**David Sivieri**

**STAT8480 – Project Phase 2**

**12/20/2022**

**Introduction:**

I have had a longstanding interest in U.S. politics and political science. When given the opportunity the chose a data mining project, I decided to research a topic that would involve analyzing recent U.S. election data. Since we learned a great deal about binary classifiers this semester, I wanted to fit my project into that particular model. Ultimately, I decided to study the 2018 U.S. mid-term elections as a classification problem. By way of background, every two years congressional elections are held in the U.S. The country is divided into 435 geographic areas, with each corresponding to one congressional seat. The seats are apportioned to states based on population, with every state being guaranteed at least one seat. Within states, each congressional district should be roughly equal in population size. Every ten years, congressional seats are reapportioned based on the decennial census results.

If you look at congressional election results, you realize that incumbent parties get re-elected in the United States at a very high rate – in 2018 there was a 90% re-elect rate. The majority of congressional districts continuously vote for the same party and this occurs over very long stretches of time. There are some obvious explainers for this – candidate incumbency bias is one commonly cited reason. For example, once a district has been voting for a particular candidate over many cycles, they tend to vote for the same candidate and the incumbency bias gets stronger over time. However, I do believe that there are other factors at work. I wanted to test and see how strongly the demographic and socio-economic characteristics could predict partisan lean. Therefore, I decided to build a model that could predict which political party would represent a district based on demographic factors alone.

There are a number of reasons one may want to build such a model. Political scientists interested in the makeup of the party’s political coalition would be one user group. For example, they might want to know how much does racial makeup or education level play in influencing a district’s partisan lean. Are geographics factors – such as how urban or rural a district is influencing the partisan bias? By looking at the factors which most influence voting patterns, one can gain an understanding of the demographic makeup of the two parties national political coalition. Another potential use case for the model would be political professionals working for the parties. When deciding whether to invest money in a particular district’s election, it would be helpful to know if your party is a good fit for the district or if the incumbent party is a poor fit. For example, imagine there is a county currently represented by a Republican, but which our model says has a 70% chance of being represented by a Democrat. From the Democratic party’s perspective, it may make sense to invest a significant amount of money into defeating the incumbent Republican because the model determines they are an un-natural fit for the district’s demographics. This model can also give us insight into the previously mentioned decennial redistricting process. The redistricting process is controlled by different parties in different states and there are perennial disagreements about parties “gaming” the process to favor their own incumbents. By understanding what combination of demographic factors would give one party a durable advantage, one can look at proposed district maps and determine if they were drawn in a neutral manner or if they were drawn in a manner that gives one party or the other a distinct partisan advantage.

**Data Description:**

I used combined two distinct datasets in this project – election data and census/demographic data. Both are described below.

*Election Data:*

The election data was downloaded from a link provided by the MIT Election Lab (<https://electionlab.mit.edu/>). The particular dataset used is maintained by data scientist Stephen Pettigrew and downloadable from: <https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/UYSHST>. This dataset contains county-level election results for the 2018 congressional election. It lists the state, county, congressional district, and a unique federal government code called the “fipscode”, which identifies the county. It also lists the vote totals for the Democratic, Republican, and third-party candidates. From this county-level dataset, I was able to produce two datasets – a district level set and a county level set. Some county records appeared multiple times in the dataset because some districts cross county lines and multiple candidates run in different elections in the same county. In this case, I decided to sum together the vote totals of different candidates of the same party. This would prevent single counties getting over-represented in the dataset. Essentially, votes for different candidates of the same party were aggregated together. Obviously, every candidate is unique and has unique characteristics and incumbency status which could affect votes for them. However, using county-level data, I could create a dataset with over 3100 input rows as opposed to just 435 with the district set. In order to get access to this greater amount of input variation, I decided it was worth the potential downsides involved in preparing county data. I also decided to keep election data on counties and districts where candidates ran unopposed. I decided that the very reason a candidate runs unopposed is related to party strength in the district and therefore valuable for the model.

Preparing the election data left me with these fields: "state", "fipscode", "dem\_prop", "rep\_prop", "other\_prop", "party\_winner", "rep\_won". These will be described in detail in the data dictionary table below.

*Demographic Data:*

The source of the demographic and socioeconomic data is the American Community Survey (ACS), which is administered by the U.S. Census bureau. This is a demographics survey program that randomly samples 3.5 million households per year and provides point estimates and 90% CI’s on all numeric statistics. I used the 2014-2018 (ACS 5-Year Estimates). Initially, I made use of a subset of about 20 variables of data provided by MIT’s Election Lab in a context spreadsheet. These were variables that were commonly studied by political scientists and were known to relate to partisan preference. For my final analysis, I made use of an R package called **tidycensus**, which allows anyone to get direct access to the U.S. Census bureau’s API (by obtaining a free API key.) The API gives you access to thousands of variables from multiple years that the U.S. Census bureau maintains. I added some variables to the data provided by MIT. I wanted data on income inequality in the district, and the Census bureau actually does collect this information and makes a gini score available called “gini\_idx.” I also wanted a finer grained information about education level, particularly of white voters. My earlier analysis had shown that the education level of white voters in a district is a particularly powerful predictor of partisan preference. I was able to create variables for the proportion of the white population over age 25 that has no H.S. education, no bachelor’s degree, and a bachelor’s degree or higher. In phase 1 of this project, I also had a categorical variable that described population density. I was since able to find a numerical variable for population density. Since population density is meaningful and correlates with partisan affiliation, I thought it would be better for the model to have this variable in a numeric form. It was created by dividing the land area in square miles of the district by the total population

The tables with election data are joined to the demographic data by using the FIPS code for county, which is a standard numeric code for identifying geographic entities used by the federal government. The 20 input variables, target variable, and unused variables are described below.

**Data Description Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Model Role | Measure Level | Description |
|  |  |  |  |
| state | ID | Nominal | State name |
| county | ID | Nominal | County name |
| fipscode | ID | Nominal | County FIPS code |
| dem\_prop | Rejected | Interval | The proportion of the vote that the Democrat won in the county or district. |
| rep\_prop | Rejected | Interval | The proportion of the vote that the Republican won in the county or district. |
| other\_prop | Rejected | Interval | The proportion of the vote that the third parties won in the county or district. |
| party\_winner | Rejected | Nominal | Name of the party that won. |
| rep\_won | Target | Binary | Winner Code: 0 = Democrat, 1 = Republican |
| total\_pop | Input | Interval | Total population. |
| cit\_vote\_pop | Input | Interval | Citizen voting-age population. |
| under\_29\_prop | Input | Interval | Population 29 years or under as a proportion of total population. |
| over\_65\_prop | Input | Interval | Population 65 years or older as a proportion of total population. |
| female\_prop | Input | Interval | Females as a proportion of total population. |
| median\_age | Input | Interval | Median age of the population. |
| white\_prop | Input | Interval | Non-Hispanic whites as a proportion of total population. |
| black\_prop | Input | Interval | Non-Hispanic blacks as a proportion of total population. |
| hispanic\_prop | Input | Interval | Hispanics or Latinos as a proportion of total population. |
| foreignborn\_ prop | Input | Interval | Foreign-born population as a proportion of total population. |
| med\_HH\_inc | Input | Interval | Median household income in the past 12 months (in 2018 inflation-adjusted dollars.) |
| unemp\_prop | Input | Interval | Unemployed population in labor force as a percentage of total population in civilian labor force. |
| no\_hs\_prop | Input | Interval | Population with an education of less than a regular high school diploma as a proportion of population over age 25. |
| no\_bach\_prop | Input | Interval | Population with an education of less than a bachelor's degree as a proportion of population over age 25. |
| grad\_plus\_prop | Input | Interval | Population with an education of more than a bachelor's degree as a proportion of population over age 25. |
| wht\_no\_hs\_prop | Input | Interval | White population with an education of less than regular high school as a proportion of population over age 25. |
| wht\_bach\_plus\_prop | Input | Interval | White population with an education of a bachelor's or higher as a proportion of population over age 25. |
| gini\_idx | Input | Interval | Summarizes the dispersion of income across the entire income distribution. |
| rural\_prop | Input | Interval | Rural population as a proportion of total population. |
| density | Input | Interval | Number people per square mile of land. |

**Data Issues:**

*Missing Data*

The demographic data from the U.S. Census Bureau was very complete with one exception – the **rural\_prop** variable. 23 districts and 93 counties had very significant missing data with this one variable. Because the prior phase of this project had indicated that this variable was quite significant, I did not want to attempt to impute it. Since the overall proportion of missing variables was small in relation to whole dataset, I decided to remove these rows, so I could continue to use the variable. With those counties and districts removed, the final county dataset had 3107 records, while the district level dataset had 412 districts.

*Balance*

One of my datasets has a target variable imbalance. As can be seen below, about 80% of the counties in the U.S. are Republican leaning. The districts are much more closely balanced. As will be described in the methodologies, I tried a couple different techniques to deal with this imbalance.

|  |  |  |
| --- | --- | --- |
|  | Districts | Counties |
| Republican Winners | 191 (46%) | 2445 (79%) |
| Democrat Winners | 221 (54%) | 662 (21%) |
| Total | 412 | 3107 |

*Non-normality*

I used the **basicStats** package to check the distribution of my inputs and see if any were highly non-normal. I inspected the skewness and kurtosis statistics. Because the logistic regression model was sensitive to non-normality and because all of variables have values greater than zero, I decided to use a BoxCox transformation on the input variables for the regression model.

**Methodology**

I decided to try four supervised learning binary classification models. The goal for each of these models was to predict the target variable (which is the party that won the most votes in the district or county.) The difference between the primary and secondary outcome was not relevant to me. The coding of 0 to the Democrats winning and 1 to the Republicans winning was entirely arbitrary and based on alphabetical order. I care equally about the sensitivity and specificity of the models. The model’s hyperparameters were tuned using misclassification or F1 (depending on the particular model.)

I decided to use try four different model styles:

* Decision Tree
* Logistic Regression
* Random Forest
* Artificial Neural Network

For some of these models, I tried four different datasets:

* District (which is roughly naturally balanced in terms of party winner)
* County (which is highly unbalanced in favor of Republicans)
* County Balanced (this is a down-sampled dataset with a perfect balance between the targets)
* County Alternative Threshold (an unbalance dataset with the threshold chosen by ROC curve)

The overall goal was to compare the various statistics related to model evaluation: Accuracy, Misclassification Rate, F1, Sensitivity, Specificity, Precision, and Area Under ROC Curve for all of the different models and datasets and see if there were significant enough differences to declare one model better than the others. A common random seed was used in the all the models so that I would be comparing models on a consistent train/validation set.

*Decision Trees*

I used a decision tree model on the four datasets that I mentioned above. I set my **cp** parameter to 0.0 to allow the trees to grow to a maximum length. I used a 70/30 train/validation split. For model pruning in three of the models, I used the misclassification rate to pick the best model. In the alternative threshold model, F1 score was used for pruning. The relevant statistics are shown in the appendix. Comparing the four datasets, it would seem that the county balanced model produced a good balance between sensitivity and specificity, while having a low misclassification rate.

*Logistic Regression*

For the logistic regression model, I tried three different variable selection methods using the BIC criterion: stepwise regression, forward regression, and backwards regression. Data was transformed using the BoxCox method. The misclassification rate was used to pick the best model, except in the case of the alternative threshold model, when the F1 score was used. The relevant statistics and plots are shown in the appendix. In this case, the county balanced and alternative threshold models seem to perform similarly, with a good balance of sensitivity and specificity.

*Random Forest*

For the random forest models, I used the same datasets as the other models, but instead of using an alternative threshold, I tried both externally down-sampling and internally down-sampling. For all models, I used 500 trees and tuned the **mtry** parameter (using a range of 2 to 10) using F1 score. I would say the externally balanced county data performed well with a good balance between sensitivity and specificity.

*Artificial Neural Network*

For my artificial neural network models, I tried three different datasets: district, county unbalanced, and county down-sample balanced. Input variables were scaled and backwards regression using the BIC criterion was used to select the best subset of variables for the ANN model. I used a model with 4 neurons in the hidden layer. The activation function was **tanh.** The loss function was **binary\_crossentropy** and the metric was **accuracy**. The activation function of the output layer was **sigmoid**. Very similar to the other model, the county balanced data seemed to have the best balance between sensitivity and specificity.

**Model Summary**

In each of the four model, the county balanced data seemed to produce a good balance between the various model evaluation statistics, so I decided to compile a table where I would compare all four models using county balanced data and see if one particular model stood out. The random forest data shown is for the externally balanced model to keep everything consistent.

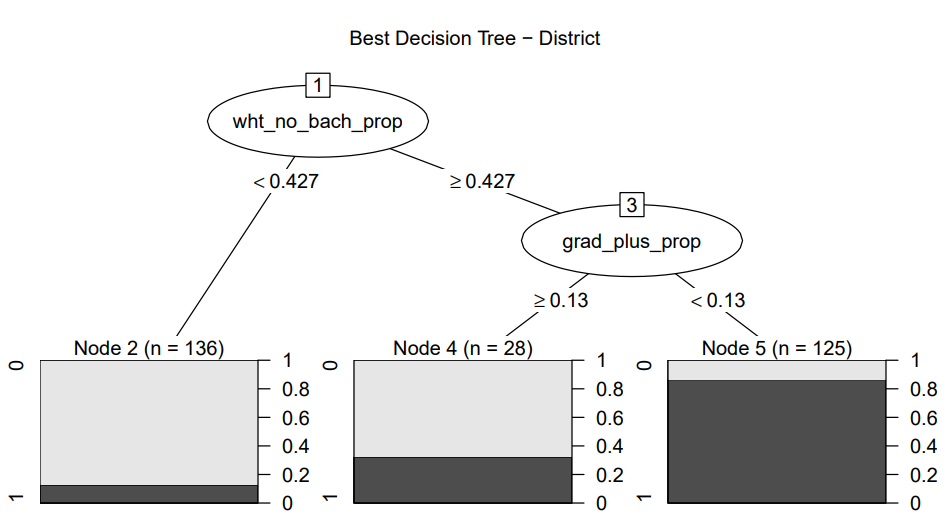
***Model Evaluation Parameters for County Unbalanced Data:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Decision Tree | Logistic Regression | Random Forest | ANN |
| Accuracy | 0.855 | 0.851 | 0.858 | 0.841 |
| Misc. Rate | 0.145 | 0.149 | 0.142 | 0.159 |
| F1 | 0.906 | 0.901 | 0.907 | 0.894 |
| Sensitivity | 0.884 | 0.862 | 0.873 | 0.854 |
| Specificity | 0.749 | 0.809 | 0.804 | 0.794 |
| Precision | 0.928 | 0.943 | 0.943 | 0.939 |
| AUC | 0.816 | 0.836 | 0.839 | 0.824 |

From comparing the four models, it is difficult to choose one best model. Most of the important evaluation statistics are very close and the probably all well with the margin of error when considering that different train/test splits will produce different results. If I had to pick one, I would suppose I would choose the random forest because it has the highest accuracy and F1 score as well as a good balance between sensitivity and specificity.

**Variable Importance**

There are many ways to evaluate the variables most determinative of partisan bias. I will show variable importance metrics for some of the models in the appendix. All of the models seem to determine that race and education level variables are quite correlated with partisan preference. For purposes of illustration, I will display the simplest model that I produced (decision tree – district), yet which has an accuracy of over 80%.

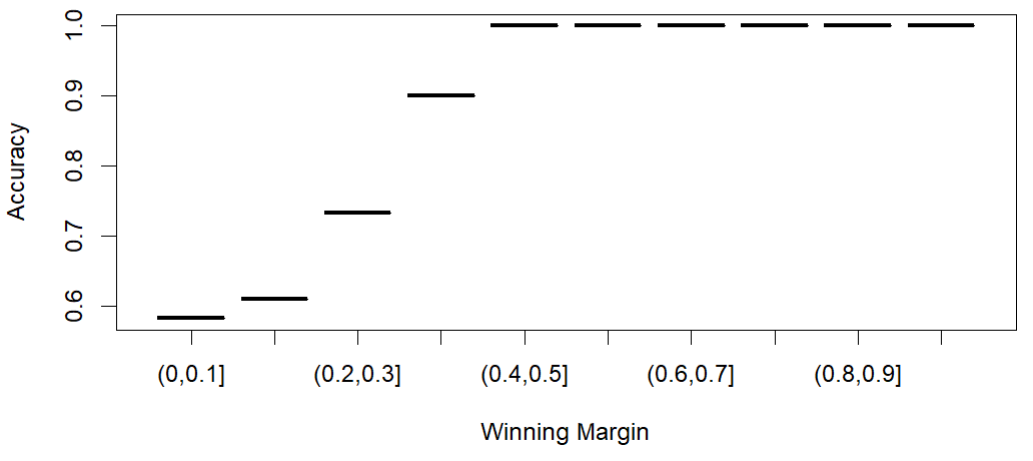


As can be seen, with just two variables, the model can predict fairly accurately the party that will represent a district. Essentially, if the proportion of the population that is white without a college degree is below 43%, the district is highly likely to be democratic. If the proportion is higher, than the proportion of the population the has a graduate degree becomes highly determinative.

**Accuracy vs. Winning Margin**

I was curious at how well the model would perform when tasked with predicting results where the actual winning margin is important. I decided to choose my best random forest model for district data and plot the accuracy of the predictions based on the winning margin of the candidate. The model was trained on 70% of the district data. The below plot is generated from only the 30% of data used for validation. Essentially, on the right-hand side of the side of the x-axis are districts that are won by margins of 90% to 100%. As you move left on the x-axis, the winning margin shrinks until you hit nearly 50/50 districts. As can be seen, the model is highly accurate until you get to very balanced districts at which point the model accuracy drops to a little under 60%. I found this interesting, in that it shows that it’s not very difficult to predict the results of most districts because most districts are highly biased in one partisan direction.

**Accuracy vs. Winning Margin for Districts**



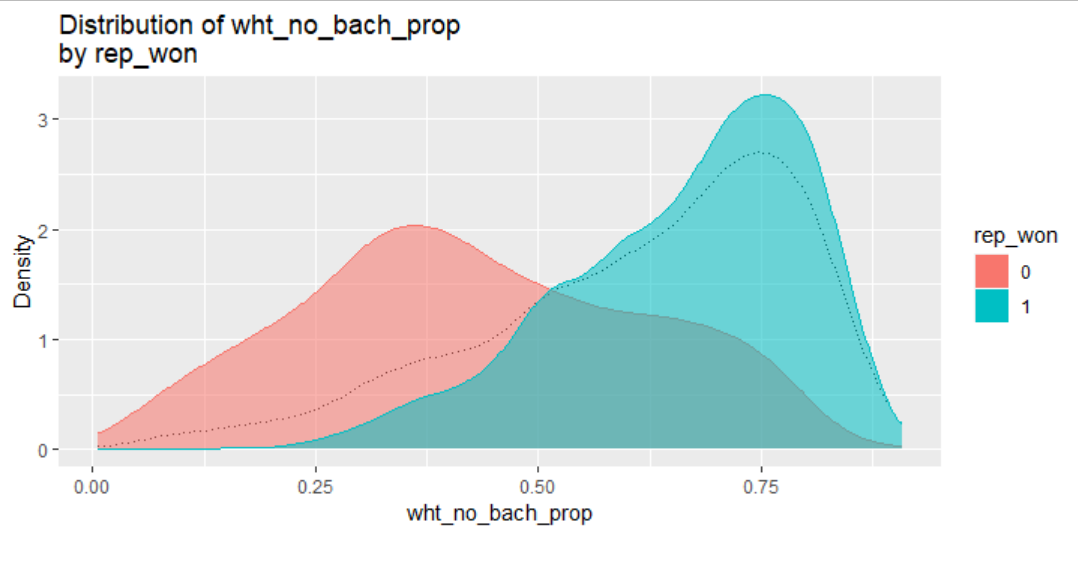
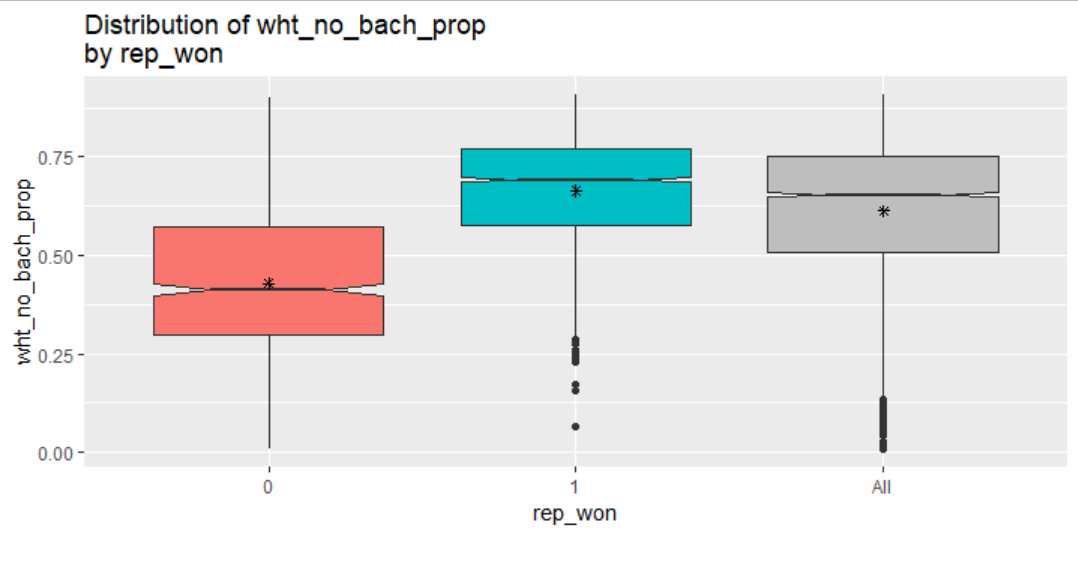
**Conclusions/Further Study**

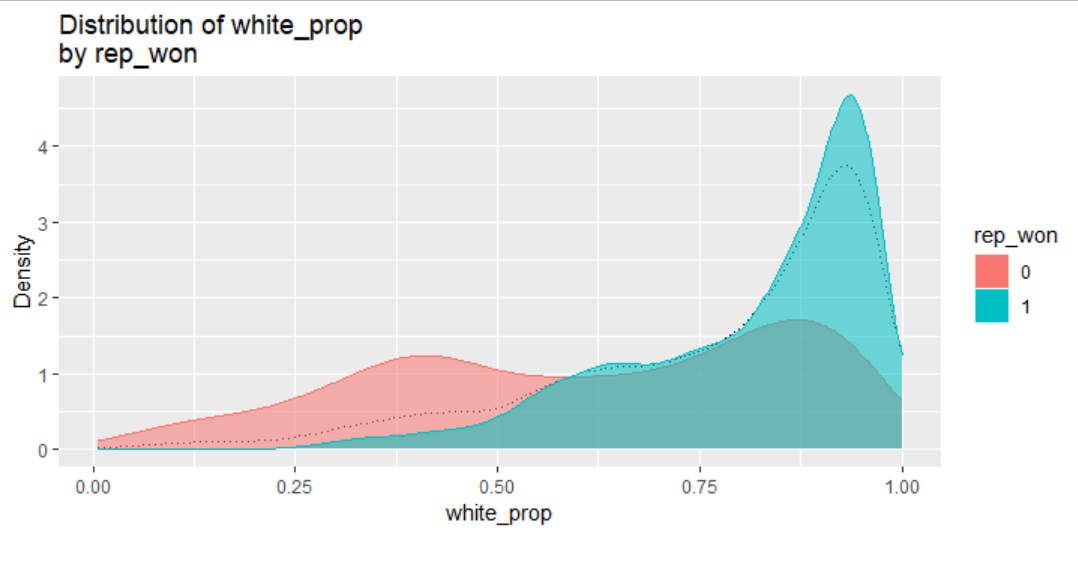
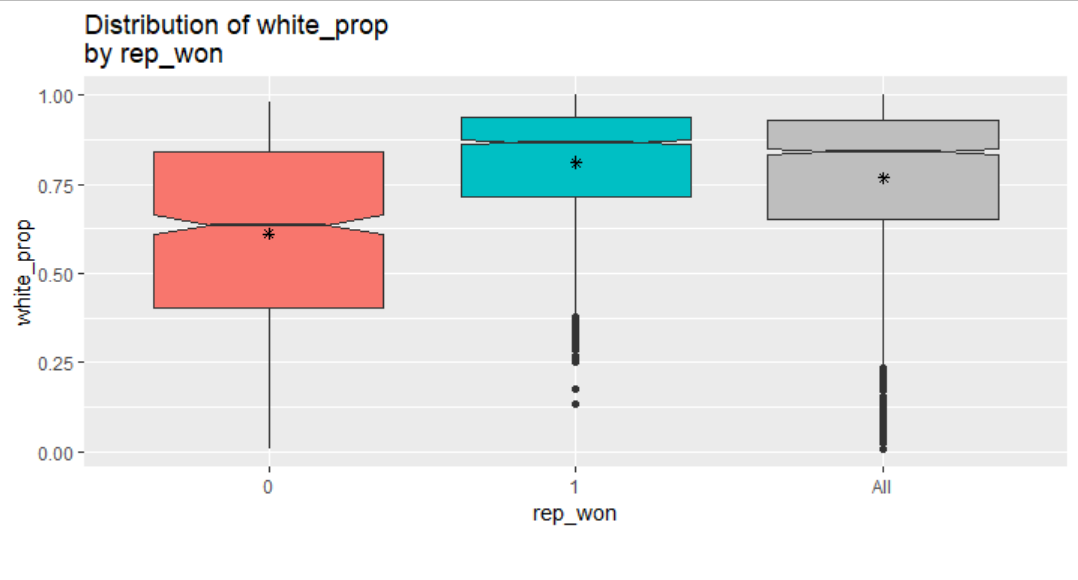
I found the results interesting, but ultimately somewhat disappointing from a prediction perspective. It seems that it is actually not very difficult to predict the election outcome in the vast majority of districts. A very simple model based on education level and race does very well. Most districts are simply very highly biased in one partisan direction or the other. What is difficult, however, is predicting the outcome of districts that have very close elections. In those cases, my model was only a little better than 50% accurate. Some possible ideas for future study would be trying to put more weight on very close districts or perhaps finding new input variables that better predict outcomes in closer districts.

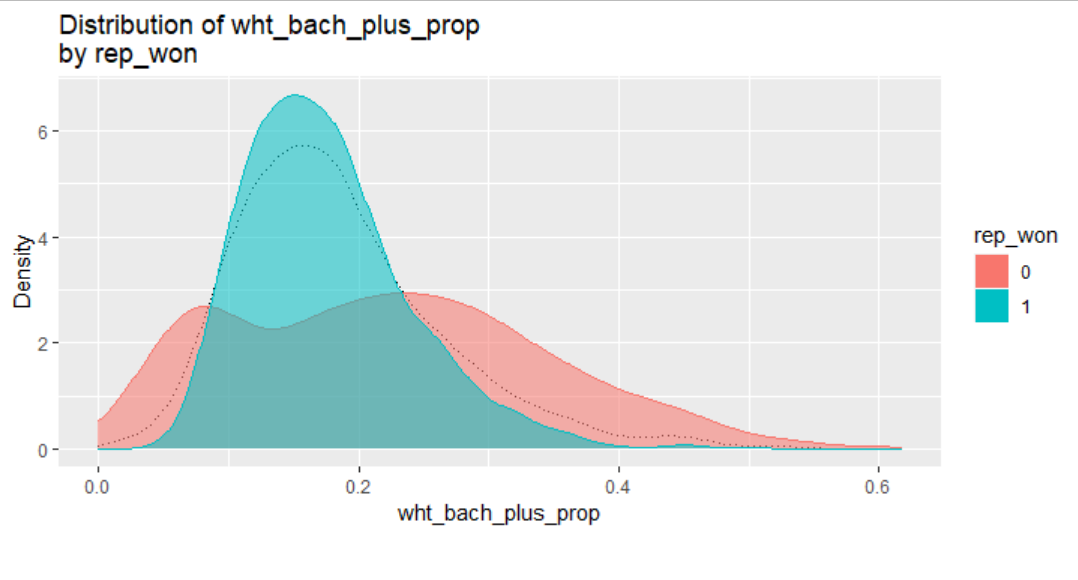
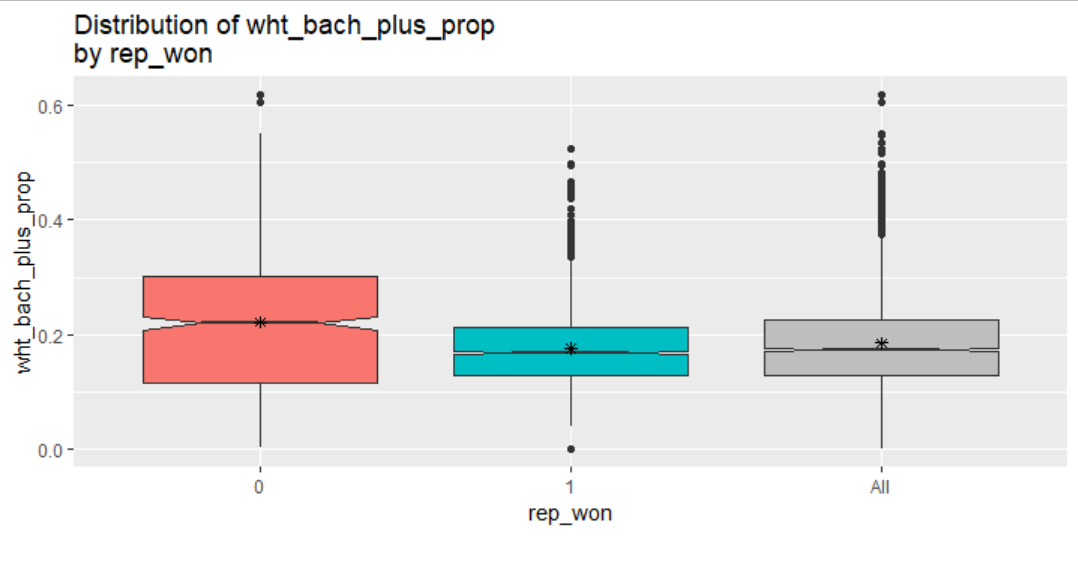
**Appendix**

In the first part of this appendix, I will include some boxplot and density plots of some of the variables that the models found were most important for predicting the target. Note that in all of these, the **rep\_won** variable corresponding to 1 is a Republican won county and 0 is a Democrat won county.

*County Level Variables grouped by party winner:*

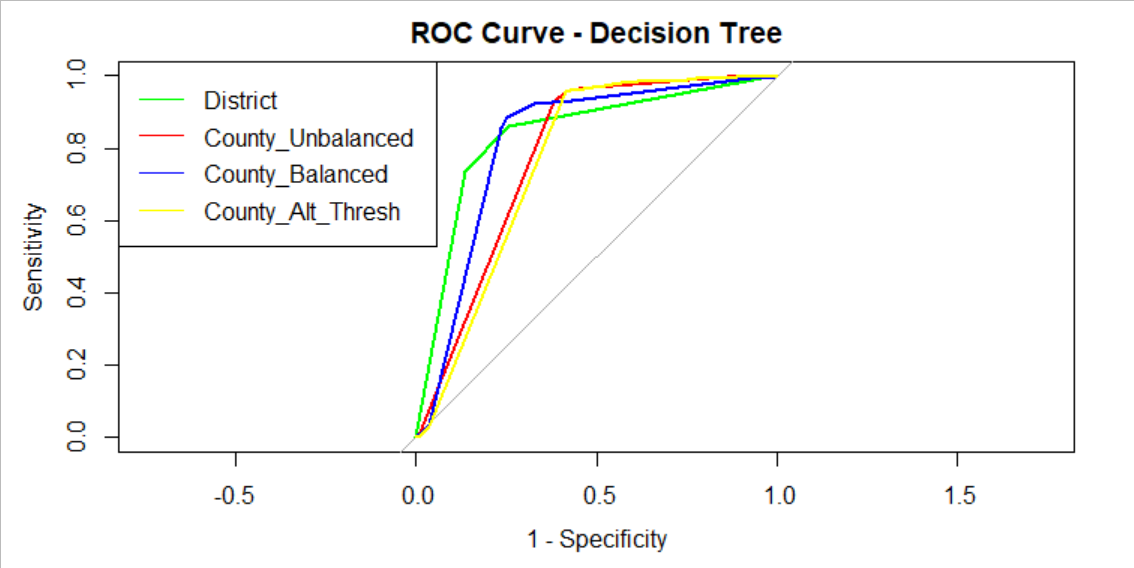


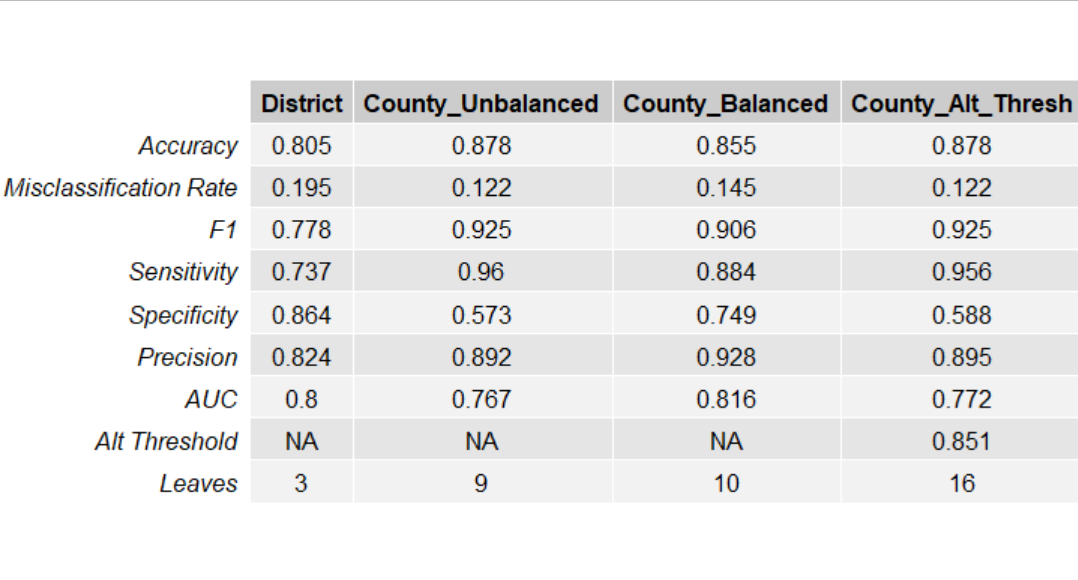




**Decision Tree Model Plots and Data**

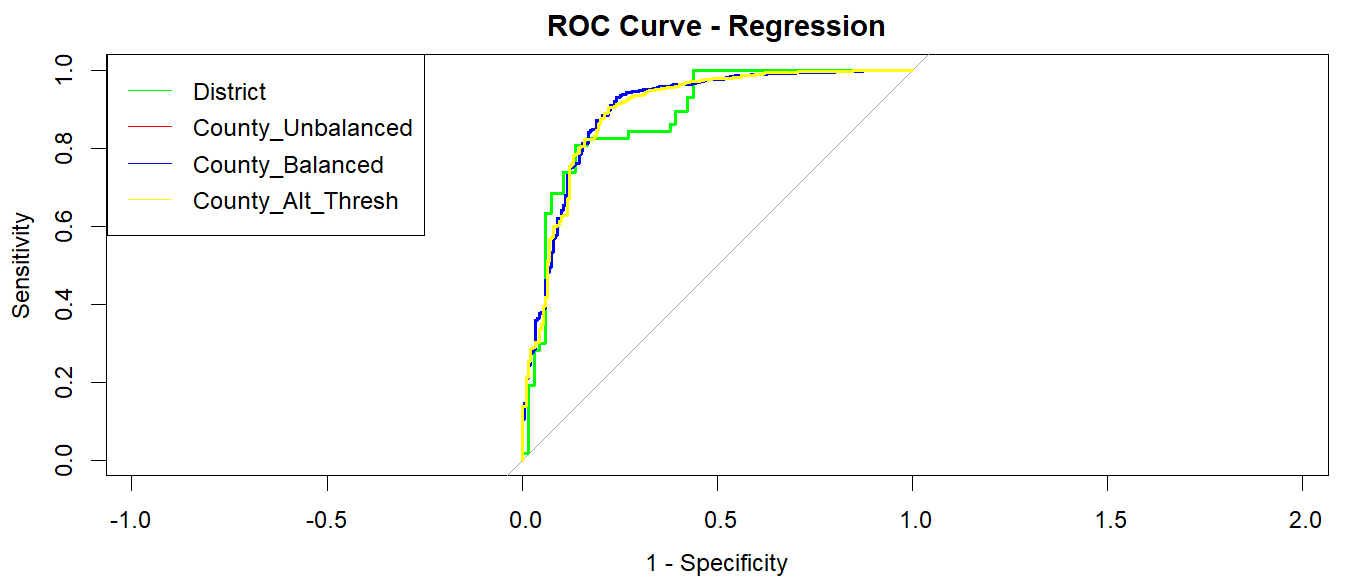
Below are the ROC curves for the four decision tree models used followed by a full summary of model evaluation statistics:

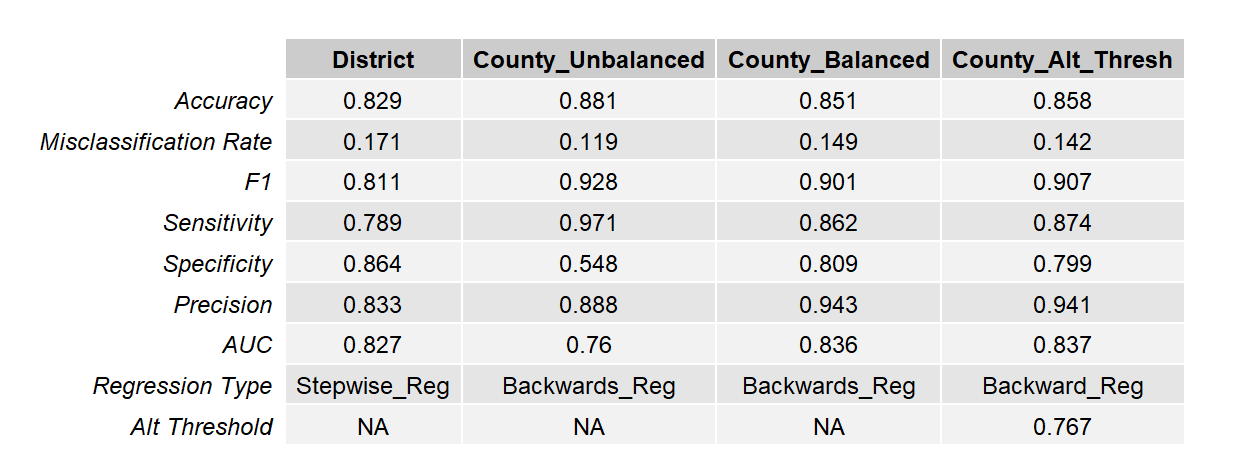




**Logistic Regression Model Plots and Data**

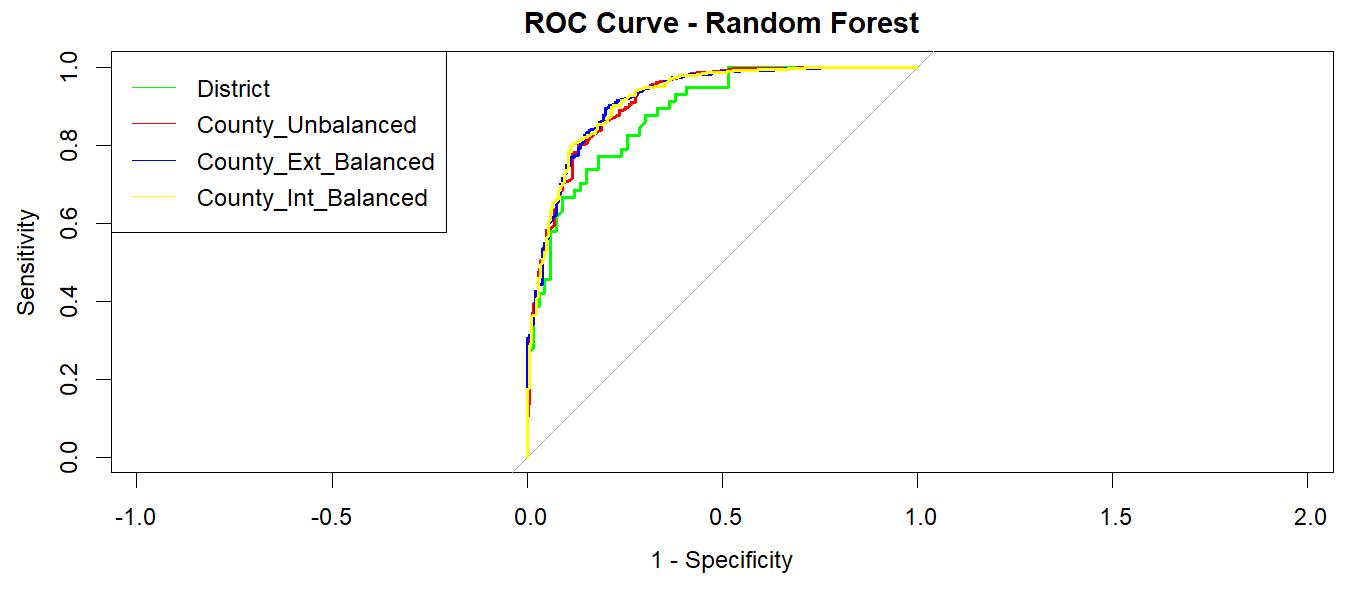
Below are the ROC curves for the four logistic regression models used followed by a full summary of model evaluation statistics:

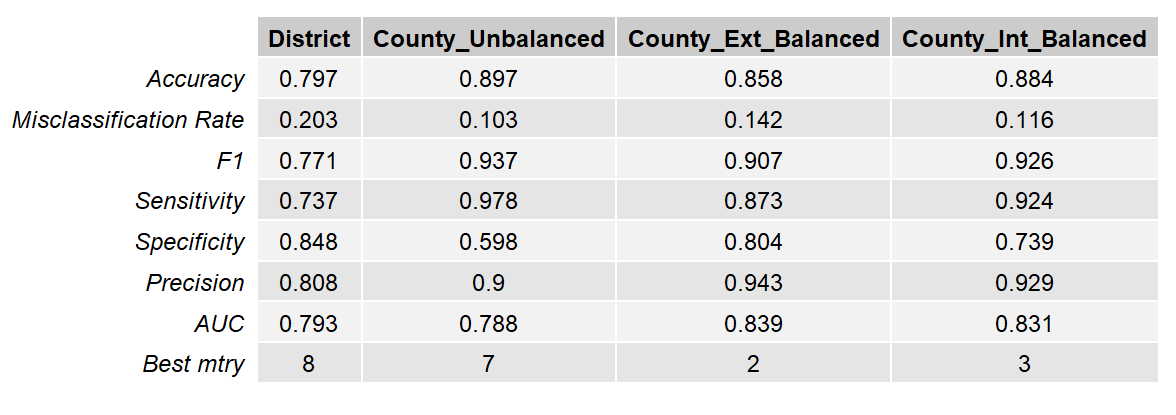




**Random Forest Model Plots and Data**

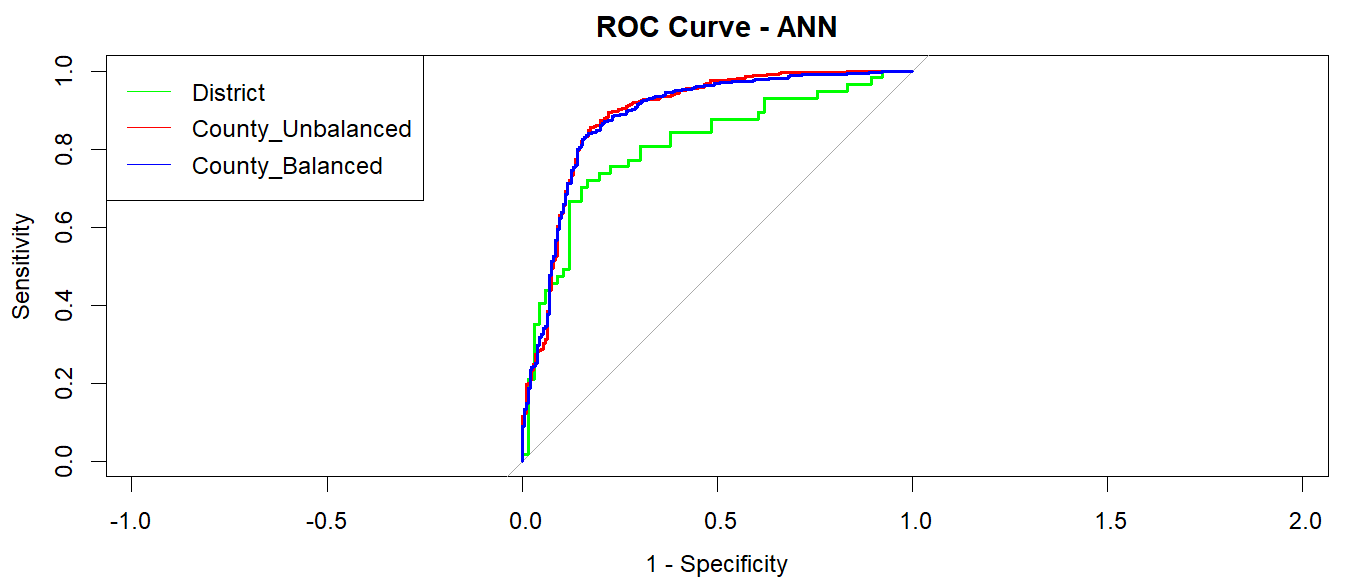
Below are the ROC curves for the four random forest models used followed by a full summary of model evaluation statistics:

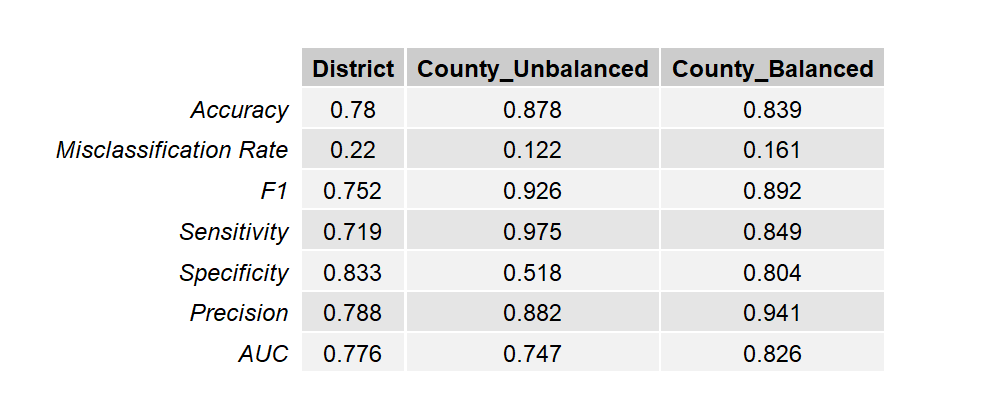




**Artificial Neural Network Model Plots and Data**

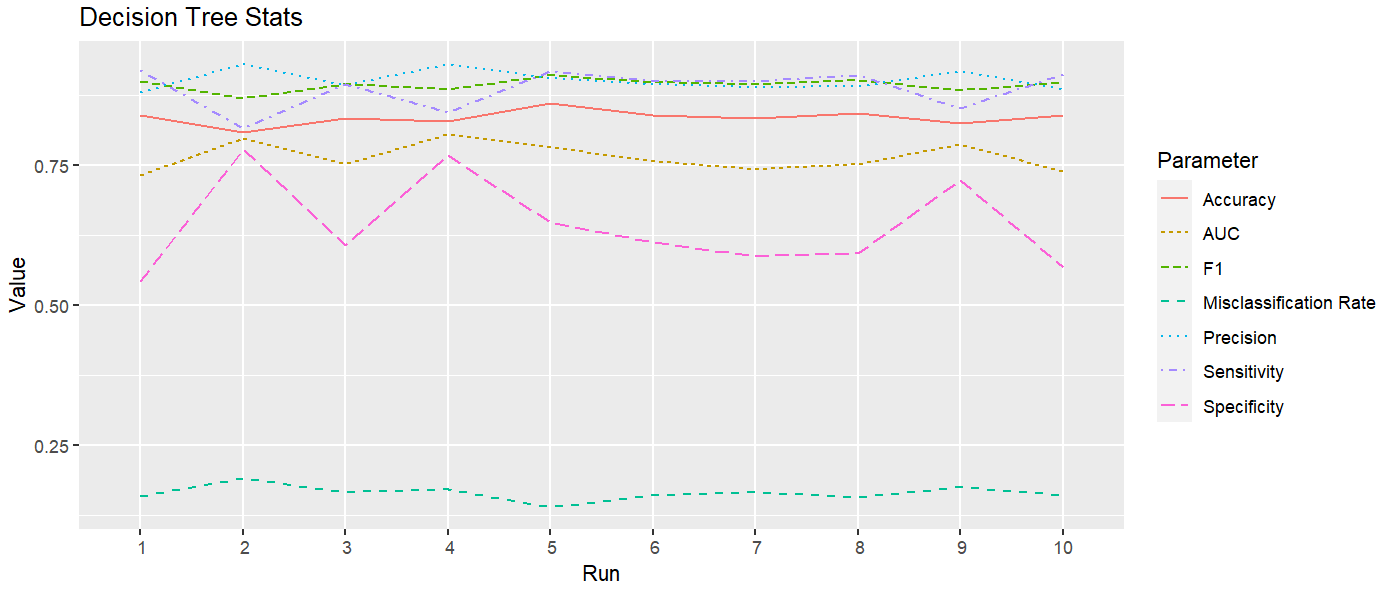
Below are the ROC curves for the three ANN models used followed by a full summary of model evaluation statistics:

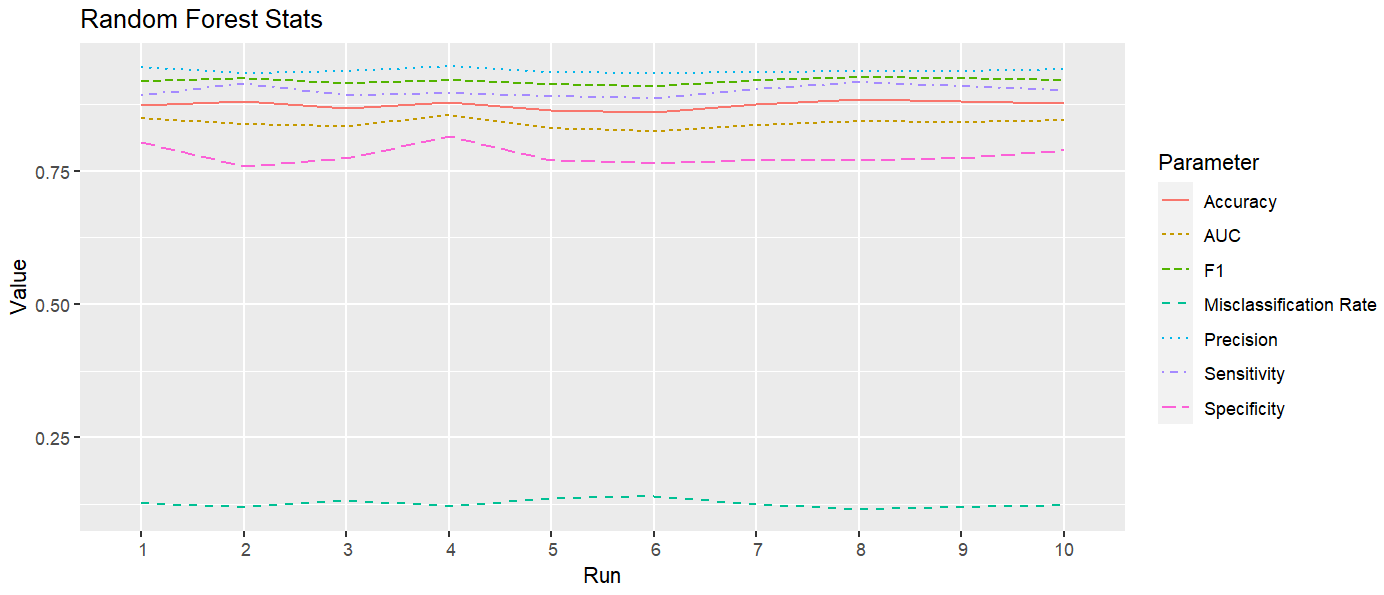




**Comparison of Decision Tree and Random Forest Model Stability.**

I had read that decision tree models can be very sensitive to changes in the training data, so I wanted to compare the stability of model evaluation parameters of the decision tree and random forest models. Below, I used county balanced data with the same ten random train/validation splits for decision trees and random forest models. I found it interesting, but expected, that the random forest model produced more consistent model results than the decision tree. Plots are below:

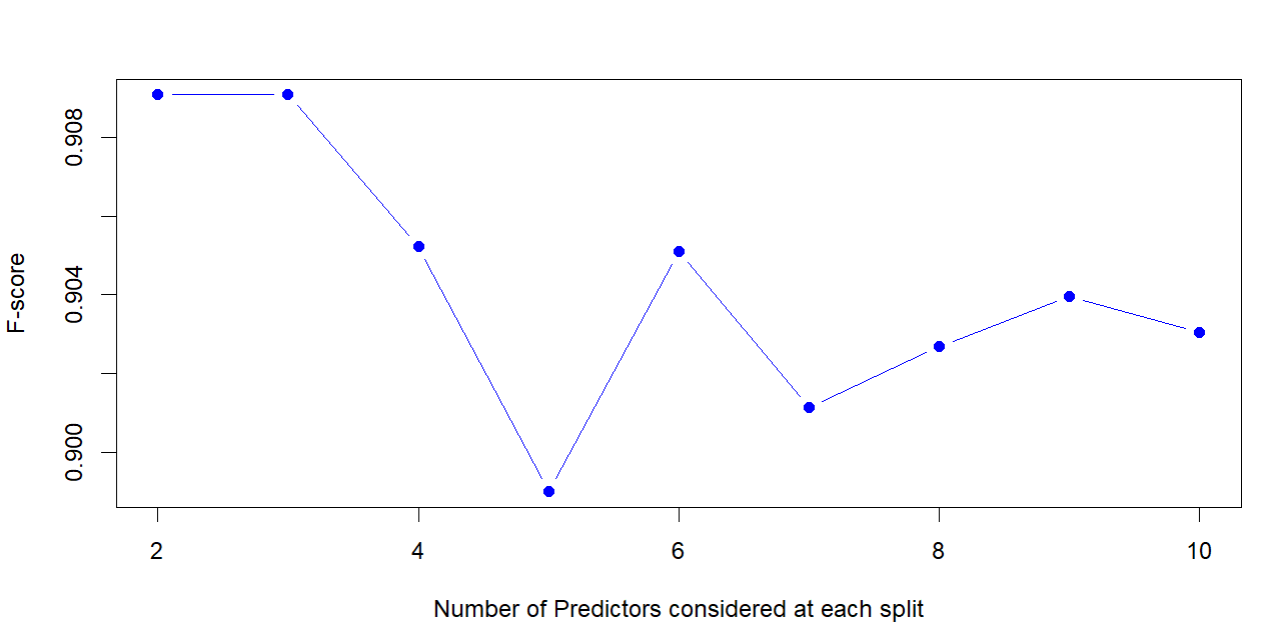




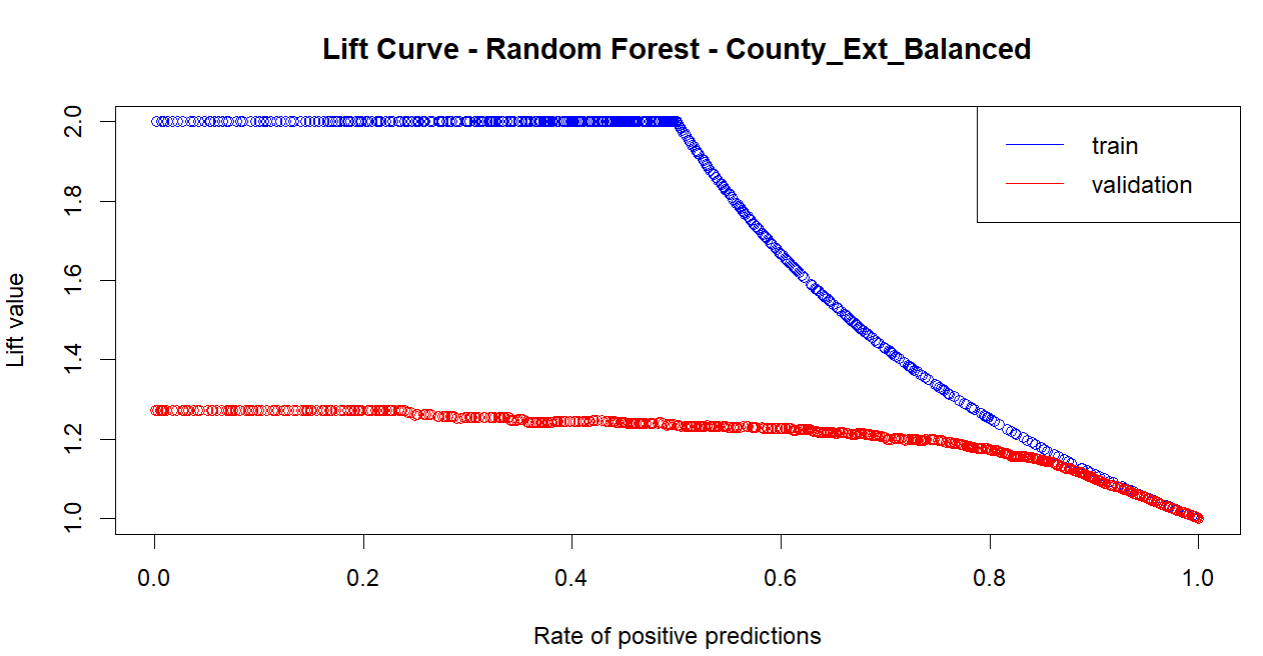
**Best Model – Details – Random Forest – County Externally Balanced**

Since I determined that the best model was the random forest - county balanced, I wanted to provided more information about the important model details.

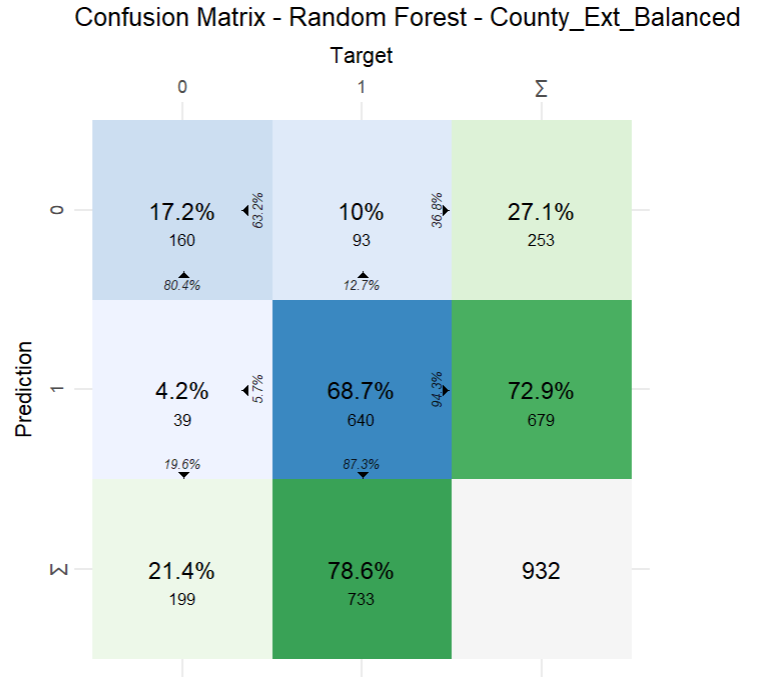
The plot used for selecting **mtry** is below:



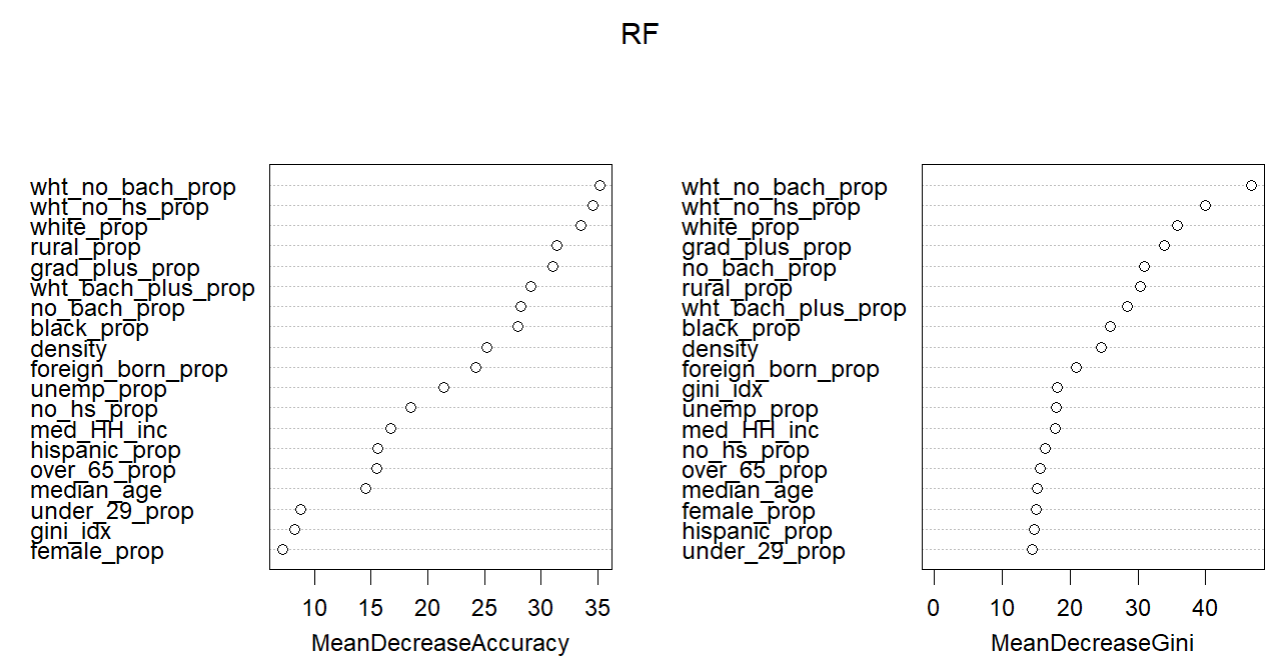
Lift curve is below:



Confusion Matrix for validation data:



Variable Importance:



**How to run:**

Most of the scripts can be run in isolation. I will briefly describe what each group does:

These files handle basic data cleaning and aggregation details for the election data:

**01-Prep\_Election\_Data.R**

**02-Aggregate\_Election\_Data.R**

These files handle linking census data to election data:

**03-Census\_Data\_County.R**

**04-Census\_Data\_District.R**

These scripts will not work unless you obtain an API key and enter it into both scripts. See link below:

<https://api.census.gov/data/key_signup.html>

These files provide some plots and information about the structure and distribution of the input variables:

**05a-District\_Analysis.R**

**05b-County\_Analysis.R**

These files handle all of the decision tree modeling and analysis:

**06a-Dtree-District.R**

**06b-Dtree-County\_Unbalanced.R**

**06c-Dtree-County\_Balanced.R**

**06d-Dtree-County\_Alt\_Thresh.R**

**06e-Dtree-Summary.R**

These files handle all of the logistic regression modeling and analysis:

**07a-Reg-District.R**

**07b-Reg-County\_Unbalanced.R**

**07c-Reg-County\_Balanced.R**

**07d-Reg-County\_Alt\_Thresh.R**

**07e-Reg-Summary.R**

These files handle all of the random forest modeling and analysis:

**08a-RF-District.R**

**08b-RF-County\_Unbalanced.R**

**08c-RF-County\_Ext\_Balanced.R**

**08d-RF-County\_Int\_Balanced.R**

**08e-RF-Summary.R**

These files handle all of the ANN modeling and analysis:

**09a-ANN-District.R**

**09b-ANN-County\_Unbalanced.R**

**09c-ANN-County\_Balanced.R**

**09d-ANN-Summary.R**

These files compare the stability of the decision tree and random forest models:

**10a-DTree-Analysis.R**

**10b-RF-Analysis.R**

This file produces a plot of winning margin vs. accuracy:

**11-Winner Analysis.R**

**Note: If you obtain a census API key, you could run every script from beginning to end. If you don’t want to obtain an API key, there are intermediate RData files included that should allow you to run from scripts**

**“05a-District\_Analysis.R” until the end in order.**