Presidential Proportions:
Predicting Percentage of Democratic Voters in the 2020 United States Presidential Election

Andra Velea, Aryan Mistry, Mateo Umaguing
University of California, Los Angeles
Professor Miles Chen, STATS 101C
July 29th, 2022

1 Introduction

The purpose of this project was to act as a statistician to find a valid model with significant variables that best predicts the percentage of voters in counties who voted for President Joe Biden in the 2020 United States presidential election. The provided training dataset contained one response variable, 213 predictors, and 2331 observations based on data gathered from demographic and education estimates made by the US Census Bureau, as well as the MIT Election Lab. In line with trends found by the popular American poll website, FiveThirtyEight[1], as well as the Pew Research Center[2], we expect White, Black, Asian, and Hispanic or Latinx groups, both male and female, to be associated with the response variable. Additionally, we suspected the age groups of voters to be significant predictors, as older generations tend to vote more conservatively and younger generations more liberally.

2 Exploratory Data Analysis (EDA)

One of the first steps taken was to reduce the dimensionality of the dataset. Specifically, we mainly focused on the percent estimates, allowing us to work with only 103 potential predictors, since the original data was redundant. Using our initial inferences for our predictors of interest, we plotted their relationship with the response variable. In particular, we examined the demographics associated with the highest margins in Democratic versus Republican votes. These were (according to Pew Research Center) African-American, Asian, Hispanic, White, college-educated Black, post-graduate, and younger voters. The results are shown in Figures 1, 2, 3, and 5.

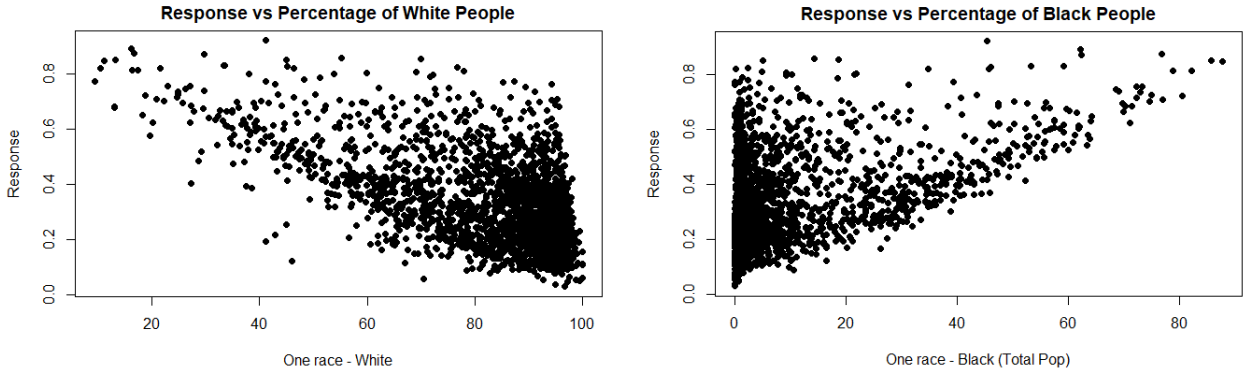

Figure 1. Response vs Percentages of White and Black people. In a somewhat intuitive way, the relationships seem to be inverses of each other; as the population percentage of white people increases, the percentage of Democratic voters decreases, while the opposite is true for the population percentage of black people.

[1] The Partisan, Gender and Generational Differences Among Black Voters Heading Into Election Day | FiveThirtyEight

[2] Behind Biden's 2020 Victory | Pew Research Center

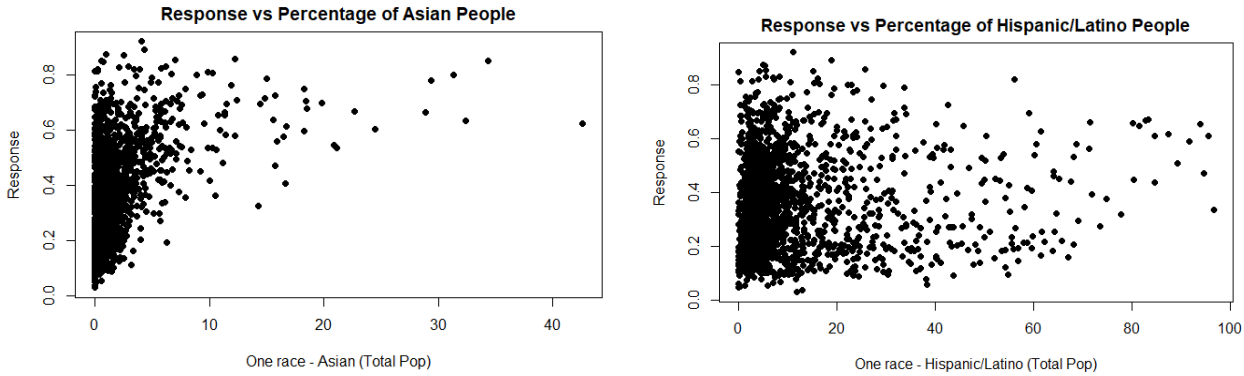

Figure 2. Response vs Percentages of Asian and Hispanic/Latino people. There doesn't seem to be much of a linear relationship for either of these variables, as the spread of the data is largely nonuniform and clustered towards the left. Indeed, the population percentage of these two variables are minorities, hence the clustering, but there does not seem to be a clear relationship for either regardless.

To further demonstrate the relationship between race and education, we interacted certain demographics that have a high tendency to vote for or against the Democratic candidate (Figure 3).

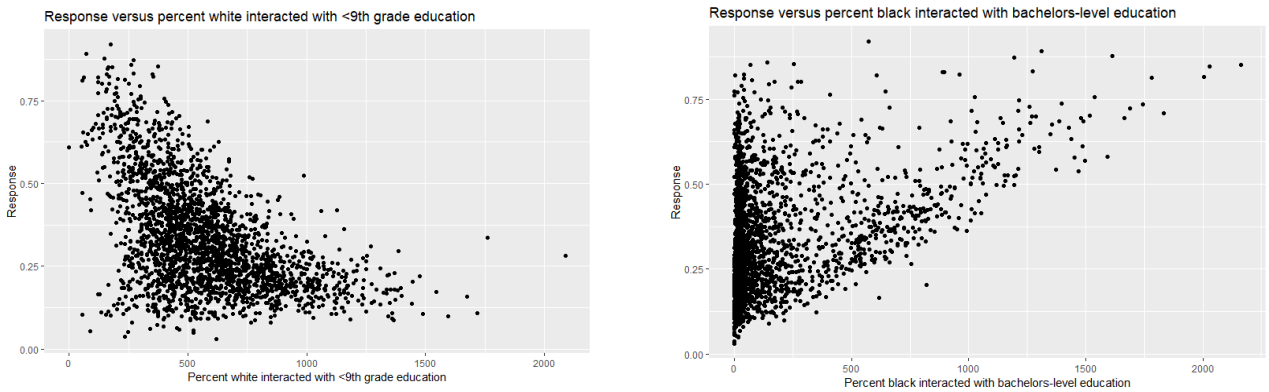

Figure 3. Response vs interactions between the white population and less than 9th grade education and the black population with bachelor's-level education. Both relationships seem highly nonlinear, indicating splines may be an accurate way to predict the response.

Evidently, all races do not follow a linear pattern. Even with a log transformation, the graphs do not improve (Figure 4); instead, the relationships become even more nonlinear and flexible. However, the outcomes indicate a correlation between the variables and the response.

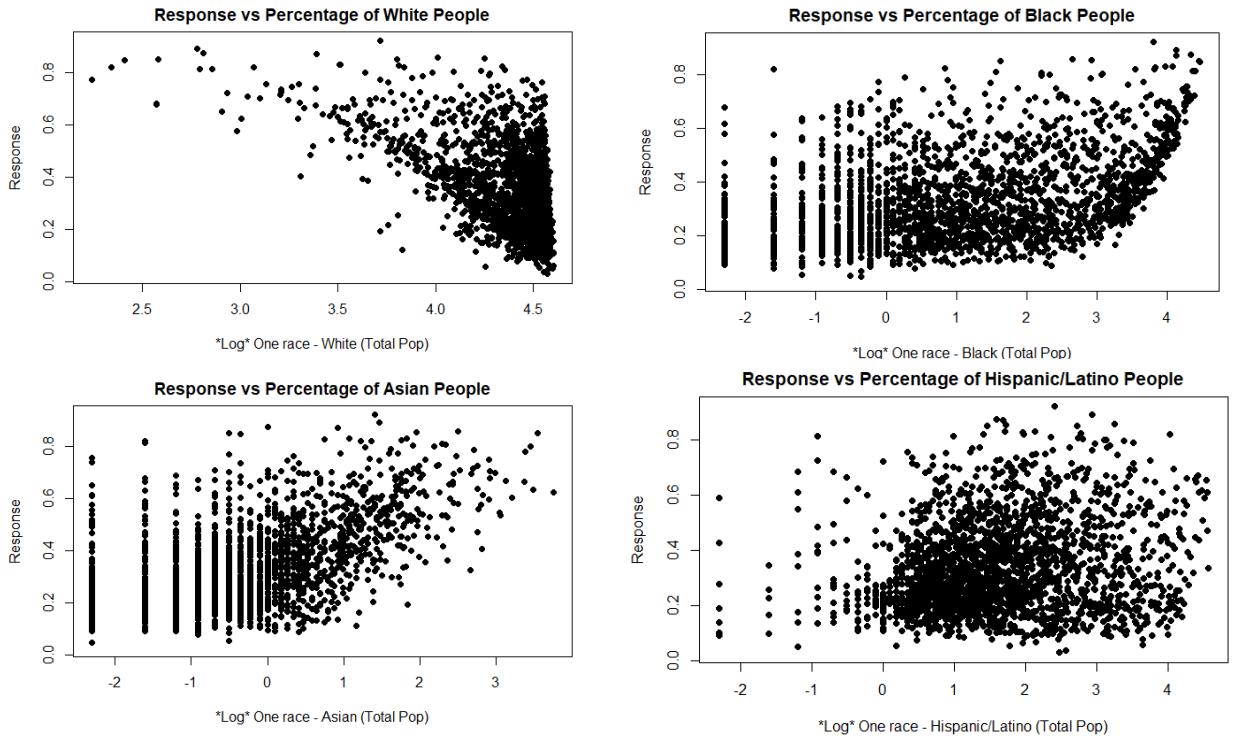

Figure 4. Log transformation of Figures 1 and 2. The results are highly nonlinear. It is evident splines could be useful. In the second graph, the values become meaningless as the population percentage goes negative.

While comparing races, we also wished to test our hypothesis regarding younger voter groups being more likely to vote democratically than older voter groups. Plotting the percentages of voters aged 25-34 against the response in Figure 5, we can see a positive correlation. This means that in counties where individuals aged 25-34 make up a larger percentage of the county population, the county is more likely to have a higher percentage of Democratic votes. The opposite effect was seen in voters aged 65-74. The graph in Figure 4 displays a negative correlation, which suggests that counties in which 65-74 year old voters are more prevalent are less likely to vote democratically. However, in the case of both race and age, the correlations between the predictors and the response are not exactly linear.

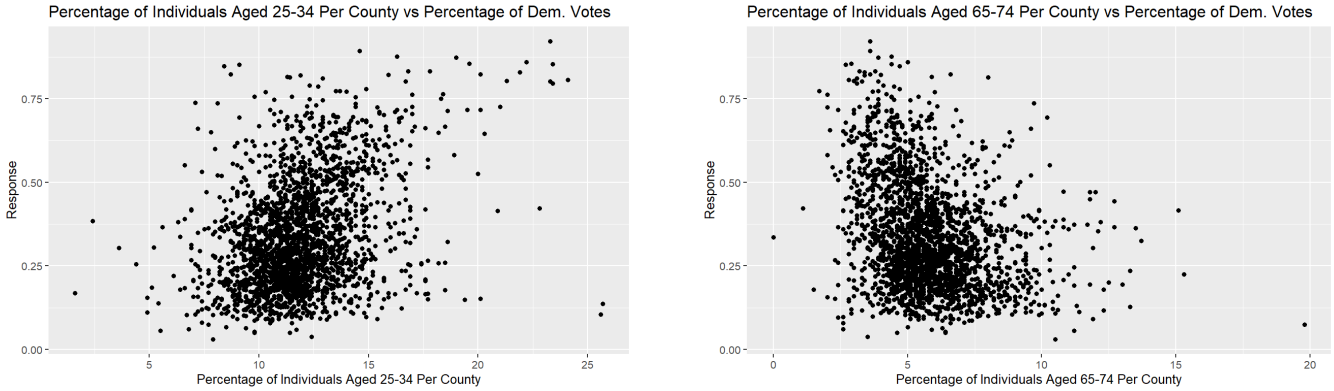

Figure 5. Response vs percentages of voters aged 25-34 and 65-74 years old. The younger voter group exhibits a positive correlation between percentage of younger voters in the country vs. percentage of Democratic votes in the county, while the older voter group exhibits a negative correlation between percentage of older voters in the county vs. percentage of Democratic votes in the county.

Furthermore, in our efforts to see which variables were the most important, we used a random forest on the reduced data. Then, using the varImp() function with conditional set to true from the package caret, we arranged the most important variables and compared them to the ones given from the importance() function from the randomForest library. Afterwards, to investigate any other non-obvious interaction effects, we visualized the correlation of the "total" most important variables. The output of the resulting corrplot is shown in Figure 6.

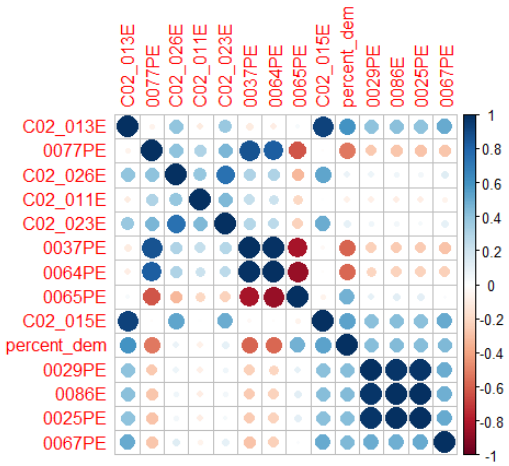

Figure 6. Corrplot of most important variables. Some of the noticeable correlated pairs: 0064PE and 0065PE, 0037PE and 0077PE, 0029PE and 0086E, 0037PE and 0064PE, etc.

The issue with the pairs derived from Figure 6 is the redundancy; for example, the pair 0037PE and 0064PE are strongly correlated, but they both represent the population percentage of whites with the only difference being the description of 0064PE as the race alone or in combination with others. However, our team focused primarily on the interactions discussed above. In fact, when the function findCorrelation() with cutoff set as 0.8 from the caret library was called, many of our initial guesses were accurately gauged as significant interactions. Thus we continued our exploration in formulating recipes.

3 Preprocessing/Recipes

For the first three models in Table 1, there was no preprocessing or recipe. For the fourth model, the predictors were selected in accordance with the trends gleaned from FiveThirtyEight polls and the cited Pew Research article. Namely, we decided to focus on voter gender, race, and age group. More specifically, we chose White, Black, Latino, Asian, and Pacific Islander voters of varying age groups ranging from 25 years old to over 85 years old. The predictors were centered and scaled using step_center and step_scale. No other transformations were performed.

Our final model included 103 predictors and 10 interaction steps. We included total population, housing units, and every predictor in percentage of the county population value. We excluded actual estimates since percentages were already a standardized metric and since the models did not perform better with their inclusion. Regarding interaction steps, we included the effects between the racial population percentages, education level percentages, and gender percentages. We chose to focus on these interactions based on the demonstrated voter margins as per Pew Research and suspected relationships per social media activity and election history. For example, non-college-educated white individuals voted heavily against the Democratic candidate, therefore we interacted the percentage of non-college-educated individuals with the white population percentage.

4 Candidate Models

Over the course of the project, we focused on four types of regression models for prediction: linear models, generalized additive models, random forests, and boosted trees. As a preface, our team used many different intermittent models; however, we decided to discuss the top five that significantly improved our RMSE since the rest did not have large effects between submissions.

We first fit a simple linear regression with the first seven predictors from a principal component analysis (PCA) with loadings >=0.1 on the entire dataset (213 predictors). Only the first seven were chosen since their correlation with the response was greater than 0.5, despite this being an egregiously low number of predictors relative to the size of the data. However, the initial run with all seven variables yielded insignificant coefficients. Using the vif() function from the car library and the output from the linear model, we removed the insignificant predictors and investigated potentially significant interactions. From here, we built models with more predictors and interactions deemed most important to the response.

The next model that significantly reduced the RMSE score was a generalized additive model (GAM). This did not include any interaction effects, but it took into account the nonlinearity of the most important variables discussed in Section 2. The resulting model was formed by running three ANOVA tests comparing two, four, and six degrees of freedom for each predictor and subsequently choosing the best value.

Moving on, the third model that better predicted the response was a random forest. This model was almost trivial; it tuned no hyperparameters, accepted the defaults from the randomForest library, and used every predictor with no interactions. It was essentially used as a “baseline” for the following intermittent models. It yielded better results than the GAM model, which is unsurprising given the accuracy improvement of random forests. Nonetheless, it was not the best model as it had much room for improvement. Specifically, we focused on tuning the hyperparameters and running cross-validation for an even better fit.

Generally, gradient-boosted trees outperform random forest; thus, we then attempted various xgboost models using the Tidymodels "xgboost" library. One initial model used voter gender, age group (divided into 10-year subdivisions from 25 years old to 85 years and older), as well as the races that were most likely to vote Democratic according to FiveThirtyEight. These included Whites, Blacks/African-Americans, Latinos, Asians, and Native Hawaiians. The model employed a grid search to tune several hyperparameters, such as the number of trees, learning rate, tree depth, sample size, and loss reduction. An example of the hyperparameter tuning process for the model "xgbsub2" can be found in Figure 7.

After similar intermittent models, we derived our final model. Here, we incorporated the gradient boosted trees model with the engine "xgboost" along with our interactions that we deemed important to the response. These were race percentages interacted with varying levels of education percentages across county populations and also male percentages. After setting up our recipe, we used the grid_latin_hypercube() function in order to perform a grid search to find the best hyperparameters. The results of all discussed models above can be found in Table 1.

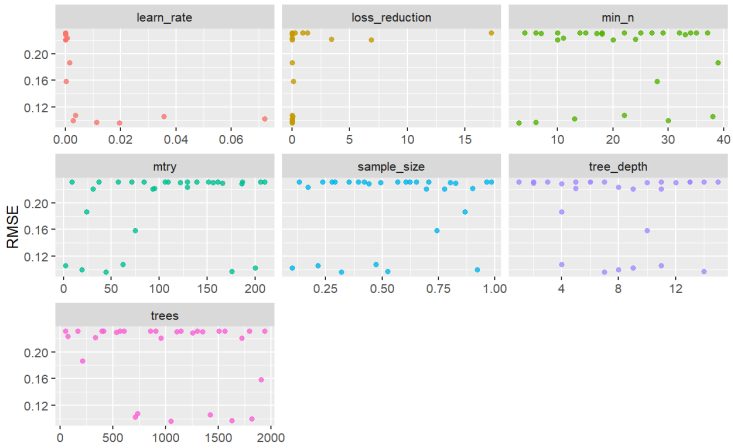

Figure 7. Multiple plots of the grid search process for determining the optimal model tuning parameters for the model xgbsub2. The parameters are plotted against RMSE.

Table 1. Summary of significant models.

Model Identifier	Model Type	Model Engine	Recipe/List of Predictors Used	Hyperparameters
firstdraft	Simple Linear Model	"lm"	PCA: C02_013E+0064PE+C02_015E+0077PE+C02_024E+C02_015E:C02_013E+ C02_013E:C02_0	N/A

			24E+0064PE:0077PE+ C02_015E:C02_024E	
gam_fit1	Generalized Additive Model	“gam”	s(C02_013E, df = 4)+s(X0077PE, df = 4)+s(C02_026E, df = 4)+s(C02_011E, df = 4)+s(C02_023E, df = 6)+s(X0037PE, df = 4)+s(X0064PE, df = 4)+s(X0065PE, df = 4)+s(C02_015E, df = 2)+s(X0029PE, df = 4)+s(X0086E, df=4)+s(X0025PE, df=2)+s(X0067PE, df=4)	N/A
train_rf	Random Forest	“ranger”	All predictors from reduced data	Default from randomForest library
xgbsub2	Boosted Tree	“xgboost”	All predictors from reduced data	trees = 1054 mtry = 44 min_n = 3 tree_depth = 7 learn_rate = 0.01960946 loss_reduction = 4.090772e-07 sample_size = 0.3201957
xgb2_mu (final model)	Boosted Tree	“xgboost”	0001E + 0086E + all predictors with “P” + all predictors with “C02” +	trees = 1000 mtry = 48 min_n = 21 tree_depth = 8 learn_rate =

			0037PE:C02_008 E + 0037PE:C02_010 E+ 0037PE:C02_021 E + 0037PE:0002PE + 0038PE:C02_010 E+ 0038PE:C02_021 E + 0038PE:0002PE + 0071PE:C02_010 E+ 0071PE:C02_021 E + 0071PE:0002PE	4.191037e-02 loss_reduction = 3.710132e-03 sample_size = 0.5318252

5 Model Evaluation and Tuning

The first model, firstdraft, was trivially the worst model run. Its RMSE score was above 0.09 and violated many assumptions of linear regression. The low accuracy of this model was apparent beyond the RMSE score, as our EDA showed clear signs of nonlinearity. Thus, our next candidate models involved treating these issues.

We then moved to more flexible model types, namely splines, general additive models, random forests, and ultimately gradient-boosted decision trees. The gam_fit1 model significantly lowered the RMSE compared to the linear regression. To assess its significance, an ANOVA was run for the two and the output supported the fact that splines better predicted the response, due to the increased flexibility of fit. The next model, train_rf, outperformed the GAM model, but no parameters were tuned nor were any interactions taken into account, indicating this model was not reaching its maximum potential. As such, the following two models focused on these problems.

To make sure our models were not overfitting to the training data and remained generalizable, we ran v-fold cross validation, tuned all hyperparameters, and compared all models against each other for the smallest RMSE. With the cross-validation process, we used both 5 and 10 folds as the optimum number, depending on the model that was used and the differences in processing power of the group's personal machines. The final model employed 10-fold cross validation. The model comparisons are summarized in Table 2, and the following autoplot can be found in Figure 8.

Table 2. Model comparisons.

Model Identifier	RMSE (Private score)	SE of RMSE
firstdraft	0.09244	NA
gam_fit1	0.07465	NA
train_rf	0.06614	NA
xgbsub2	0.06872	0.002448
xgb2_mu	0.05830	0.001591

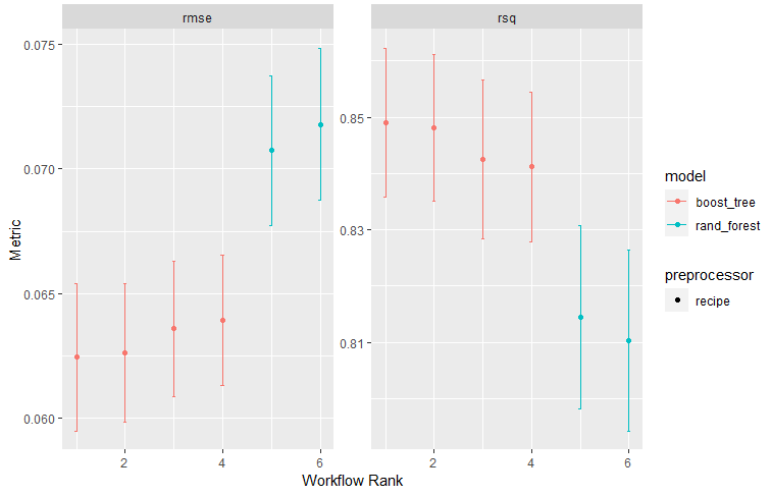

Figure 8. Metrics of boosted trees and random forest models used. XGBoost had the lowest rmse and highest R^2 values while random forests had higher rmse and lower R^2 values. The initial linear model and generative additive models were not created with tidymodels, however their metrics are detailed in Table 2.

6 Final Model Discussion

Ultimately, we used the xgboost model with the lowest RMSE on the public values of the testing set. Our team's final model was chosen for a variety of reasons. One of the most important was the number of hyperparameters that could be tuned alongside the internal regularization methods (both LASSO and ridge) that help prevent overfitting. The RMSE of this final model was also below 0.06, indicating a good prediction accuracy. The advantages of this model compared to the others included the smallest standard error of the cross-validation relative to intermittent models (omitted from table), and tuning of hyperparameters.

Some potential sources of error may stem from the lack of variable transformation and interaction effects. Namely, we did not transform any variables into categories, nor experiment

with many different interactions. In the future, it may prove fruitful to create a new variable out of others in the data; for example, a categorical variable for different races. Additionally, we did not take advantage of the large flexibility of Tidymodels in regards to the recipes used on the data. Perhaps incorporating different step functions would have improved the RMSE of our model. Another variable or piece of data that might be useful is the number of people registered as each party, but that isn't disclosed in every state. Also, knowing the median income salary could predict the response better since income has historically been a predictor of political party affiliation. Despite these limitations, however, our model is ultimately a good predictor for the amount of democratic voters per county for the 2020 U.S. presidential election.

Appendix A: Final Annotated Script

```
library(tidyverse)
library(tidymodels)
library(xgboost)

# Load in data
dem_data <- read_csv("train.csv")
test <- read_csv("test.csv")

# select percentages, total population, housing units
dem_data4 <- dem_data %>% select(
  'percent_dem', '0001E', '0086E',
  contains("P"), contains("C02")
)

# model
xgb_model <- boost_tree(mode = "regression",
                        engine = "xgboost",
                        trees = 1000,
                        mtry = 48,
                        min_n = 21,
                        tree_depth = 8,
                        learn_rate = 4.191037e-02,
                        loss_reduction = 3.710132e-03,
                        sample_size = 0.5318252
)

# recipe
dem_recipe <-
  recipe(percent_dem ~ ., data = dem_data4) %>%
  step_interact( ~ `0037PE`:C02_008E) %>% # white * 9-12 edu
  step_interact( ~ `0037PE`:C02_010E) %>% # white * some
college
  step_interact( ~ `0037PE`:C02_021E) %>% # white * bachelors+
  step_interact( ~ `0037PE`:`0002PE`) %>% # white * male
  step_interact( ~ `0038PE`:C02_010E) %>% # black * above
  step_interact( ~ `0038PE`:C02_021E) %>%
  step_interact( ~ `0038PE`:`0002PE`) %>%
  step_interact( ~ `0071PE`:C02_010E) %>% # hispanic * above
  step_interact( ~ `0071PE`:C02_021E) %>%
  step_interact( ~ `0071PE`:`0002PE`)

xgb_submitted <- read_csv("xgb_mu.csv")

# create xgb workflow
set.seed(1)
```

```
xgb_test_wflow <-
  workflow() %>%
  add_model(xgb_model) %>%
  add_recipe(dem_recipe)

# fit model to test data using workflow
set.seed(1)
xgb_test_fit <- fit(xgb_test_wflow, dem_data)

# predict values from test data
xgb_test_results <-
  test %>%
  select(id) %>%
  bind_cols(predict(xgb_test_fit, new_data = test))

# create output file
names(xgb_test_results) <- c("Id", "Predicted")
write.csv(xgb_test_results, 'xgb_test_mu.csv', row.names =
FALSE)
```

Appendix B: Team Member Contributions

Andra Velea

Initial paper rough draft, all sections of paper, intermediate models, research

Aryan Mistry

All sections of paper, intermediate models, research

Mateo Umaguing

All sections of paper, final model, final script, research