and standard deviation of the train data. Keeping this in mind, I use (1) to manually scale each numerical feature and save the mean and standard deviation to easily scale user input data after deployment.

$$Scaled\,Value = \frac{(Original\,Value - Training\,Mean)}{Training\,Standard\,Deviation} \quad (1)$$

Removing linear dependence is the last pre-processing step. Using the correlation matrix, I found that multi-candidate committee participation and contributions received (.47), as well as unemployment and education (-.66), were loosely correlated, so I used PCA to remove linear dependence across all features. Figure 3.5 plots the cumulative amount of explained variance as the number of principal components increases. The goal is to balance over and under fitting by finding the number of components that explain the most variance without overfitting to the sample group. A good goal is to aim for around 95% cumulative explained variance- I chose 6 components.

Figure 3.5

(PCA) Explained Variance vs. Number of Components



4 MODELING

The 4 initial models were trained on the principal components derived in the preprocessing steps. In total 5 models were trained: a neural network (multilayer perceptron), random forest, gradient boosting, support vector machine (SVM), and afterwords, a weighted voting ensemble model.

Initially, I train each model using their default hyperparameters; then, I use cross validation and grid search to find the optimal parameters and update the pipeline with these new values. The optimized hyperparameters for each model are detailed in the bulleted list:

- Neural Network: activation ‘relu’, alpha = 0.003, hidden layer sizes = 1, solver ‘lbfgs’.
- Random Forest: max depth = 6, number of estimators = 60.
- Gradient Boosting: learning rate = 0.02, max depth = 4, number of estimators = 400.
- Support Vector Machine: C = 5, kernel ‘rgb’.

A couple things to note. The ‘number of estimators’ parameter for both the Random Forest and Gradient Boosting models is the number of individual models (forests) it will train, and the max depth is how many splits it will make before it settles on a prediction. Also, the ‘C’ value in the support vector machine model determines how it will balance finding a hyperplane with the largest minimum margin and separating as many instances as possible. C > 1 suggests prioritizing the correct classification at the expense of maximizing the minimum margin – the optimal value of C was determined to be 5.

An ensemble model was trained using the predictions from the 4 initial models. To find the optimal voting weights, I created a short python script that iterated through a custom list of voting weight combination, this allowed me to find the weights that resulted in the highest F1 score on the test set. The optimal weights are 5 Random Forest, 3 Neural Network, 3 Support Vector Machine, and 2 Gradient Boosting.

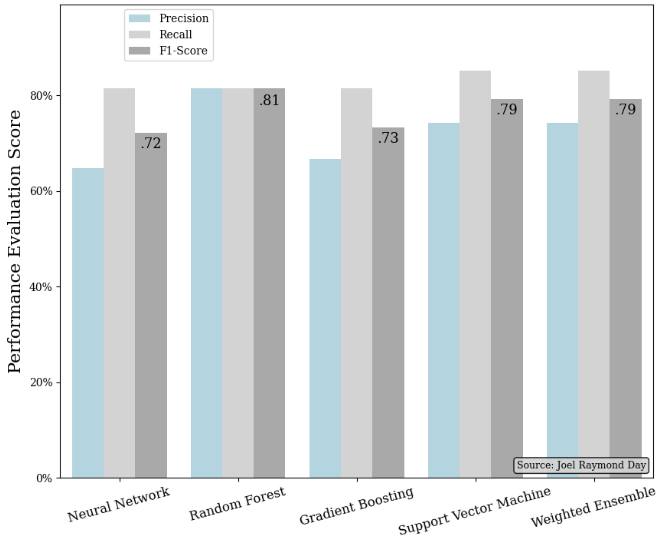
5 PERFORMANCE EVALUATION

The random forest model performed the best, it can predict which candidates will achieve a vote share over 50% with an F1 score of 81% - without knowledge of the candidate’s state. I use the F1-score because it is the balances both precision and recall; I weigh them equally at 50 recall /50 precision because false negatives and false positives are equally harmful.

This project establishes the positive correlation between financial contributions from individuals and vote share. To offer a comparison, Kennedy et al. (2017) presented a model with 80% accuracy, which was increased to more than 90% when able to incorporate polls. Inspired by their success, the next stage of this project is adding social media activity to the training data to see if the features in this model can adequately fill in the gaps by explaining what the election current models can’t.

Figure 5.1

Precision, Recall, & F1 Scores for all Models



REFERENCES

Acharjee, P. B., Magadum, A. A., Thejovathi, M., Jain, R., Umarani, K., & Nishant, N. (2023). An innovative method for election prediction using hybrid A-BiCNN-RNN approach. *2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS)*, 765– 770. https://doi.org/10.1109/ICACRS58579.2023.10404211

Brito, K., & Adeodato, P. J. L. (2023). Machine learning for predicting elections in Latin America based on social media engagement and polls. *Government Information Quarterly*, *40*(1), 101782. https://doi.org/10.1016/j.giq.2022.101782

Kennedy, R., Wojcik, S., & Lazer, D. (2017). Improving election prediction internationally. *Science*, *355*(6324), 515– 520. https://doi.org/10.1126/science.aal2887

Myilvahanan, K., P, Y., Pasha, S., Ismail, M., & Tharun, V. (2023). A study on election prediction using machine learning techniques. *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, 1518–1520. https://doi.org/10.1109/ICAIS56108.2023.10073693

Oppenheimer, B. I., Stimson, J. A., & Waterman, R. W. (1986). Interpreting U.S. congressional elections: The exposure thesis. *Legislative Studies Quarterly*, *11*(2), 227–247. https://doi.org/10.2307/439877

Quek, K., & Sances, M. W. (2015). Closeness Counts: Increasing Precision and Reducing Errors in Mass Election Predictions. Political Analysis, 23(4), 518–533. http://www.jstor.org/stable/24573190

Takashi, I. (1981). Explaining and predicting Japanese general elections, 1960-1980. *Journal of Japanese Studies*, *7*(2), 285. https://doi.org/10.2307/132204