Predicting The 2020 USA Presidential Election Using Supervised ML Models

Yuuki Inada, Nelson Niu, Jerry Tann

Abstract

The 2020 Presidential election was fought between the incumbent president Donald Trump and Joe Biden. Regarding the polarized political climate in the US, both candidates had clear supporter states but there were also tossup states where the results can end up on either side. This paper aims to predict the results of the presidential election for the tossup-states using classical machine learning (ML) models and advanced ML models and compare the accuracy. The results show that using the Gradient Boosting Decision Tree (GBDT), the model is able to predict the election results of 15 states out of 16 total tossup-states and had a clear advantage in its performance compared to traditional ML models such as Logistic Regression and Naive Bayes.

Keywords: Supervised Machine Learning, Classification, 2020 US Presidential Election

1. Introduction

Predicting the US election result has long been a topic of interest for many researchers, this is because the result of US election can have a major impact not only on the local market but also on the global economy (Zolghadr et al., 2018). Unlike elections in many other countries, presidential elections in the US are conducted differently: winning the popular vote in these elections does not necessarily mean winning the election as a whole. Instead, 51 statewide elections are held for each state and the District of Columbia (DC)[1]. Each state is allocated a varying number of electoral votes, depending on the population of each state. A candidate will win all[2] the electoral votes of a state if said candidate wins a plurality in that state. Since 1964, there are 538 electoral votes. The candidate that wins a simple majority of the electoral votes will win the election as a whole and becomes president-elect. As such, the outcome of presidential election depends on the outcome of each state.

Studies to predict US election results started as early as the 19th century, using explanatory variables such as opinion polls, historical data, and demographic information to make predictions about how people will vote. With the advancement of computation power and statistical models, ML has become increasingly popular in recent years to predict the outcome of US elections given its efficiency and accuracy in analysing big data. For example, one study (Isotalo et al., 2016) used ML models with linear regression which produced a reasonably accurate result. They also found significant correlations between polls and betting odds and polls and Facebook page like. In another study (Sinha et al., 2020), the researchers used lasso regression with various economic factors to forecast the vote share

[1] In this paper from this point forward, 'State' refers to all 50 US states and DC.

[2] In this paper, we assume so. In reality, Maine and Nebraska split their votes according to the election of each of their congressional districts. In addition, we also assume that there are no faithless electors.

for the Republican and Democratic parties in 2020. The factors included inflation, unemployment rate, economic growth, gold prices, oil prices, and exchange rate. In Holbrook and DeSart's study (1999), they used state polls to forecast the presidential outcomes in each state in 2004 based on the Monte Carlo simulation method.

Based on described existing studies, we decided to implement state polling data as our primary predictor to forecast the 2020 US presidential election outcome in American states. Our analysis will be based on using explanatory variables for each state and classifying each state. Furthermore, we propose to use GBDT as one of our ML classifiers. GBDT was initially developed to improve decision tree performance and has become increasingly popular in modern ML tasks. However, the application of the GBDT in forecasting election result context remains rare.

In summary, our contributions can be described as follows: 1). We propose to use state-level polls to predict voting outcomes in the states as the outcome of presidential election depends on the outcome of each state. We will use the states that are considered "safe" with clear party support direct as training data, to predict states that are considered "tossup" where the supported party can be on either side. 2). With polling data as our primary explanatory variable to predict the election outcome, we propose to adjust the recent polling data with mean errors of past presidential election results which is be detailed in Section 3. 3). In addition to using basic ML models such as Logistic Regression and Naive Bayes model to predict the election result, we propose to incorporate advanced ML model GBDT.

2. Problem Formulation

2.1 Logistic Regression Model

$$p(x) = \frac{1}{1+e^{-\frac{(x-\mu)}{s}}} \qquad (1) \qquad\qquad p(x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}} \qquad (2)$$

The Logistic function is in the form of Equation (1) where μ is a location parameter where $p(\mu) = 1/2$ and s is a scale parameter. It can also be expressed in the form of Equation (2) where $\beta_1 = \frac{1}{s}$ and $\beta_0 = -\frac{\mu}{s}$ is the y-intercept of the line $y = \beta_0 + \beta_1 x$.

2.2 Naive Bayes Model

Let $\mathbf{x}$ be a vector of n predictors $(x_1, ..., x_n)$ and C_k be the k-th class where there are K classes.

$$p(C_k|\mathbf{x}) = \frac{p(C_k)p(\mathbf{x}|C_k)}{p(\mathbf{x})} \qquad (3) \qquad\qquad p(C_k|\mathbf{x}) \propto p(C_k)\prod_{i=1}^{n} p(x_i|C_k) \qquad (4)$$

Using Bayes' theorem, the conditional probability can be decomposed as Equation (3).

Suppose that all predictors in vector $\mathbf{x}$ are mutually independent conditional on the k-th class C_k. Then, the conditional probability Equation (3) can be expressed as Equation (4).

Therefore, the Bayes classifier assigns a class label $\hat{y}$ such that

$$\hat{y} = \underset{k\in\{1,...,K\}}{\operatorname{argmax}}\ p(C_k)\prod_{i=1}^{n} p(x_i|C_k) \qquad (5)$$

2.3 Gradient Boosting Decision Tree (GBDT)

The basic idea behind GBDT is to sequentially add decision trees to the model, where each tree is trained to correct the mistakes made by the previous tree. Let us consider a simple decision tree model, which is represented as:

$$f(x) = \sum w_i * I(x \in R_i). \tag{6}$$

To train a decision tree, we need to learn the weights w_i for the ith leaf node. This is done by minimizing the loss function $L(y, h(x))$ between the true label y and the predicted label $h(x)$ for a given training example (x, y).

Now, let us consider the case of a GBDT model with K decision trees. The prediction made by the GBDT model can be represented by the following equation:

$$F_k(x) = F_{k-1}(x) + \eta \cdot f_k(x) \tag{7}$$

where $F_k(x)$ is the prediction of the GBDT model for a given input x, η, $(0 \leq \eta \leq 1)$ is the learning rate for generalizing the model, and $f_k(x)$ is the prediction made by the kth decision tree in the model.

To train a GBDT model, we first initialize the model with a single decision tree, which makes predictions based on some initial set of weights w_i. Then, we sequentially add $K - 1$ more decision trees to the model, where each tree is trained to correct the mistakes made by the previous tree. Let us define the residual error r_i for the ith training example as:

$$r_i = y_i - F_k(x_i). \tag{8}$$

To train the kth decision tree in the GBDT model, we can minimize the loss function:

$$L\left(y, F_k(x)\right) = \sum \left(r_i - f_{k^{\cdot}}(x_i)\right)^2. \tag{9}$$

This loss function measures the error between the true label y_i and the current prediction $F_k(x_i)$ for each training example, and the kth decision tree is trained to correct this error by minimizing the residual error r_i.

3. Proposed Solution

3.1 Classifying Safe and Tossup States and Choosing Training Data

We obtain the statewide polling data for the 2020 Presidential Election. The polling data is then filtered only to consider the Democratic nominee and the Republican nominee (i.e. Joe Biden and Donald Trump), including the polling data before they were nominated. Polls that include other candidates are also considered, but only the data between Joe Biden and Donald Trump are used. For each state, if the Democratic nominee wins in all the polls, that state is classified as 'D'; if the Republican nominee wins in all the polls, that state is classified as 'R'; and, if both candidates win in different polls, or there exists a tie, that state is classified as 'tossup'.

States that are classified as 'D' or 'R' in this part are considered 'safe' states, and will be classified as such in our final predictions. These states will also be used as training data in our prediction models.

Under these classification rules, using the 2020 polling data, 35 states are classified as either 'D' or 'R', while the remaining 16 are tossup states. For our following predicting models, these 35 non-tossup states will be used as the training set, while the remaining 16 tossup states will be used as the testing set.

Compared with the actual 2020 Presidential Election results, the 35 states classified 'D' or 'R' are indeed the correct classification, and in addition, they are safe states that resulted in a margin of victory larger than 10%. On the other hand, 14 out of 16 of the tossup states had a margin of victory of less than 10%. Therefore, our classification rules are highly accurate in determining which states are safe for 2020.

3.2 Explanatory Variables

3.2.1 Filtering and Calculating Lead Margins for the 2020 Polling Data

As mentioned in the Introduction, the national popular vote does not determine the winner of the election. Therefore, in this study, we instead consider the statewide polling data. Using the 2020 polling data obtained in Section 3.1, the data is filtered only to consider the Democratic nominee and the Republican nominee, including the polling data before they were nominated. Polls that include other candidates are also considered, but only the data between the Democratic and Republican nominees are used.

We also filter the polling data to only consider polling data from a year before the election to the nearest month, i.e. from November 2019 to November 2020. Then, for each state, we take the difference between the percentage of Republican votes and Democratic votes to obtain the lead margin, where a negative value indicates a Democratic lead, 0 indicates a tie, and a positive value indicates a Republican lead. We then take the unweighted mean of the lead margins for all polls of each state. The unweighted means of the lead margin for the 2020 polls are shown in Table 1 under the column "2020 Lead".

3.2.2 Calculating and Adjusting for Polling Error

We believe that there exist systematic polling errors varying between each state. It can be seen in 2016 when the Republican nominee Donald Trump significantly overperformed in the election compared to the polls in Republican and tossup states. From this, we hypothesise that since 2016, polls conducted in Republican and tossup states tend to lean more Democratic compared to the election results. This may be due to Democratic voters being more likely to participate in polls compared to Republican voters. Therefore, we will adjust the polling data based on historic errors for each state.

Similarly, in Sections 3.1 and 3.2.1, we obtain and calculate the polling lead margins from 2004 to 2016 are calculated using the same method. The true election lead margins from 2004 to 2016 are also obtained and calculated similarly without taking the mean. Then, the polling error of each year for each state will be the difference between the true election results and the unweighted mean of the lead margins, where a negative value indicates that the poll leans more Democratic than the true result, and a positive value indicates that the poll leans more Republican than the true result.

We repeat[3] this for the elections of 2004 to 2016. Then, we take the mean[4] of the errors across all years for each state as shown in Table 1 under the 'Error' columns. We define this value as the mean error. Then, we adjust for polling error by subtracting the mean error from the 2020 unadjusted lead polling data as shown in Table 1 under the '2020' columns.

We will be using the 2020 adjusted lean margins as our main explanatory variable.

Table 1: The errors from 2004 to 2016, the mean error across 2004 to 2016, the 2020 lead margins, and the 2020 adjusted lead margins for the first three states

State	Error (%)					2020 (%)	
	2004	2008	2012	2016	Mean	Lead	**Adjusted**
AL	-6.000	-0.376	-5.700	-7.730	-4.951	20.277	**25.229**
AK	-0.550	-9.690	NaN	-10.596	-6.945	6.533	**13.478**
AZ	-2.764	1.480	-6.257	-1.573	-2.278	-2.808	**-0.529**
...	...	...	...	...	...	...	**...**

3.2.3 Demographic Explanatory Variables

Aside from the main explanatory variable - polling data, we have chosen 5 additional independent variables to predict the US presidential election as shown in Table 2. These variables have previously been shown to be associated with US election outcomes in several other similar studies (Zolghadr et al., 2018; De Neve, 2014; Lewis-Beck and Tien, 2014).

Table 2: Description of Explanatory Variables

Index	Variable Name	Variable Description
1	Polling data	2020 adjusted lead margins (See Table 1)
2	Income	Personal income, RPPs, income per RPPs in each state
3	Education level	Ratio of people with high school qualification or bachelors
4	Population	Population density in each state
5	Labour	Unemployment rates in each state
6	Race	Percentage of different race in each state

3.3 Statistical Assumptions

3.3.1 Multicollinearity

Multicollinearity refers to the linear relationship between data between explanatory variables and presents detrimental issues such that the partial regression coefficient becomes highly volatile between different samples. (Allen, 1997). To detect multicollinearity, we use the Variance Inflation Factor (VIF) which is defined as $VIF_i = \frac{1}{1-R_i^2}$ where R_i is the

[3] There exist NaN error values, where no polling took place within a year in that state. It is assumed that these states are safe states due to the one-sidedness of recent historical results and the political landscape of said states.

[4] The mean error ignores NaN values such that only the years without NaN values are considered.

Coefficient of determination for the ith individual variable. A large value in VIF indicates multicollinearity in the given set of data. In this paper, the threshold of indicating multicollinearity within the explanatory variable will be set to 10 (Alin, 2010).

3.3.2 Data Pre-processing Benchmarks

With the aforementioned statistical assumptions cleared we are able to fit the data into our model. Figure 1 is the flowchart of the benchmarks for our data-pre-processing procedure.

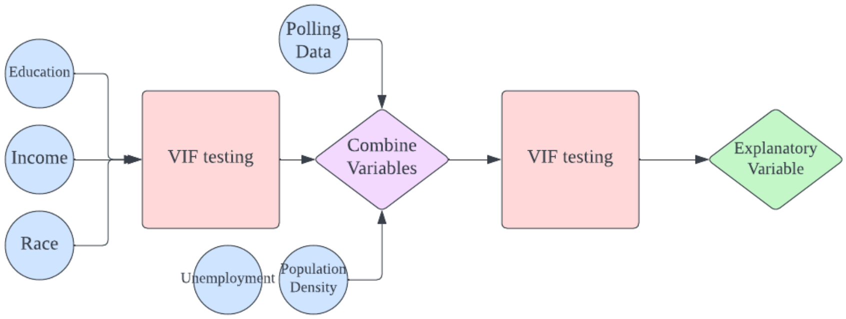

Figure 1: The explanatory variables having multiple indices will be filtered in the VIF test to ensure heteroskedasticity within their attributes. Then we combine all the data and check for multicollinearity. The set of variables clearing the second stage of VIF testing will be used as the explanatory variable in our model.

4. Numerical Experiments

4.1 Multicollinearity within the variable attributes

We first check for multicollinearity within our explanatory variables to ensure heteroskedasticity. We use 6 types of variables described in Table 2. Within these variables, 4 variables; Polling data, Income data, Education level data, and Race data have multiple data columns within their attributes. However, we split the model based on the adjusted/un-adjusted polling data illustrated in Section 3.3.2 so we would need to check for multicollinearity within the variables illustrated above except the polling data. This paper omitted several features within Income, Education, and Race attribute to ensure heteroskedasticity within our attributes.

4.1.1 Multicollinearity within our Mixed Explanatory Variables

With the variables that we ensured we have heteroskedasticity within the attributes, we now want to concatenate our variables to use as the explanatory variables for our models. However, we have to ensure once again heteroskedasticity within the mixed variables as explained in Figure 1. From this, we find that the education attribute, personal income attribute, and labour attribute have a high correlation with each other and cannot coexist in our explanatory variables when modelling. Considering the magnitude of importance of these three variables, we decided to make split models, the first model having an income variable, the second model having an education variable, and the third model having the labour variable. In addition, the **white_percentage** variable had a high correlation between all the three variables explained above so we decided to omit it altogether. The devised model specification and its VIF are shown in Table 3.

Table 3: Model Specification and its VIF values

Model	Data Feature	Attribute	VIF	Result
Model With Income Variable	income_per_RPPs	Income	6.42	Pass
	B01001_calc_PopDensity	Population	2.21	Pass
	hispanic_percentage	Race	2.79	Pass
	black_percentage	Race	3.11	Pass
	american_indian_percentage	Race	1.61	Pass
	asian_percentage	Race	2.15	Pass
	other_race_percentage	Race	4.87	Pass
	2020_polls_adjusted	Polls	2.64	Pass
Model With Education Variable	PercentageBachelorsOrHigher	Education	6.74	Pass
	B01001_calc_PopDensity	Population	2.22	Pass
	(same as above)	Race	...	Pass
	2020_polls_adjusted	Polls	2.33	Pass
Model With Labour Variable	unemployment_rate	Labor	2.19	Pass
	B01001_calc_PopDensity	Population	7.73	Pass
	(same as above)	Race	...	Pass
	2020_polls_adjusted	Polls	2.40	Pass

4.2 Summary of Classification Results

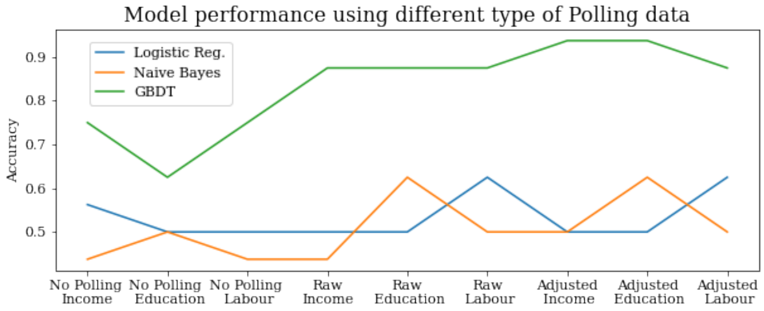

Figure 2: Linegraph showing the Accuracy scores across different models and Explanatory variables. GBDT performance is stellar for all patterns.

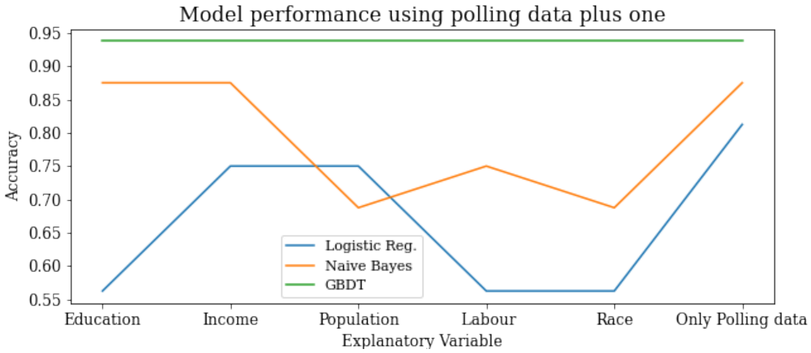

Figure 3: Linegraph showing the Accuracy scores across different models using Adjusted Polling data and one more attribute of Explanatory Variable.

4.2.1 Models using Mixed Variable Attributes

In accordance with Table 3, we make 3 models containing different sets of explanatory variables. Figure 2 shows the classification results and their accuracy for all the models using different attributes for explanatory variables. All models took under a second to train and did not see any significant difference in time complexity.

As shown Figure 2, GBDT was able to predict the results with much higher accuracy in comparison to Logistic Regression and Naive Bayes models. The accuracy for GBDT model is 93.75%, missing only Florida. Therefore, with the adjusted polling data and other demographic data produced before the election, we were able to predict most of the states.

Figure 2 shows the accuracy scores for all the devised model using different types of polling data. We could see that the polling data seems to be a significant factor when it comes to predicting the election results, increasing the accuracy tremendously compared to the model without any polling data. In addition, Education data seems to be a very good predictor since for all the polling data patterns, the model using education attributes performs the best. On the contrary, labour data seems to lack predicting power as they tend to perform worse. Lastly, we could interpret the significance of adjusting the polling data since just by using the raw polling data, the prediction accuracy does not reach the level of 93.75% which is realized using adjusted polling data. This signifies the flaw in the polling data whereas the raw polling data does not accurately reflect the whole population by a considerable margin.

4.2.2 Models using Adjusted Polling data plus one more attribute

In Section 4.2.1, we are able to make very accurate predictions using the GBDT algorithm with adjusted polling data. Now we investigate which data attributes hold the most predicting power together with the adjusted polling data. To check this, we use the adjusted polling data together with one more data attribute and check the accuracy of its prediction. Figure 3 shows the line plot showing the accuracy for all the predictions using adjusted polling data and one more explanatory variable attribute. We could see that GBDT can consistently produce high accuracy predictions with 93.75% across all attribute combinations. On the other hand, the Logistic Regression model was not able to produce a very accurate classification not having much improvement across all combinations. Also, Naive Bayes was able to increase their accuracy compared to a mixed approach having 87.5% for Education and Income attributes for Naive Bayes, misclassifying two states.

When comparing each of the attributes, the Income variable tends to perform well across all ML models. By comparing the results illustrated in Section 4.2.1, the Income variable has strong prediction power in itself but does not reinforce the accuracy when it comes to a mixed variable model. In addition, we can indicate that the Race attribute alone does not yield accurate predictions, which can be found from the dip in the line graph. However, all of these findings can be seen as irrelevant if we see the accuracy result of "Only Polling Data" since all the ML models are able to predict the results well by just the adjusted polling results, meaning the main predicting factor used in all of the models is the polling data, which again reinforces the significance of the adjusted polling data.

5. Conclusions, Limitations and Future Research

As shown in Section 4.2, our devised model was able to yield very accurate results using solely the data which was present before the actual presidential election. Especially the GBDT model produced extremely accurate predictions across every combination of data and was the best among other ML algorithms that this paper investigated. Combining this with our initial classification of safe states, we managed to accurately predict 50/51 states.

However, there are points where the results defied the initial speculation. Firstly, we learned that all the devised models used adjusted polling data for their main predictor and other variables were insignificant and undermined the quality of predictions. With the characteristics of ML, an assumption was made, that the more variables we use to generate the prediction the more robust and accurate the predictions will become. However, looking at Figure 3, this assumption was contradicted whereas all the ML models produced high accuracy when using solely the adjusted polling data. From this, we can say that when some variable explains a significant portion of the result, other variables cannot reinforce the quality of the result but rather make the model less accurate. However, this paper acknowledges that this cannot be generalized since this prediction was trained from a small sample size with only 35 non-tossup states and may have gotten away with a highly overfitted model. Further research can be done to investigate more models that amend these small sample issues when combined with ML.

Secondly, the static nature of our model may diminish the robustness since the main variating variable for our input is solely the polling data. For our explanatory variables, we mainly used demographic data that was updated inconsistently. Therefore, the model is rather static and may become obsolete as it does not incorporate the most recent data. Further investigations into what kind of data is used for the explanatory variable, which is frequently updated and released, are needed.

Lastly, the output for our predictive model is merely a categorization value (D or R) and does not provide any numerical election results for actual votes. Therefore, our output does not provide any insights into the margin of the results which can be an important certainty factor for our prediction value. For future research, we could incorporate a regression model and output a confidence interval for our result for a more informative prediction.

References

Aylin Alin. Multicollinearity. *WIREs Computational Statistics*, 2(3):370–374, 2010. doi: https://doi.org/10.1002/wics.84. URL `https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.84`.

Michael Patrick Allen. *The problem of multicollinearity.* Springer US, 1997. ISBN 978-0-585-25657-3.

Jan-Emmanuel De Neve. Ideological change and the economics of voting behavior in the us, 1920–2008. *Electoral Studies*, 34:27–38, 2014.

Veikko Isotalo, Petteri Saari, Maria Paasivaara, Anton Steineker, and Peter A Gloor. Predicting 2016 us presidential election polls with online and media variables. In *Designing Networks for Innovation and Improvisation*, pages 45–53. Springer, 2016.

Michael S Lewis-Beck and Charles Tien. Congressional election forecasting: structure-x models for 2014. *PS: Political Science & Politics*, 47(4):782–785, 2014.

Pankaj Sinha, Aniket Verma, Purav Shah, Jahnavi Singh, and Utkarsh Panwar. Prediction for the 2020 united states presidential election using machine learning algorithm: Lasso regression. 2020.

Mohammad Zolghadr, Seyed Armin Akhavan Niaki, and STA Niaki. Modeling and forecasting us presidential election using learning algorithms. *Journal of Industrial Engineering International*, 14(3):491–500, 2018.