Kaggle Report - Classification

Statistics 101C - Introduction to Statistical Models and Data Mining

Professor: Miles Chen

Team: Daniel Do, Khang Thai, Andy Ho, Jessie Yu, Christopher Le

August 3, 2024

Introduction:

In this project, classification analysis was performed based on datasets that contained information about 2020 Presidential Election winners from the specific demographics, by training the model through the train and test datasets. The primary objective of this project was to predict the response variable "winner", given the explanatory variables from the provided dataset.

From the initial inspection of the datasets, the explanatory variables of median age and Cuban population were narrowed, which could have contributed to a strong association of the response variable "winner". These variables could be based on the specific culture and societal values that these specific demographic groupings have been exposed to. Throughout this report, models were developed to have a more refined prediction of the presidential election winners based on the demographic identification variables.

Exploratory Data Analysis

Figure 1 uses boxplots to show the total number of votes cast for each county winner. The figure shows that Biden tended to accrue a higher number of votes for his county victories compared to Trump; his maximum and upper quartile votes surpassed Trump's voting numbers.

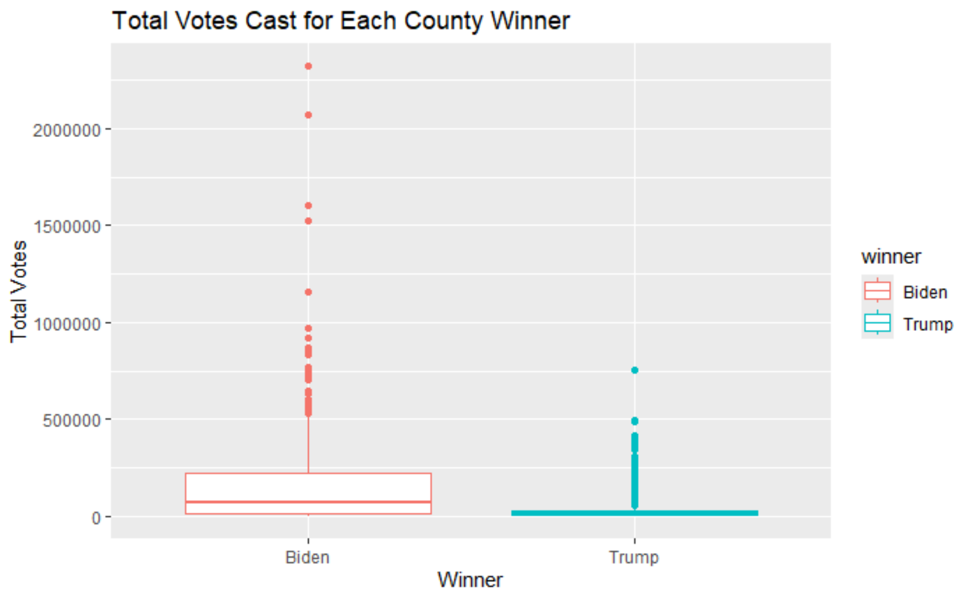


Figure 1: Total Counts Cast for Each County Winner

Figure 2 depicts a histogram of the total number of votes across the 3,111 counties in the data set. Looking at the figure, the histogram seems to be heavily right-skewed. This means that many counties have a small number of total votes while few have a higher total count. Because of the skew, we can apply logarithmic transformations to account for this during the pre-processing portion of the analysis.

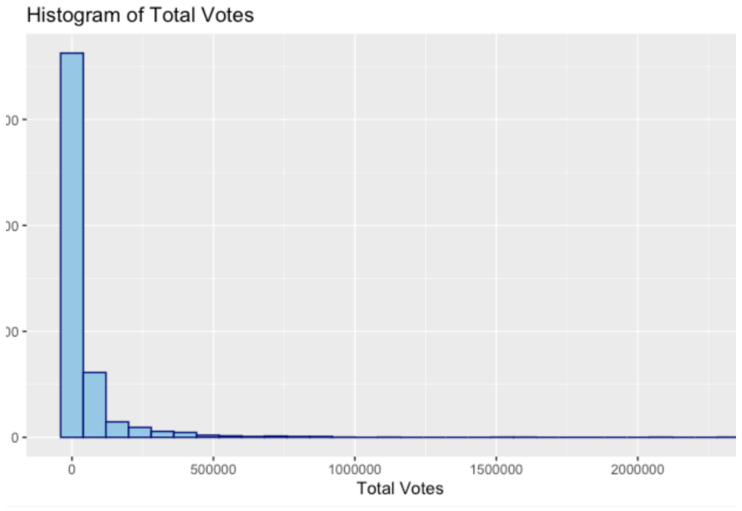


Figure 2: Histogram of Total Count of Votes

In Figure 3, a correlation map plot is depicted to visualize the correlation coefficient values between all of the variables. These values represent the relationship among all the variables in the data set. Some variables have extremely high correlations with one another, as seen in the plot. Therefore, issues involving multicollinearity could arise, which can be dealt with by removing these highly correlated variables.

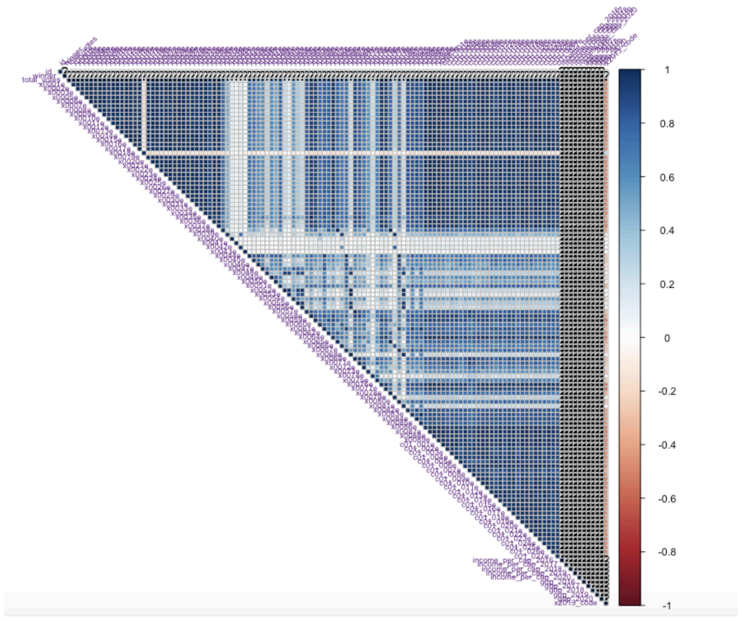


Figure 3: Correlation Map

In Figure 4, a box plot of the median age of the voters based on the explanatory variable of winner was developed. Voters who preferred Biden had a much younger median age compared to the median age of voters who preferred Trump. This could be an influential factor that determines the presidential winners.

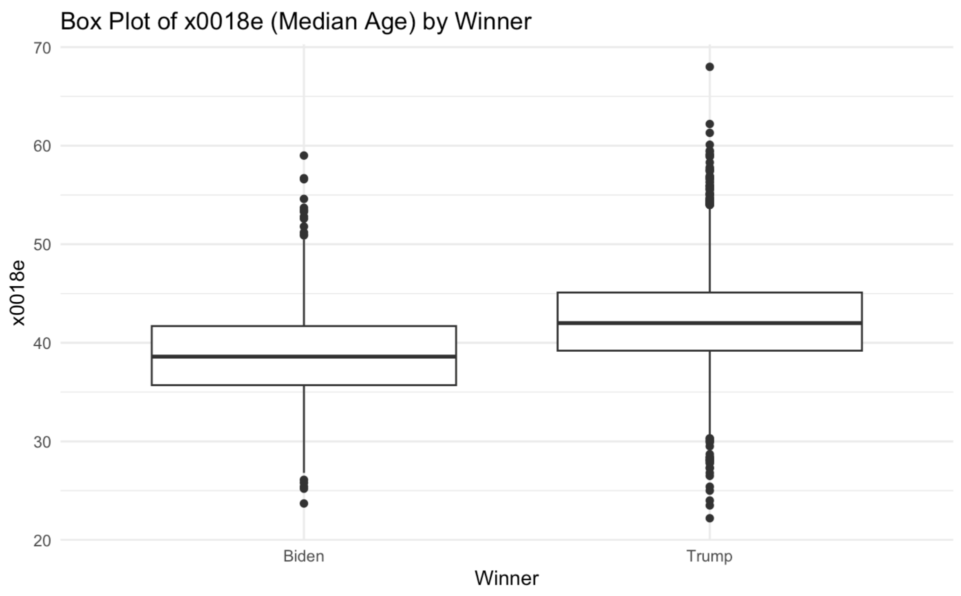


Figure 4: Box Plot of Median Age by Winner

In Figure 5, a bar plot was developed to show the Cuban population voters with a relationship to the categorical explanatory variable winner. The bar plot implies that Cuban voters are more likely to vote for Biden than Trump. Therefore, the cultural background that Cuban-Americans come from could lead to political views that align more to Biden's than that of Trump's.

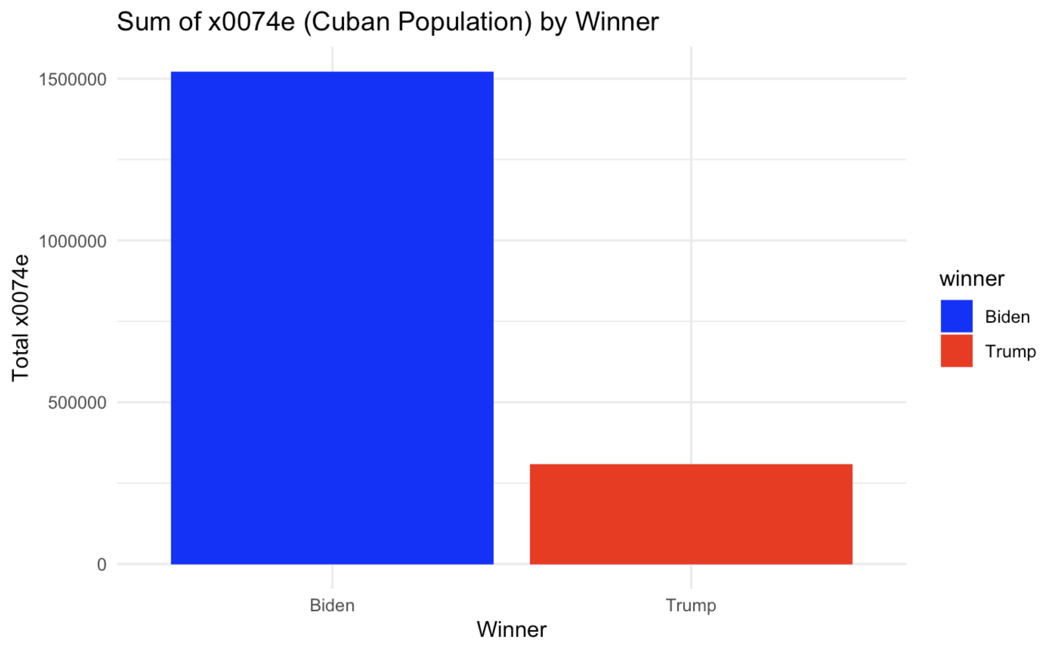


Figure 5: Bar Plot of Cuban Population by Winner

Figure 6 shows the bar plot of the urban/rural code based on the presidential election winners. To give context, the x2013_code represents the urban/rural code: 1 is most urban (least rural) while 6 is most rural (least urban). This bar plot demonstrates that those who live in more urbanized areas are more likely to vote for Biden, whereas those who live in more rural areas are more likely to vote for Trump.

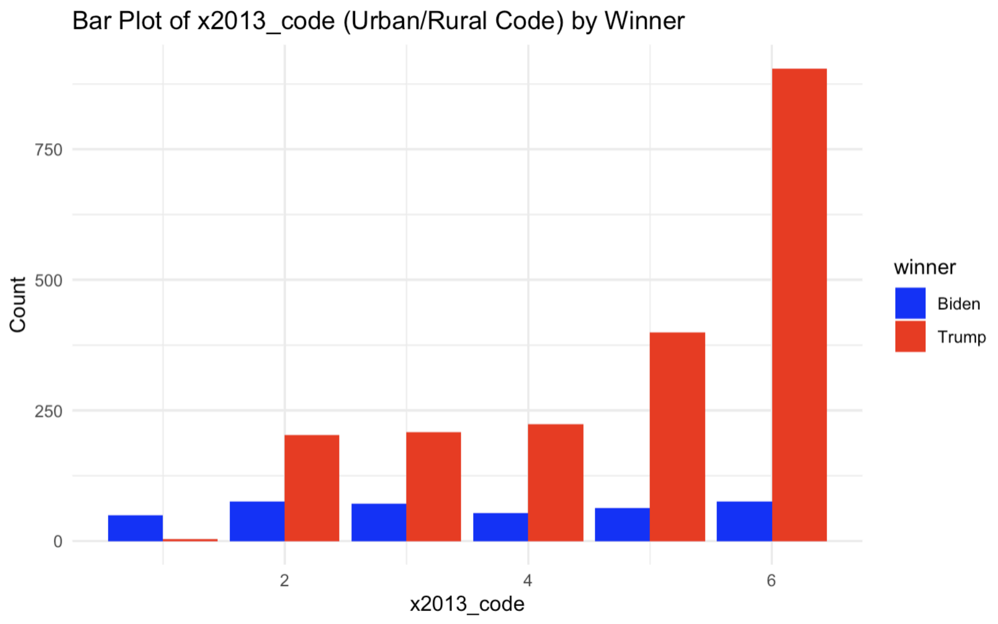


Figure 6: Bar Plot of Urban/Rural Code by Winner

Figure 7 displays the bar plot of the Asian population by the presidential election winners. From the data visualization, it can be seen that Asians are more likely to vote for Biden compared to Trump. This could be significant to Presidential campaigns, in which Biden (Democratic Party) could aim more commercial ads to Asian Americans by the more aligned views compared to Trump (Republican Party).

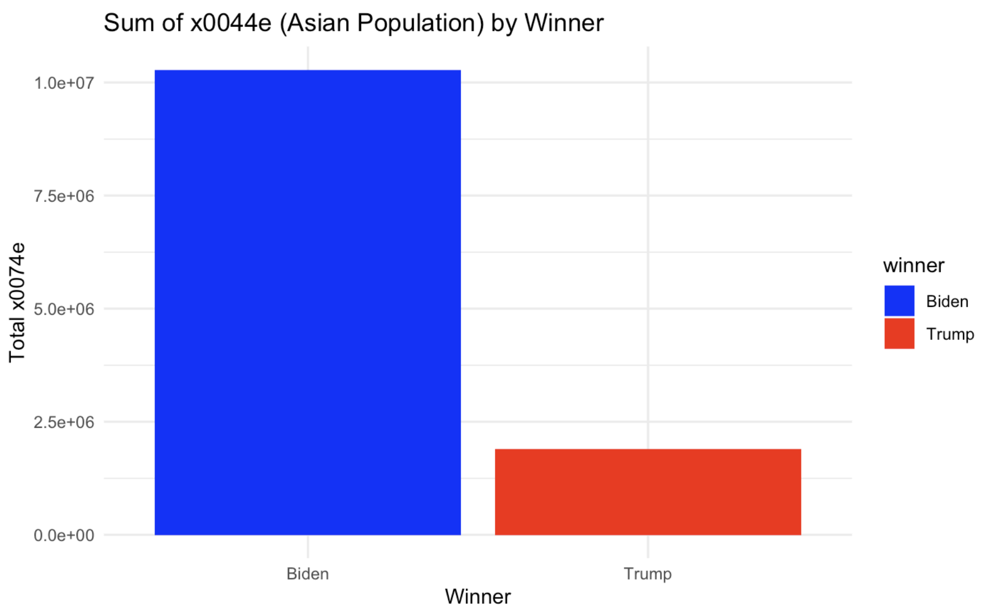


Figure 7: Bar Plot of Asian Population by Winner

Figure 8 exhibits the age population of 65 years or older with relationship to the response variable winner. The bar plot implies that the explanatory variable of age being 65 years or older may not have a significant indication to the response variable, as the counts between that of Biden and Trump are relatively similar. Thus, the age of 65 years or older does not have a high correlation with the response variable winner.

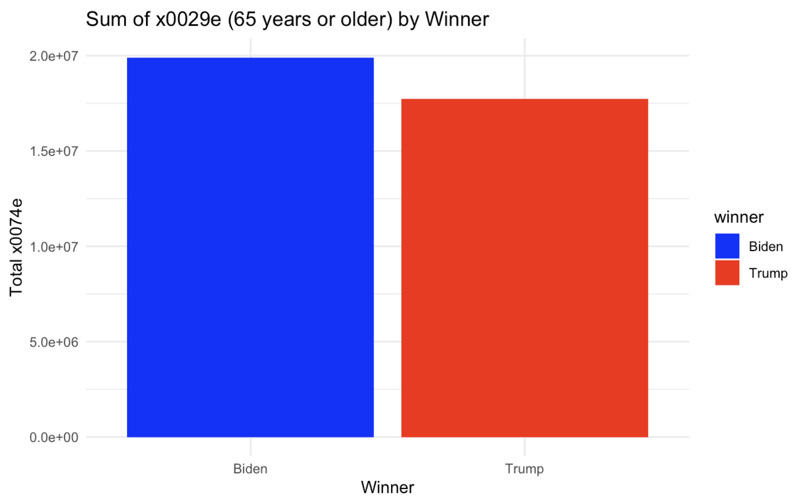


Figure 8: Bar Plot of Population 65 years or older by Winner

Figure 9 displays the number of Bachelor’s Degree holders or higher by winner for those that are at least 25. Since the age of matriculation for most bachelor’s degrees falls on average at a minimum of 22 years old, it’s apt to say that this information displays the number of people with a bachelor’s or advanced degree by each winner. As a result, we can see that there does seem to be correlation with the number of bachelor’s or advanced degrees and the response variable “winner”, shown by how the count for Biden is much higher.

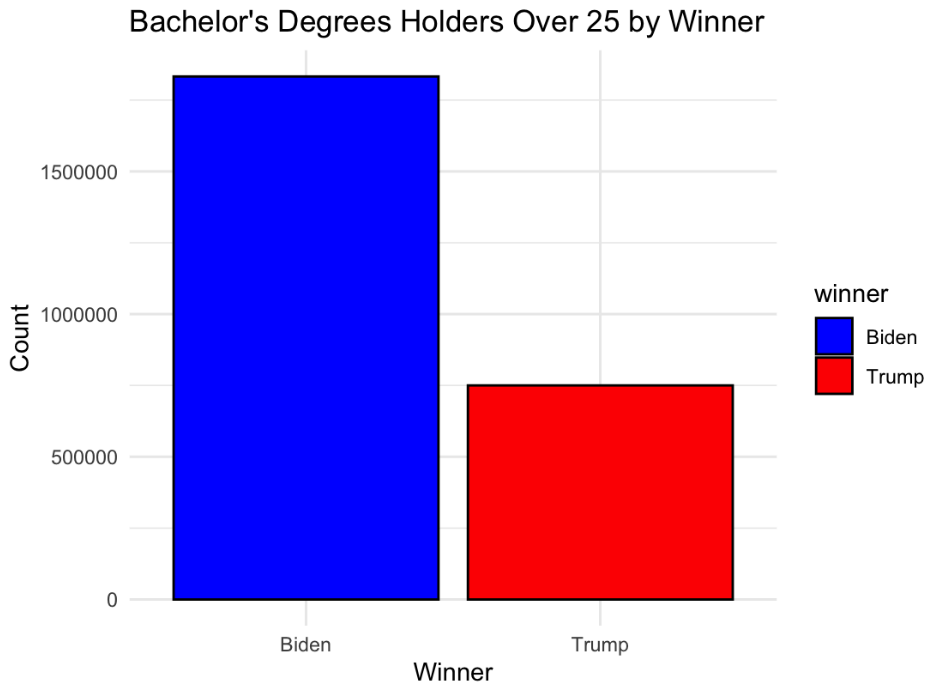


Figure 9: Bar Plot of Population 25 years or older with a Bachelor’s or Higher by Winner

Pre Processing

From the various models that were built by the team members, pre-processing was implemented to improve the validity of the predictions produced.

For example, to handle categorical data, step_dummy was applied to convert all categorical predictors into dummy variables. This enabled the model to effectively process and analyze categorical data by breaking it down into binary format. This improves the model by providing a clearer representation of qualitative variables.

In addition, normalization of numeric predictions was performed using step_normalize. This step scales numeric data to have a mean of zero and standard deviation of one. Normalization helps significantly enhance the model’s performance by preventing predictors with larger scales in the model.

To address multicollinearity, step_corr was utilized to remove numeric predictions that have a high correlation greater than 0.8 with other numeric predictors. Eliminating these highly correlated variables helps protect the model from overfitting and multicollinearity, which can skew the model's predictions and make it less generalizable to new data.

Candidate Models

Model 1:
A decision tree model was used with the "rpart" engine and classification mode in order to attempt to predict the variable "winner" which had two outcomes. We use a decision tree due to the fact that its interpretability combined with the fact that they are robust to irrelevant features make them very suitable for classification problems, especially those with a large number of predictors. The preprocessing for this model included the removal of numeric predictors which had correlations greater than 0.8 as well as performing a log transformation of all numeric predictors in order to reduce skewness. Furthermore, the model used several different combinations of hyperparameters for cost complexity, tree depth, and minimum sample size to further improve the model as best as possible.

Model 2:
The logistic regression model was used with the "glm" engine and classification mode to predict a binary target variable, "winner." Logistic regression is a simple and efficient model for binary classification. The preprocessing recipe includes the normalization of all numeric predictors to help improve the model's performance and dummy variables for all categorical predictors, which allows the model to handle categorical data effectively and improves its performance. Median imputation was also used to handle missing values as well as the zero variance function which removes any predictors that have the same value in all samples and doe not provide any useful information for the model. There was no hyperparameter tuning performed for this model. These preprocessing steps with the regression model help ensure that the data is cleaned and suitable for modeling, leading to a more accurate prediction.

Model 3:
A random forest model is selected due to it being efficient and flexible for classification problems. The model included the ranger engine so that many trees could be utilized to prevent overfitting the data. Looking at the model's recipe, the step_log() function is used to compute logarithmic transformation on the total_votes variable since Figure 2 depicts its right-skewness. We also use the step_interact() function to explore any interaction terms between the "0001E" variable and variables that start with "C01". Specifically, this step of the recipe helps to explore any interaction terms between the total population estimate and the percentage of the population among various age and education levels. We also use the step_impute_median() function to

impute missing values. The step_corr() function is also utilized with a threshold of 0.85 to account for variables with extremely high correlations. Normalization of the variables also helps to decrease the influence of the varying scales among the numeric predictor variables. For tuning, the hyperparameter "trees" indicates the amount of trees being used for the random forest model.

Model 4:
A logistic regression model was selected due to the binary nature of the outcome variable, where the winner was either Trump or Biden. The model included a glm engine and classification mode to assess a linear model for the binary data. In the model's recipe, missing data was substituted by the median of its respective column, numeric predictors were normalized to reduce any weight differences between the different scales of the predictors, and a multicollinearity threshold of 0.9 was established for the numeric predictors to account for correlations between the predictors. Since the glm engine does not include hyperparameters, no tuning was performed for this model.

Model 5:
A decision tree model was selected because it utilizes variables from the observations to partition the data into smaller chunks. From there, the tree would use the explanatory variables to form binary splits on the variable to form branches so that the model can predict the response variable of winners with the two specific outcomes. This model was most plausible by having the complexity to form different trees and branches to understand how it can fit with the training dataset so it could be applied to the testing dataset. The recipe went through some specifications, such as substituting any missing values from both the numerical and nominal predictors. Hyperparameters of cost complexity, tree depth, and minimum number of samples were tuned to further the model-tuning process.

Table 1: Candidate Model Summary

Model / Author	**Model Identifier**	**Type of Model**	**Engine**	**Recipe Used**	**Hyperparameters**
Model 1 / Daniel	decision_tree	classification	rpart	recipe <- recipe(winner ~ ., data = train) %>% step_corr(all_numeric(),-all_outcomes(), threshold = 0.8) %>% step_log(all_numeric(),-all_outcomes())	cost_complexity = tune(), tree_depth = tune(), min_n = tune()
Model 2 / Khang	logistic_reg	classification	glm	data_recipe <- recipe(winner ~ ., data = train) %>%	

				step_rm(name) %>% step_dummy(all_nominal_predictors(), -all_outcomes()) %>% step_zv(all_predictors()) %>% step_normalize(all_predictors()) %>% step_impute_median(all_numeric_predictors ())	
Model 3 / Andy	rand_forest	Classification	ranger	vote_recipe <- recipe(winner ~ ., data = vote_train) %>% step_log(total_votes, base = 10) %>% step_interact(~ "0001E":starts_with("C02")) %>% step_impute_median() %>% step_corr(all_numeric_predictors(), threshold = 0.85) %>% step_normalize(all_numeric_predictors())	trees = 1000
Model 4 / Jessie	logistic_reg	classification	glm	pres_rec4 <- recipe(winner ~ ., data = pres_train) %>% step_impute_median() %>% step_normalize(all_numeric_predictors()) %>% step_corr(all_numeric_predictors(), threshold = 0.9)	
Model 5 / Chris	decision_tree	classification	rpart	recipe(winner ~ ., data = train) %>% step_impute_mean(all_numeric_predictors()) %>% step_impute_mode(all_nominal_predictors()) %>% step_smote(winner)	cost_complexity = tune(), tree_depth = tune(), min_n = tune()

Model Evaluation and Tuning

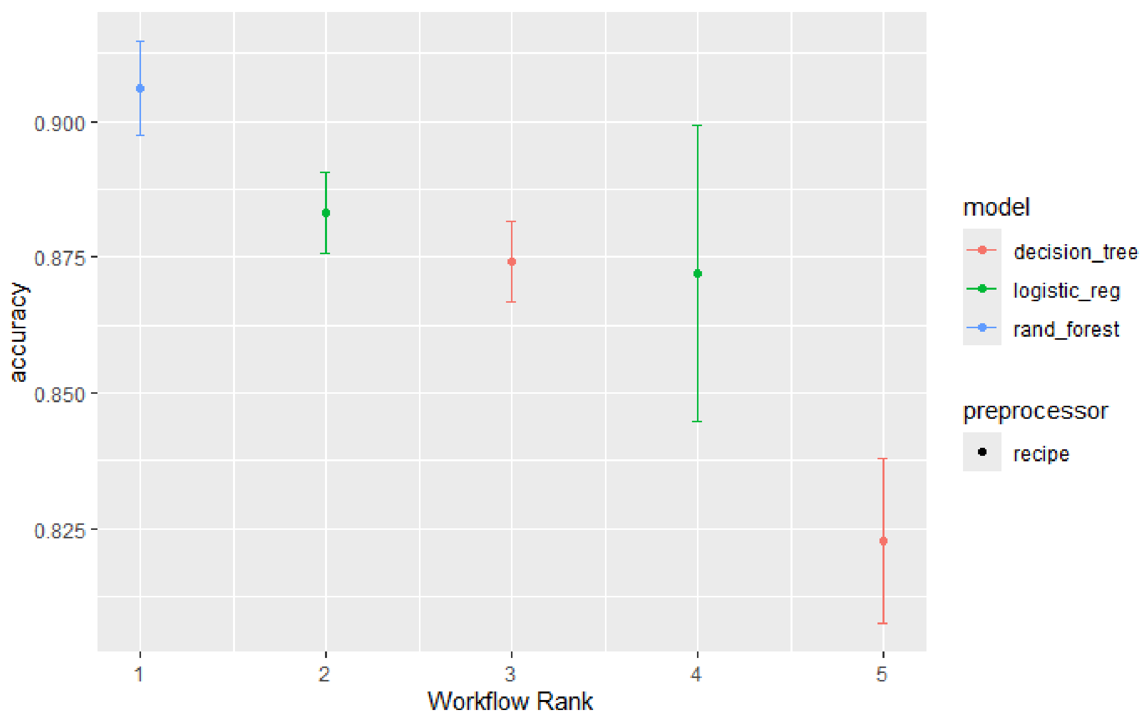
For the 5 models, v-fold cross-validation was performed to split the training dataset. By performing v-fold cross-validation, the produced regression models can be controlled on how cross-validation and resampling will perform. Specifically, we set v = 10 and also set the strata argument to be the “winner” variable from our data set. From there, the performance metric is calculated for each of the models to determine the accuracy of the models shown in Table 2.

Table 2: Performance of Models

Model	**Model Identifier**	**Metric score**	**SE**

Looking at the table above, Model 3 has the highest metric score, with its score being approximately 0.90732699.

Autoplot



We constructed an autoplot of the five models respectively indicated in Table 2, ranking them based on the accuracy metric. Model 3, which used a random forest model, was ranked the highest in accuracy, with an average score hovering slightly above 0.9.

Discussion of Final Model

The final model selected was Model 3. As mentioned earlier, Model 3 had the highest metric score of approximately 0.90732699. This model is a random forest model using the "ranger" engine and the "classification" mode. The random forest model has many strengths, which include being flexible and accurate for large data sets. The random forest model can be accurate as it utilizes a large number of decision trees to generate predictions for the output variable. The ranger engine helps create these decision trees and makes the final prediction by combining all the predictions from every tree used. Finally, the classification mode allows the model to follow tasks made for classification. Specifically, in random forest classification, many different trees are generated to use different and random subsets of the data being observed. Predictions will be made for each tree, and then the most popular result will be used as the final prediction after combining the individual results. Another advantage of a random forest model is that using many decision trees will decrease the chance of overfitting. Since the data has many observations, overfitting is an even larger issue as it can model with the current data accurately but not modeling predictions for unseen data. Also, an increased chance of overfitting will consequently increase the model's variance. Therefore, it is imperative to use a model like the random forest model to prevent the overfitting problem.

For the model's recipe, certain steps were utilized to improve the accuracy of the predictions. First, logarithmic transformation was performed on the "total_votes" variable since its values have a right-skewed distribution. Additionally, we were given the task of predicting the winning candidate of the 2020 US election with demographics and education levels as the predictor variables. Therefore, exploring the interaction terms between the total population estimate and the percentage of the population among various age and education levels is useful as interactions explore deeper insights in the relationships between certain predictor variables. Including interaction terms helps increase the accuracy of the model as it explores these more complex relationships between variables. Values for some of the predictor variables are also missing. Therefore, through the recipe, the model imputes the median value of the column to account for this issue. We impute rather than remove the columns since only a small percentage of observations have missing values. The next step in the recipe involves removing variables that have considerably high correlations with one another, with the specific threshold being 0.85. This is conducted to reduce the risks of overfitting and multicollinearity. Finally, the last step involves normalizing all the numeric predictors. As mentioned in the "Pre Processing" section earlier, normalization involves scaling numeric data to have a mean of zero and a standard deviation of one. This helps enhance the model's accuracy as it reduces the influence of any predictor variables that have much larger scales.

Hyperparameter tuning was also briefly used by adjusting the number of trees. Since the data set that was used is complex with many predictor variables and is also a large set covering 3,111 counties in the United States, using a high number of trees is more ideal to create accurate predictions. Therefore, setting the number of trees to be equal to 1,000 is desirable for optimal performance.

Although there are many strengths and advantages to Model 3, there are also some weaknesses that lead to possible improvements to further enhance the model. The complexity of random forest models makes it so that it is difficult to interpret them. Therefore, understanding specific, individual features can be challenging. Additionally, random forest models can be computationally intensive, as training a high number of trees can be time-consuming, especially since the observed data set is large. Hyperparameters in random forest models are also sensitive. An improvement for the interpretability disadvantage includes fitting a single decision tree alongside the model to help approximate it. Although this singular tree will not capture the complexities of the model, it can still provide a simpler, more interpretable summary of the data. To improve the random forest model being computationally intensive, it is possible to compromise the number of trees by fitting a high enough number of trees for accuracy, but not to the point where the value involves extreme computations. Finally, sensitive hyperparameters can be improved by utilizing methods such as grid search and random search to find the optimal hyperparameters more efficiently. These improvements will help the model be more interpretable and precise in calculating the predictions for the outcome variable.

Appendix

Daniel: Pre processing, exploratory data analysis, candidate models

Khang: Pre processing, candidate models, model evaluation

Andy: Exploratory data analysis, candidate models, discussion of final model

Jessie: Exploratory data analysis, candidate models, autoplot

Chris: Introduction, exploratory data analysis, candidate models