HIGH SCHOOL EDITION
Journal of Student Research
Volume 12 Issue 3 (2023)

Using Machine Learning Techniques to Predict United States House of Representatives Elections

Sanatan Mishra and John Lee

ABSTRACT

We predict the results of the United States House of Representatives elections using machine learing techniques. We started by collecting and preprocessing data on the partisan lean of districts, the state of the economy, the national political environment, candidate political stances, news headlines about each candidate in each race, and the past election results. Then, we selected, designed, and trained the models we would use to predict those election results. We used single-tier models that took either only news headline text data or only numerical data as inputs and two-tier models that used both news headlines and numerical data. Our best-performing model was a two-tier model with a GRU as the first tier followed by a Ridge Regressor as the second tier, with a root mean squared error of under 2 percentage points. The vote share predicted by our best model was within 2 percentage points of the actual observed vote share.

Introduction

Motivation

The House of Representatives is the lower house of the legislature of the United States of America. As its function is to legislate, its composition determines the direction the nation moves in for the next two years. So, it stands to reason that predicting the House of Representatives elections can help predict the United States' policy direction. Predicting the United States' policy direction can be useful for many groups and individuals like businesses, stock-holders, and taxpayers.

Literature Review

In the recent past, several researchers have used machine learning techniques to predict various polls and elections with reasonable accuracy.

Isotalo et. al used social media activity from Twitter on certain relevant hashtags to predict the polls using linear regression, and as they got an adjusted root mean squared error of 0.0058 percentage points, they proved that Twitter activity can be used to predict polls (and therefore, elections) [11].

In the past, many researchers have tried to predict elections using sentiment data from discussions on Twitter. The paper by Tsai et. al used a recursive neural tensor network (RNTN) to help predict the 2018 midterm elections in the United States [22]. The paper by Jose and Chooralil used a classifier ensemble with SentiWordnet, Naive Bayes, and hidden Markov models on similar data to predict election results in Delhi, India in 2015 [13]. Joseph used a decision tree to forecast the 2019 Parliamentary elections in India and produced a prediction result with around 97% accuracy [14].

A paper by Zolghadr et. al used artificial neural networks and support vector regression to predict United States presidential elections, but they used several independent variables in lieu of sentiment data. Specifically, they considered the number of terms the incumbent has been in office, personal income, the electoral vote of the incumbent in the

previous election, the votes for the incumbent party in the last Senate and House of Representatives elections, the president's approval rating, the unemployment rate, and the number of times the three-month GDP growth rate (an indicator of the health of the economy) went over 3.2% in the past four years [27]. Eventually, they chose the president's approval rating because it was the independent variable most correlated with their dependent variable of the number of votes the incumbent garnered [27].

In this paper, we first collect and pre-process a wide range of input data as explained in section **2**. Before feeding the data into ML models, we do some preliminary analysis to understand various trends and correlations, discussed in section 3. The models we used, the reasoning behind choosing them, and their advantages, disadvantages, and applicability are discussed in section 4. Our analysis of the results, metrics, and methodology for evaluation is discussed in section 5. Finally, the paper is summarized in section 6.

Methods

In the work presented in this paper, we started by collecting and preprocessing data on the partisan lean of districts, economics, the national political environment, candidate political stances, news headlines about each candidate in each race, and the past election results. Some of the inputs like national economic and political environment were chosen based on past research [27] while others like district partisan lean, candidate political stances, and news headlines were chosen based on conventional wisdom or our own intuition. In the next step of our research, we analyzed some of our variables to find which of them were correlated. Our variables could be sorted into two categories: text-based news headline data and other numerical data. So, we selected and designed the models we would use to predict the election results, which included single-tier models that focused either only on news headline text data or only on the other numerical data and two-tier models that used both news headline and other numerical data. Finally, we trained and tested those models, and analyzed their performance. The rest of this section goes through each of these steps in detail.

Data Sources and Pre-processing

We used data from all the districts in the State of California and the Commonwealth of Virginia from the 2018 US House of Representatives election cycle for this purpose. Our output variable is the final vote share garnered by the Democrats, which was obtained from Wikipedia [1] [2].

News Headline Sentiment Data

We used the news headline data to capture the sentiment of how the election is progressing according to the people who shape voters' opinions (specifically the media). We used all the headlines related to each candidate from the "News" tab of Google Search, published during the month leading up to the election (in total,756 headlines). We numerized and vectorized the data using the Keras Tokenizer [21]. This data is also referred to as "sentiment data" or "text data" in the rest of this paper.

Non-sentiment Data

While media opinion can be an important factor in predicting elections, many other factors, including the district's partisan lean, the state of the economy, the national political environment, and the candidates' stances on important topics, play a crucial role in determining the election outcome. We used tabular numerical data with those variables to augment our sentiment data. We refer to them as "other" data or "non-sentiment" data at various points in this paper.

District Partisan Lean

We used the composite vote share by party from each district, as given by Dave's Redistricting Application [6] [25]. The composite vote share of each party is the average vote share garnered by the party in a district across state office, senatorial, and presidential elections over a window of time [18]. This gives us the intrinsic partisan lean of the district, which is likely one of the most important factors determining the winning candidate.

Economic Data

We used Gross Domestic Product (from the St. Louis Federal Reserve [10]), Personal Consumption Expenditure (PCE) Inflation (from YCharts [23]), and Unemployment Rate (from YCharts [24]) as the state of the economy has historically affected voter behavior. We used the average of the Quarter 2 and Quarter 3 values of these metrics for the election year.

National Political Environment Data

We used the Generic Ballot percentage for each party on October 1 as given by poll aggregator FiveThirtyEight [15] as an indicator for the national political environment.

Candidate Stance Data

This data represents candidates' opinions and rhetoric on important issues that voters may care about (longstanding and recent issues, e.g. abortion, border security, taxation, and healthcare), picked from campaign websites, voting records, and Ballotpedia archives. We qualitatively assigned each candidate's stances a number between -6 and 6, with nonexistent Republican opponents to unopposed Democrats having a score of -6, Republicans with unknown stances a score of -5, firebrand Republicans having a score of -4, passionate Republicans having a score of -3, mainstream Republicans having a score of -2, moderate Republicans having a score of -1, moderate Democrats having a score of 1, mainstream Democrats having a score of 2, progressive Democrats having a score of 3, self-described socialist Democrats having a score of 4, Democrats with unknown stances a score of 5, and nonexistent Democratic opponents to unopposed Republicans having a score of 6.

Preprocessing

Data preprocessing was necessary because we needed to convert data from various sources and formats to one single format that could be easily fed into our models. For ease of providing input to the models, we put all of our variables in a Pandas DataFrame [29]. We used Scikit-learn to help us train all our models with the DataFrame [30]. We aggregated data from the above sources and formatted them into a table with a row for each race and a column for each of the data. In total, we had 54 rows and 16 columns. Essentially, our numerical data had 14 features (dimensions) across 54 data points (samples), while our text data was processed to 54 sequences (samples) of length 1185 (1185 dimensions).

Preliminary Analysis

In order to assess which input variables actually had an effect on our output variable of a party's final vote share, we performed some preliminary analyses.

The scatter plot in Fig 1, with each point in the plot representing a district, shows the Democratic candidates' composite vote share over 4 years on the x-axis and the Republican candidates' composite vote share over 4 years on the y-axis. There is a strong negative correlation between the two, suggesting that there were not very many third-party candidates who had much impact on the races. So, we decided that we could exclude the third-party candidates' vote share as irrelevant to this analysis.

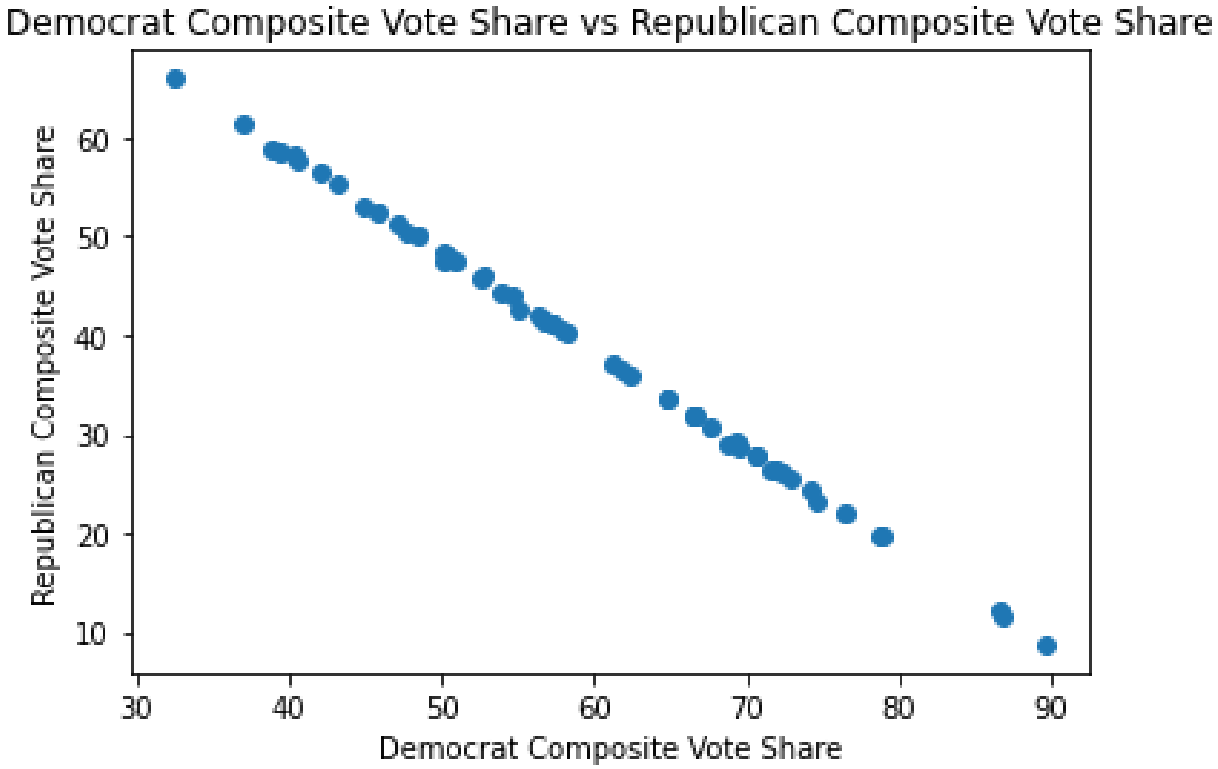

Fig 1: Democrat Composite Vote Share vs Republican Composite Vote Share

Fig 2 shows that the Democrats' composite vote share over 4 years on the x-axis and their final vote share in the 2018 House election on the y-axis. Each data point in the scatter plot represents a district. The Democrats' final vote share in the House election largely showed a relatively strong positive correlation with their composite vote share over 4 years. This meant that the Democrats' composite vote share in the previous elections, a metric for a district's intrinsic partisan lean, was a good predictor of their final vote share.

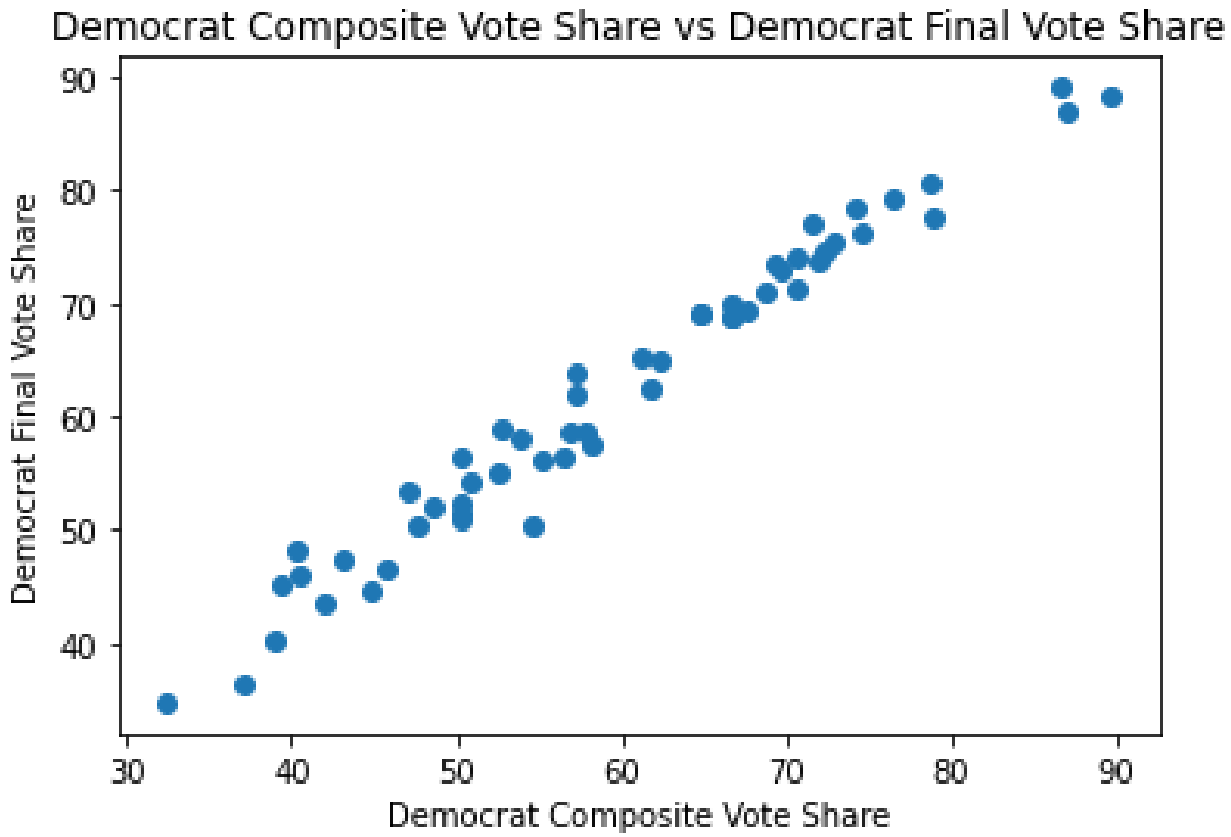

Fig 2: Democrat Composite Vote Share vs Democrat Final Vote Share in 2018 House Elections

Since not all of the variables are necessarily relevant for prediction or correlated with each other, we decided to create a correlation matrix to weed out the irrelevant ones. On the other hand, doing this with merely a correlation matrix could remove important variables that just happen to be related to others in a non-linear manner [31], meaning we had to exercise some caution and not remove every variable that was not correlated.

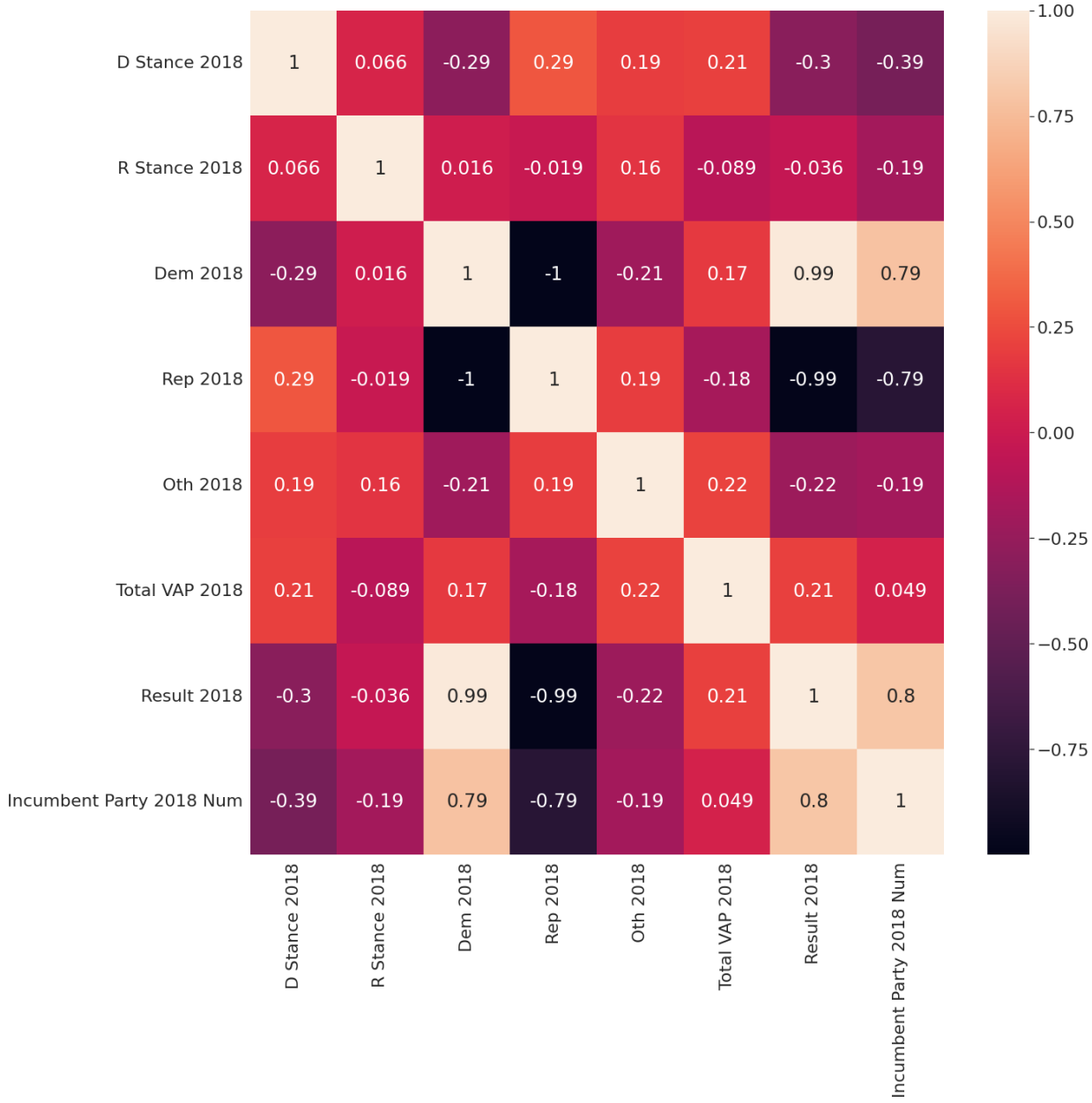

Fig 3: Correlation Matrix of Numerical Variables

Fig 3 is a correlation matrix heatmap of all our variables, where the numbers represent the correlation coefficient between two variables. Most of the variables are not significantly correlated as the absolute value of their correlation coefficient is below 0.5. However, some are directly correlated with each other, like the Democrats' final vote share (variable named "Result 2018") and the incumbent party (variable named "Incumbent Party 2018 Num"), with a correlation coefficient of 0.8, while others are inversely correlated with each other, like the Republicans' composite vote share from 2016 to 2020 (variable named "Rep 2018") and the incumbent party, which have a correlation coefficient of -0.79. As the Democrats' final vote share and the Democrats' composite vote share ("Dem 2018", the district's partisan lean) are extremely strongly correlated with a correlation coefficient of 0.99, we can tell that the third party composite vote share ("Oth 2018") is not very relevant, leading us to remove it from our final dataset. Another variable that does not seem to be particularly correlated with any other variable is the voting age population

of the district ("Total VAP 2018"), so we removed it from further analysis as well. While the candidate stance variables, "D Stance 2018" and "R Stance 2018" are not strongly correlated with anything else, candidate stances are generally perceived to be important, so we decided to keep them in our final dataset used for training our models.

The above preliminary analyses using scatterplots followed by further analysis using a correlation matrix for non-sentiment (numerical) variables showed that third-party candidates' vote share in preceding statewide races was irrelevant, and the district's incumbent party, its partisan lean, and its final vote share are all highly correlated.

Models Used

We want to use sentiment (text) data as well as non-sentiment numerical data to train our models as we consider both types of data relevant to predicting the election result. Text data and numerical data work better with different models. Since a single model can predict the outputs based on only one of the two types of data, we will need at least two models, one for text data and one for numerical data.

In our case, we fed the output of the model trained on text data to the input of the model trained on numerical data, giving us two "tiers" for a model system. The applicability, strengths, and weaknesses of our explored models are discussed next.

Overview of Models Explored

The type of model needed to work on sentiment (text) data is different from the type of model needed to work on numerical data. For starters, RNNs are expected to be good at sentiment data but not our numerical data [5]. Also, input dimensions for sentiment data are completely different from our other data, further implying that training on both sentiment and other data requires two different models.

Ridge Regression is often used for multiple regressions where the input variables are highly correlated, like in this scenario where many variables, such as the Democrats' composite vote share over 4 years and the district's incumbent party, are correlated. We chose Ridge Regression as the baseline because it is the most simplistic out of all the models used, but is still good for multivariate tabular data where some of the input variables might be correlated [26]. Its main disadvantage is that it takes into account all the variables, including the irrelevant ones [19].

k-Nearest Neighbors finds the k most similar elements in the training data and takes their average. It is often used for general machine-learning applications. We chose it because it seems to be decent at analyzing sentiment data and really good at analyzing non-sentiment data by virtue of it basing predictions off the most similar data points [20]. Its other strengths include that it is good for non-linear data, while its prime disadvantage is that it is relatively inefficient [12].

Regression Trees are decision trees with continuous outputs, and they work by dividing the feature space into smaller and smaller regions, with each region corresponding to a particular value of the output variable. As they use multiple decision points from the input variables, they would be good for complex, multivariate data like ours [7]. As such, an advantage of tree-based models (like Regression Trees) is that they are among the best approaches for tabular data like our numerical data [9].

Random Forests are ensemble models that use multiple Regression Trees and take the average out of all of them to get a better result than a single Regression Tree [17]. As they are based on regression trees, they have the same advantages and disadvantages, except that they are more accurate.

LSTMs and GRUs are types of Recurrent Neural Networks (RNNs). As opposed to feedforward neural networks, RNNs also have an internal state (memory) that allows them to have improved performance when working with sequences. This makes them especially useful for natural language processing [16]. However, as RNNs are not traditionally used for non-sequential numerical data like our non-sentiment data, we used the LSTMs and GRUs for only sentiment data [5]. As RNNs are types of neural networks, and as neural networks are not good for tabular data

(like our non-sentiment data), we did not use them for our non-sentiment data [3].

The Two-Tier Models Used

For numerical data, the models we used included Ridge Regression, k-Nearest Neighbors (kNN), Regression Trees, and Random Forest Regression, while for sentiment (text) data, we used Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU).

Fig 4 shows the architecture of our LSTM model, while Fig 5 shows the architecture of our GRU model, the models we explored for only text data.

Fig 4: Architecture of LSTM Regressor

Fig 5: Architecture of GRU Regress

To combine both the sentiment data and the numerical data, we needed to use a two-tier model where the output of the model trained on the sentiment data (the first tier) was added to the numerical data used to train the second tier, as shown in Fig 6.

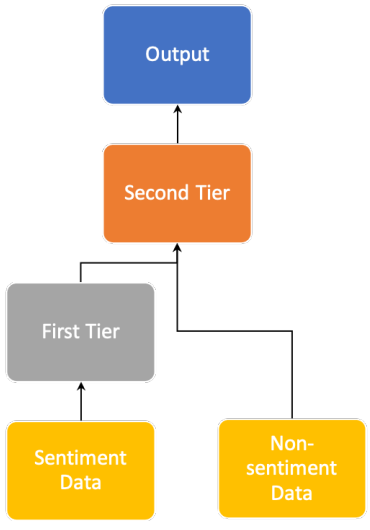

Fig 6: Architecture of Two-Tier Models

Table 1 shows the various combinations of first and second-tier models, where the first-tier models are used for sentiment data and the second-tier models are used for numerical data.

Table 1: Table of Combinations of First and Second Tier Models used

Model for Sentiment Data	Model for Tabular Data
Ridge Regressor	Ridge Regressor
kNN	kNN
Tree Regressor	Tree Regressor
Random Forest Regressor	Random Forest Regressor
LSTM	Ridge Regressor
LSTM	kNN
LSTM	Tree Regressor
LSTM	Random Forest Regressor
GRU	Ridge Regressor
GRU	kNN
GRU	Tree Regressor
GRU	Random Forest Regressor

Results and Discussion

Metrics

We used the Root Mean Squared Error (RMSE) of predicted vote share percentage, (as shown in Table 2), as the model performance metric. The lower the RMSE is, the better, as it is the error that needs to be minimized. If $\hat{y}_i$ is the predicted value of the output variable's ith sample, y_i is the actual value of the output variable's ith sample and n is the number of data points, RMSE is calculated as follows [28].

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)}{n}} \quad (1)$$

We also chose the Mean Squared Error (the square of the RMSE) as our loss function.

Model Evaluation

We did an 80-20 split of the data, where 80% of the data was used for training the models and 20% of the data was used for testing. This was done so we could detect overfitting if it occurred. Overfitting is the phenomenon where the model fits the training data so closely that it has the inability to generalize, leading it to yield inaccurate results on any new data [4]. Of the models we explored, due to its nature, Ridge Regression is less likely to overfit than the other models, while Tree Regression (and therefore its derivative Random Forest Regression) is more likely to overfit than the others [8].

As the models performed roughly the same on both the training and the test datasets, there was minimal overfitting or underfitting, if any. Initially, the features were manually put into a CSV file. Later on, we used the Pandas library to clean (remove outliers, NaN, and missing data) and prepare the data for training. We used the Scikit-learn library to split our data into train and test sets, as well as for feeding our data into the models. We got our neural network models from Keras, while we got our non-neural network models from Scikit-learn. After defining our models, we trained them and then tested them on our testing dataset, comparing the RMSEs between each model.

Analysis

Table 2: Model Performance Table

Architecture	RMSE (lower is better)
GRU Sentiment + Ridge Regressor Other	1.77
LSTM Sentiment + Ridge Regressor Other	2.04
Forest Regressor Single-Tier Other	2.26
LSTM Sentiment + Forest Regressor Other	2.33
GRU Sentiment + Forest Regressor Other	2.44
LSTM Sentiment + Tree Regressor Other	2.66
GRU Sentiment + kNN Other	2.77
LSTM Sentiment + kNN Other	2.77
GRU Sentiment + Tree Regressor Other	2.84
kNN Two-Tier	3.11
kNN Single-Tier Other	6.59

Forest Regressor Two-Tier	8.73
kNN Single-Tier Sentiment	13.56
GRU Single-Tier Sentiment	13.73
LSTM Single-Tier Sentiment	14.12
Tree Regressor Single-Tier Other	14.35
Forest Regressor Single-Tier Sentiment	14.38
Ridge Regressor Single-Tier Other	14.93
Ridge Regressor Two-Tier	14.93
Ridge Regressor Single-Tier Sentiment	15.00
Tree Regressor Two-Tier	15.03
Tree Regressor Single-Tier Sentiment	17.94

Table 2 summarizes the results from various combinations of models we used. The two-tier models usually outperformed (had lower RMSE) most of the single-tier models, indicating that both sentiment (text) data and numeric data (partisan lean, economic, political environment, etc.) are needed for accurate election vote share prediction.
There are some two-tier models which performed worse. For example, the two-tier Tree and Forest Regressors performed worse than their single-tier versions trained on non-sentiment data. This could be because the two-tier models involved two regressions (one on sentiment data and one on numerical data) and Decision Tree-based models (like Ridge and Forest Regressors) are especially bad at regression [8].

We also observe that the Ridge Regressor outperformed all other models for the numerical data in a two-tier setting with an RNN. One factor leading to this could be that parametric approaches like Linear Regression (and similar models like Ridge Regression) perform better than non-parametric approaches when there is a scarcity of data, like our non-sentiment data here [20]. Another factor could be that the Decision Tree-based models are worse at regression than kNN and Ridge Regressor models [8].

The two-tiered model with a GRU to analyze the sentiment and a Ridge Regressor to combine that with the analysis of all the other numerical variables perform the best with an RMSE of around 1.766 percentage points.

Summary and Future Work

In this paper we tried to predict US House election results using machine learning techniques. We used a range of input data including partisan lean of districts, economics, the national political environment, candidate political stances, and news headlines about each candidate in each race. To combine both text (sentiment data) and numerical data, we used two-tier models. In general, the two-tier models outperformed (had lower RMSE) than most of the single-tier models, indicating that both sentiment (text) data and numeric data (partisan lean, economic, political environment, etc.) are needed for accurate election vote share prediction.

Using our two-tier model, the GRU and Ridge Regressor combination obtained an RMSE of under 2 percentage points, meaning that we predicted the final vote share to within 2 percentage points.

Future Work

In our prediction we have excluded scenarios where either party's candidate runs unopposed or if both candidates are from the same party. It also assumes a two-way race, making it fail in any situation where independent/third-party candidates have an especially strong showing. Extending our work to consider such scenarios would be important to accurately predict the full range of US House of Representatives elections.

Another important manner in which our work can be extended to better predict most US House of Representatives elections would be using more data in the form of studying more districts and more election cycles.

References

[1] Wikimedia Foundation. (2023, May 19). *2018 United States House of Representatives elections in California*. Wikipedia. https://en.wikipedia.org/wiki/2018_United_States_House_of_Representatives_elections_in_California

[2] Wikimedia Foundation. (2023a, April 8). *2018 United States House of Representatives elections in Virginia*. Wikipedia. https://en.wikipedia.org/wiki/2018_United_States_House_of_Representatives_elections_in_Virginia

[3] Borisov, V., Leemann, T., Sessler, K., Haug, J., Pawelczyk, M., & Kasneci, G. (2022). Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 1–21. https://doi.org/10.1109/tnnls.2022.3229161

[4] Brownlee, J. (2019, August 12). *Overfitting and underfitting with machine learning algorithms*. MachineLearningMastery.com. https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/

[5] Brownlee, J. (2022, August 15). *When to use MLP, CNN, and RNN Neural Networks*. MachineLearningMastery.com. https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks/

[6] Bradlee, D., Rinn, D., Ramsay, A., & Crowley, T. (2021, November 9). *CA 2020 Congressional*. Dave's Redistricting. https://davesredistricting.org/maps#stats::d6ccadfa-243e-4ecc-9bb2-716b3f82afee

[7] Reader, T. C. (2021, October 4). *Decision tree regression*. The Click Reader. https://www.theclickreader.com/decision-tree-regression/

[8] Castillo, D. (2023, April 7). *Decision trees in machine learning explained*. Seldon. https://www.seldon.io/decision-trees-in-machine-learning

[9] Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022, July 18). *Why do tree-based models still outperform deep learning on tabular data?*. arXiv.org. https://doi.org/10.48550/arXiv.2207.08815

[10] St. Louis, F. R. B. of. (2023, April 27). *Gross domestic product*. FRED. https://fred.stlouisfed.org/series/GDP

[11] Isotalo, V., Saari, P., Paasivaara, M., Steineker, A., & Gloor, P. A. (2016). Predicting 2016 US presidential election polls with online and media variables. *Designing Networks for Innovation and Improvisation*, 45–53. https://doi.org/10.1007/978-3-319-42697-6_5

[12] Joby, A. (2021, July 19). *What is K-nearest neighbor? an ML algorithm to classify data*. Learn Hub. https://learn.g2.com/k-nearest-neighbor

[13] Jose, R., & Chooralil, V. S. (2016). Prediction of election result by enhanced sentiment analysis on Twitter data using classifier ensemble approach. *2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)*, 64–67. https://doi.org/10.1109/sapience.2016.7684133

[14] Joseph, F. J. (2019). Twitter based outcome predictions of 2019 Indian general elections using decision tree. *2019 4th International Conference on Information Technology (InCIT)*, 50–53. https://doi.org/10.1109/incit.2019.8911975

[15] Jones-Rooy, A., Mehta, D., Radcliffe, M., Rakich, N., Shan, D., & Wolfe, J. (2023, May 11). *Generic Ballot : 2018 polls*. FiveThirtyEight. https://projects.fivethirtyeight.com/polls/generic-ballot/2018/

[16] Pedamallu, H. (2020, November 30). *Rnn Vs Gru VS LSTM*. Medium. https://medium.com/analytics-vidhya/ rnn-vs-gru-vs-lstm-863b0b7b1573

[17] Raj, A. (2021, June 11). *A quick and dirty guide to random forest regression*. Medium. https://towardsdatascience.com/a-quick-and-dirty-guide-to-random-forest-regression-52ca0af157f8

[18] Ramsay, A. (2020, February 7). *Election Composites*. Medium. https://medium.com/dra-2020/election-composites-13d05ed07864

[19] Engati. (2023, May 18). *Ridge regression*. Engati. https://www.engati.com/glossary/ridge-regression

[20] Carnegie Mellon University Statistics & Data Science. (2021, July 8). *Supervised Learning*. Carnegie Mellon Sports Analytics. https://www.stat.cmu.edu/cmsac/sure/2021/materials/lectures/slides/18-KNN-kernel.html#1

[21] TensorFlow. (2023, March 23). *Tf.keras.preprocessing.text.tokenizer : tensorflow V2.12.0*. TensorFlow. https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer

[22] Tsai, M.-H., Wang, Y., Kwak, M., & Rigole, N. (2019). A machine learning based strategy for election result prediction. *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, 1408–1410. https://doi.org/10.1109/csci49370.2019.00263

[23] *US PCE inflation rate (I:USPCEQIR)*. YCharts. (2023, April 27). https://ycharts.com/indicators/us_pce_quarterly_inflation_rate

[24] *US unemployment rate (I:USURSQ)*. YCharts. (2023b, April 7). https://ycharts.com/indicators/us_unemployment_rate_quarterly

[25] Bradlee, D., Rinn, D., Ramsay, A., & Crowley, T. (2022, June 12). *VA 2020 Congressional*. Dave's Redistricting. https://davesredistricting.org/maps#stats::28033d62-2027-4661-95d0-557621f823e9

[26] Zach. (2021, August 26). *When to use Ridge & Lasso regression*. Statology. https://www.statology.org/when-to-use-ridge-lasso-regression/

[27] Zolghadr, M., Niaki, S. A., & Niaki, S. T. (2017). Modeling and forecasting US presidential election using learning algorithms. *Journal of Industrial Engineering International*, *14*(3), 491–500. https://doi.org/10.1007/s40092-017-0238-2

[28] Gunjal, S. (2021). *What is root mean square error (RMSE): Data Science and Machine Learning*. Kaggle. https://www.kaggle.com/general/215997

[29] pandas. (2023, April 24). *Pandas.dataframe*. pandas 2.0.1 documentation. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html

[30] *scikit-learn*. scikit-learn: machine learning in Python — scikit-learn 1.2.2 documentation. (2023, March 9). https://scikit-learn.org/stable/

[31] Aggarwal, R., & Ranganathan, P. (2016). Common pitfalls in statistical analysis: The use of correlation techniques. *Perspectives in clinical research*, *7*(4), 187–190. https://doi.org/10.4103/2229-3485.192046