Predicting Red and Blue

Using Data Mining to Classify 2020 Presidential Election Outcomes

John Knight
Ilse Moya

The University of Texas Rio Grande Valley

STAT-4399 Data Mining

Prof. Hansapani Rodrigo

1 May, 2023

Introduction

In the 21st century, the United States of America has frequently been described as a nation divided along political lines, namely Republicans versus Democrats. Republicans might argue that geographically, America is largely a 'red' nation, as illustrated in Figure 1. In the 2020 presidential election, approximately 5 out of every 6 counties voted for Donald Trump (Riotta 2020). However, Democrats could counter that despite this, Joe Biden still managed to secure the majority of the popular vote, thanks to the concentration of the U.S. population in urban areas.

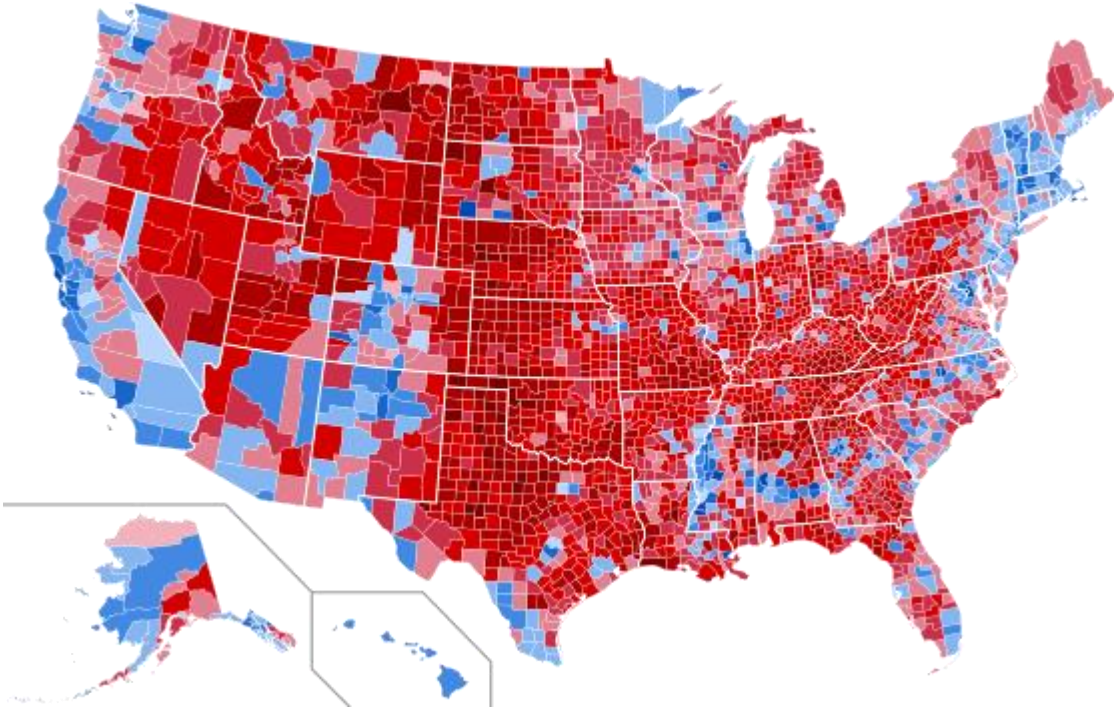

Figure 1. 2020 U.S. Presidential election map. Red counties were won by Donald Trump, blue counties by Joe Biden.

But what are the underlying causes of this division? How can one predict whether a given U.S. county is likely to be 'red' or 'blue'? Ahmed and Pesaran (2022) discovered that key determinants of voting outcomes at the county level include incumbency effects, unemployment, poverty, educational attainment, house price changes, and international competitiveness. Meanwhile, Talaifar et al. (2022) found that regions that voted for Donald Trump in 2016 and 2020 had high levels of neuroticism, economic deprivation, and low ethnic diversity.

Regarding the comparison of data mining techniques, Maroco et al. (2011) compared seven nonparametric classifiers and three traditional classifiers in a study predicting the development of dementia, concluding that random forest and linear discriminant analysis were the best performing models.

Our study aims to answer two research questions:

1. What factors are most important when classifying counties into 'red' and 'blue' in the 2020 United States presidential election?
2. Which data mining technique is the best-performing classifier for this task?

Methods

Data

Data were collected from a variety of sources (see appendix for list of sources). Table 1 shows the initial dataset with 16 variables.

Table 1. Description of 16 variables in the initial dataset.

Variable	Type	Definition
POPESTIMATE	Numeric	Estimated county population (in 2020).
MEDIAN_AGE_TOT	Numeric	Estimated median age.
Female_Pct	Numeric	Percentage of residents who are female.
White_Pct	Numeric	Percentage of residents who identify as "white only".
Black_Pct	Numeric	Percentage of residents who identify as "Black only".
Asian_Pct	Numeric	Percentage of residents who identify as "Asian only".
Hispanic_Pct	Numeric	Percentage of residents who identify as Hispanic.
Violent_Crimes	Numeric	Number of violent crimes in 2020.
Degree_Pct	Numeric	Percentage of residents with at least a bachelor's degree.
Rural_Urban_Continuum	Categorical (ordinal)	Scale from 1 (most urban) to 9 (most rural).
Unemployment	Numeric	Unemployment rate.
Median_Household_Income	Numeric	Median household income.
Poverty_Rate	Numeric	Percentage of households below poverty level.
GDP_2020	Numeric	2020 Gross Domestic Product.
GDP_Pct_Change	Numeric	Change in GDP from 2019 to 2020.
Trump_most	Response (binary)	Did Donald Trump receive more votes in the 2020 Presidential Election than any other politician? 1=Yes, 0=No.

It was decided that White_Pct would be omitted because it is strongly negatively correlated with Black_pct, Asian_pct, and Hispanic_pct, and any effect should be captured by those variables. Similarly, GDP_2020 was removed because this information is likely to be very similar to a combination of POPESTIMATE and Median_Household_Income. Finally, Violent_Crimes was converted to a rate statistic by dividing itself by POPESTIMATE. This is likely to better reflect the crime levels in a county than the total number of crimes, which will largely be a function of population size.

Missing data was handled in two ways: removal and imputation. Firstly, three rows were found in the data that had a large number (>6) of missing variables, and these three rows were removed from the dataset. It was also found that for two variables, there was a significant amount of missing data. Violent_Crimes had 390 missing data points (12.5% of cases), and GDP_Pct_Change had 69 missing data points (2.2%). These values were replaced with the means of Violent_Crimes and GDP_Pct_Change, respectively. This left a final dataset of 3112 observations with 13 predictor variables and one response variable.

Statistical Methods

Six types of classifiers were initially considered: Logistic Regression, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Naïve Bayes, Random Forest, and K-Nearest Neighbors (KNN). Each classifier has its pros and cons and can produce powerful models in the right circumstances. However, it was decided that KNN and Naïve Bayes were not appropriate for this task since they do not output the explicit feature importance, which is a crucial part of our study. Ultimately, the four methods tested were Random Forest, Logistic Regression, LDA, and QDA.

Firstly, the data were randomly divided into an 80% training set and a 20% test set. The same training set and test set were used for each classifier. Accuracy (equivalent to 1 – classification error rate) was used as the principal measure to evaluate the optimum version of each model. However, for comparison purposes, four measures were taken: accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC). Sensitivity, accuracy, and specificity are defined as follows:

$$Sensitivity = \frac{\text{Number of true positives}}{\text{Number of true positives and false negatives}}$$

$$Accuracy = \frac{\text{Number of true positives and true negatives}}{\text{All observations}}$$

$$Specificity = \frac{\text{Number of true negatives}}{\text{Number of true negatives and false positives}}$$

Sensitivity measures the proportion of positive cases correctly identified, while specificity measures the proportion of negative cases correctly identified. Accuracy measures the proportion of all observations correctly identified. AUC is designed as a more balanced measure of the model's overall classification ability, as it considers the trade-off between the true positive rate and false positive rate with a score close to 1 considered ideal.

Before running the random forest model, a simple decision tree was created using R's `tree` library. At each split in the tree, observations are partitioned according to some criterion $X < a$, with observations satisfying the criterion moving to the left branch and those not satisfying the criterion moving to the right branch. Criteria are chosen based on whichever split maximizes the reduction in impurity among remaining observations, as measured by the Gini index. In the default settings of `tree`, this process continues until the terminal nodes of the tree are too small or too few to be split. However, this can lead to overfitting, and so 5-fold cross validation was used to suggest an optimal tree size, after which the original tree was pruned to a smaller size. Accuracy measures were then compared on the test set for both the full depth tree and the pruned tree.

The weakness of a single decision tree is that it often has high variance. In other words, the tree will usually be sensitive to any changes in the input data, and different train:test splits can result in vastly

different trees. Bagging (bootstrap aggregation) is a method of reducing variance by taking a large number of bootstrap samples from the original training data, fitting a decision tree to each, and averaging the results. Random forests add an extra step by only considering a random subset of m variables at each split in the tree, to reduce the amount of correlation between trees. A general rule of thumb is if p is the total number of predictor variables in the dataset, $m = \sqrt{p}$ variables should be considered. Both bagging and random forests were performed using R's `randomForest` package, and all possible levels of m were compared on the test set. Note that when $m = p$, this is equivalent to bagging, although this can still be performed inside the `randomForest` function. Note that the test error can be estimated by the Out of Bag (OOB) error rate. This is done by testing each tree on the observations that were not included in each particular bootstrap sample (typically around one third of the data).

For the remaining models, it was decided that having 9 categories for Rural_Urban_Continuum might mean the data were too spread out. Therefore, this variable was recoded from the original 9 categories into 3. So {1,2,3} became 1, {4,5,6} became 2, and {7,8,9} became 3. Logistic regression was run using the `glm` function, initially with all 13 predictor variables. Then, using a stepwise process, insignificant variables were removed until all remaining variables showed significance at the 0.05 level. The full model and the reduced model were compared using the test data to see which had the lowest test error. Next, several chosen interaction terms were added to the superior model to see if this improved its performance on the test data. And finally, all possible pairs of interactions were added and the model run again, to see if this lowered the test error. Using the final best-performing model, standardized coefficients were calculated to rank the relative importance of each variable.

The final techniques used were Linear and Quadratic Discriminant Analysis. An important assumption of LDA and QDA is that predictor variables should be normally distributed. This was clearly not the case with many of the variables in this dataset, and so a function was constructed to transform each variable using the Box Cox method. Details of this function can be found in the appendix. Additionally, variables were centered and scaled using the `preProcess` function. Models were then fit using the `lda` and `qda` functions. The primary difference between LDA and QDA is that LDA assumes that different classes share a common covariance matrix, whereas QDA does not, meaning its decision boundary can be more flexible. It was not clear which of these models would be more appropriate, and therefore both models were fit to compare results on the test set.

Results

The unpruned decision tree had 16 terminal nodes with the first split being on Degree_Pct, and the second split on Black_Pct and Asian_Pct. This tree had a training error of 7.1% and a test error of 8.8%. After cross validation, the pruned tree (Figure 2) had 11 terminal nodes with a training error of 7.2% and a test error of 8.2%.

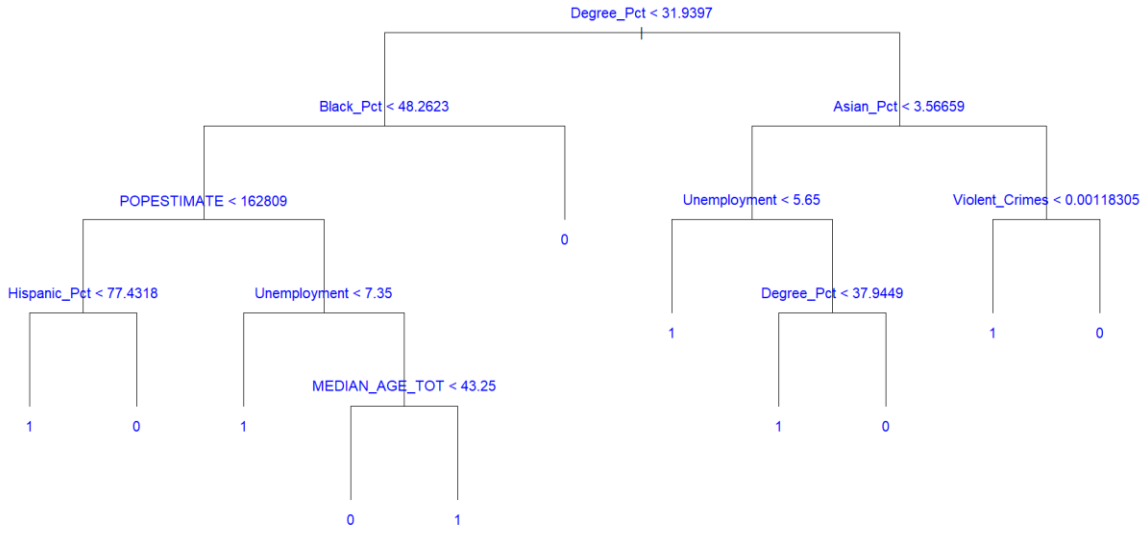

Figure 2. Pruned decision tree with 11 terminal nodes. 1=Trump, 0=Biden.

Using bagging (all 13 variables tried at each split), the OOB error estimate was 7.4%. After running random forests at all levels of m from 3 to 12, the lowest OOB error of 7.2% was found to be at m=7. However, since this could simply be overfitting based on variance, both m=7 and m=4 (from the default $\sqrt{p}$) were applied to the test set. It was found that the test error for m=4 was 6.6% compared to 7.4% for m=7. The final selected random forest model had parameter m=4 with accuracy of (100 – 6.6) = 93.4%, specificity 97.0%, sensitivity 77.4%, and AUC 0.872. Degree_Pct was the most important variable in terms of mean Gini decrease (Table 2), followed by Black_Pct and Asian_Pct.

Table 2. Variables ranked by their mean Gini decrease in the random forest model.

Variable	Mean Gini decrease
Degree_Pct	138.1
Black_Pct	108.0
Asian_Pct	102.9
POPESTIMATE	73.2
Unemployment	44.9
Hispanic_Pct	41.1
Poverty_Rate	36.4
Median_Household_Income	34.0
Female_Pct	33.7
MEDIAN_AGE_TOT	31.0
Violent_Crimes	21.8
GDP_Pct_Change	19.9
Rural_Urban_Continuum	14.2

The first logistic regression using all 13 variables had a residual deviance of 1051.9 and test error rate 7.9%. After using a stepwise process, the variables removed due to insignificance were:

MEDIAN_AGE_TOT, Female_Pct, Violent_Crimes, Rural_Urban_Continuum, and GDP_Pct_Change. The residual deviance with this final logistic regression model was 1054, with a test error rate of 8.2%, so the first model appeared to perform better.

Several interaction terms were chosen to be added to the first model. Of those tried, only three proved to be significant at the 0.05 level of significance. These were Black_Pct:Degree_Pct, Rural_Urban_Continuum:Black_Pct, and Hispanic_Pct:Poverty_Rate. The test error rate with these three interaction terms improved to 7.56%. Finally, all possible pairs of interactions were added to the model, and only those significant at the 0.01 level of significance were kept (a lower level was chosen because when a large number of interactions are tried, 5% of them would be significant at the 0.05 level purely by chance). However, the test error with these extra interaction terms increased to 7.88%, and so the model with the three interaction terms was chosen as the final logistic regression model with accuracy of 92.4%, specificity 96.8%, sensitivity 73.0%, and AUC 0.849.

Table 3 shows the standardized coefficients for the final logistic regression model. Note that 1=Donald Trump and 0=Joe Biden, therefore a negative coefficient implies the variable is negatively correlated with a county voting for Donald Trump.

Table 3. Variables ranked by their standardized coefficients in the logistic regression model.

Variable	Standardized Coefficient
Degree_Pct	-1.16
Black_Pct	-0.54
Unemployment	-0.50
POPESTIMATE	-0.30
Poverty_Rate	-0.29
Asian_Pct	-0.22
Rural_Urban_Continuum2	-0.20
Hispanic_Pct	-0.17
Rural_Urban_Continuum3	-0.17
Hispanic_Pct:Poverty_Rate	-0.12
Female_Pct	-0.08
GDP_Pct_Change	-0.03
Violent_Crimes	0.02
MEDIAN_AGE_TOT	0.05
Black_Pct:Rural_Urban_Continuum3	0.12
Median_Household_Income	0.16
Black_Pct:Rural_Urban_Continuum2	0.25
Black_Pct:Degree_Pct	0.31

After performing the Box Cox transformations, the LDA model had a test error of 12.1%, equivalent to accuracy of 87.9%. Its specificity was 95.1%, sensitivity 56.5%, and AUC 0.812. Here are the coefficients of the linear discriminants:

Table 4. Variables ranked by their coefficients in the linear discriminant model.

Variable	Coefficient of Linear Discriminant
Degree_Pct	-1.22
Unemployment	-0.55

Poverty_Rate	-0.44
Asian_Pct	-0.16
Rural_Urban_Continuum3	-0.14
Black_Pct	-0.10
Hispanic_Pct	-0.07
Female_Pct	-0.07
GDP_Pct_Change	-0.03
Violent_Crimes	0.02
Rural_Urban_Continuum2	0.07
Median_Household_Income	0.09
POPESTIMATE	0.11
MEDIAN_AGE_TOT	0.19

The QDA model had a test error of 14.1%, equivalent to accuracy of 85.9%. Its specificity was 90.9%, sensitivity 63.5%, and AUC 0.812. Note that QDA does not explicitly output the feature importance.

Finally, Table 5 compares the performance of the four different models used in this study.

Table 5. Accuracy, specificity, sensitivity, and AUC score of the four different models.

Model	Accuracy	Specificity	Sensitivity	AUC
Random forest	93.4%	97.0%	77.4%	0.872
Logistic regression	92.4%	96.8%	73.0%	0.849
LDA	87.9%	95.1%	56.5%	0.812
QDA	85.9%	90.9%	63.5%	0.812

Discussion

Of the four methods tried, the random forest proved to be the best-performing model on the test data according to all measures. In this dataset, many of the variables are likely to be correlated (for example, Median_Household_Income and Poverty_Rate) and random forests are generally known to be good at handling nonlinear relationships and interactions. It is also noticeable that all four models had much higher specificity than sensitivity. This is due to the imbalanced nature of the dataset, where 83.1% of the training set had a response variable of 1. Fitting the models based on the measure of accuracy has an inherent bias towards the more frequent classification. A possible further avenue of investigation would be to oversample the minority class using a method such as bootstrap or SMOTE (synthetic minority over-sampling technique) to create a balanced dataset and see if that changes the results.

Of the variables used in this study, Degree_Pct was the most important variable in every model. A bivariate boxplot of Degree_Pct shows that counties where Joe Biden received the majority of the vote had, on average, a higher percentage of residents with bachelor's degrees than counties that voted for Donald Trump.

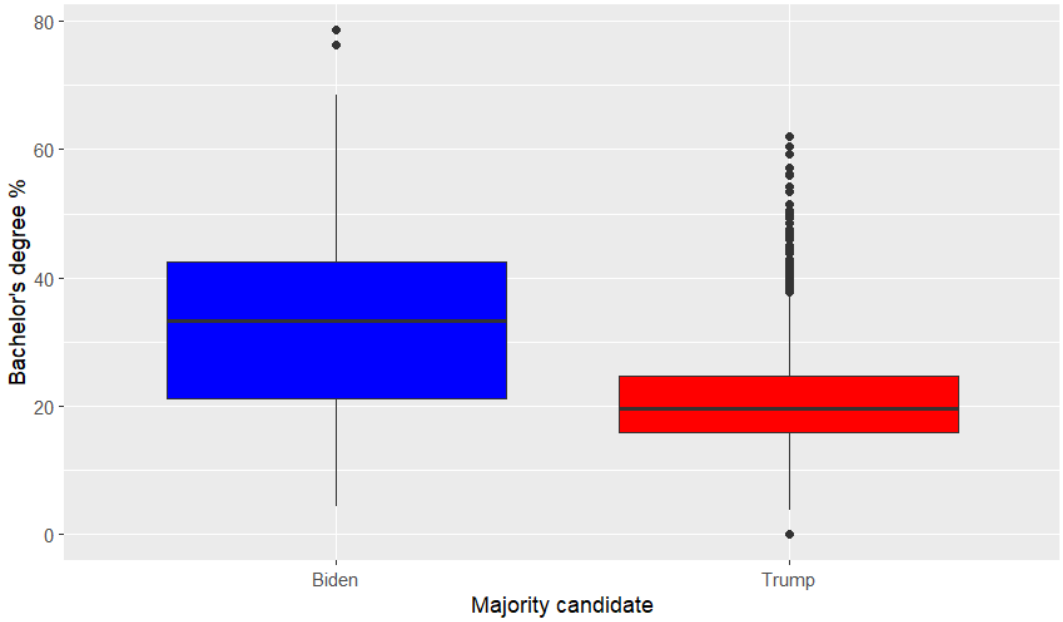

Figure 3. Boxplot comparing the percentage of residents with a college degree in states that voted for Biden and Trump.

To test the significance of the difference in Degree_Pct between counties that voted for Biden and Trump, a two sample t-test was performed. Figure 4 confirms that the difference is significant, with a 95% confidence interval showing that the difference of the means is within the range (10.64, 13.04).

```
        Welch Two Sample t-test

data:  group1 and group2
t = -19.368, df = 601.59, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -13.04024 -10.63917
sample estimates:
mean of x mean of y
 20.96411  32.80382
```

Figure 4. Two sample t-test comparing degree percentage in Biden and Trump counties.

It is also apparent from the random forest model that the racial makeup of a county can be predictive of voting tendences, with Black_Pct, Asian_Pct, and Hispanic_Pct all featuring highly. However, Rural_Urban_Continuum was the least important variable in the random forest model – perhaps surprising given the apparent clustering of blue counties around major urban areas in Figure 1.

Further development of this study might include other variables such as religion, number of COVID deaths, or gun ownership. Additionally, it would be interesting to perform the same study over multiple presidential elections to see how the contributing factors have changed over time.

Appendix

References

Ahmed, R., & Pesaran, M. H. (2022). Regional heterogeneity and US presidential elections: Real-time 2020 forecasts and evaluation. *International Journal of Forecasting*, 38(2), 662-687.

Maroco, J., Silva, D., Rodrigues, A., Guerreiro, M., Santana, I., & de Mendonça, A. (2011). Data mining methods in the prediction of Dementia: A real-data comparison of the accuracy, sensitivity and specificity of linear discriminant analysis, logistic regression, neural networks, support vector machines, classification trees and random forests. *BMC research notes*, *4*(1), 1-14.

Riotta, C. (2020, December 15). How many counties did Biden win? Less than any other president — but he still received more votes than Trump. *The Independent,* Independent Print Limited.

Talaifar, S., Stuetzer, M., Rentfrow, P. J., Potter, J., & Gosling, S. D. (2022). Fear and deprivation in Trump's America: A regional analysis of voting behavior in the 2016 and 2020 US presidential elections. *Personality Science, 3,* 1-57.

Data Sources

Presidential Election results by county, 2000-2020:

https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/VOQCHQ

Age/sex breakdown by county, race/sex breakdown by county, 2020:

https://www.census.gov/data/tables/time-series/demo/popest/2020s-counties-detail.html

Violent Crime Rate by County 2020 (have to C&P from each state)

https://www.countyhealthrankings.org/explore-health-rankings/county-health-rankings-model/health-factors/social-economic-factors/community-safety/violent-crime?year=2020&state=48&tab=1

U.S. County-level Data Sets:

(Education, PopulationEstimates, Unemployment, PovertyEstimates)

https://www.ers.usda.gov/data-products/county-level-data-sets/

GDP by county:

https://www.bea.gov/data/gdp/gdp-county-metro-and-other-areas

County FIPS codes:

https://transition.fcc.gov/oet/info/maps/census/fips/fips.txt

R Functions

Function to get standardized coefficients from the logistic regression model:

```
std.coef <- function (mymodel) {
  model_data <- mymodel$model # Include the response variable in the data
```

```
  model_formula <- as.formula(mymodel$call$formula) # Extract the model formula

  # Standardize only numeric variables (excluding the response variable)
  for (i in colnames(model_data)[-1]) { # Exclude the response variable from the loop
    if (is.numeric(model_data[[i]])) {
      model_data[[i]] <- (model_data[[i]] - mean(model_data[[i]])) /
sd(model_data[[i]])
    }
  }

  # Create a new model with standardized numeric variables
  std_model <- glm(model_formula, family = "binomial", data = model_data)

  # Compute standardized coefficients
  b <- summary(std_model)$coef[-1,1] # Exclude the intercept
  beta <- (3^(1/2))/pi * b
  return(beta)
}

#Get the coefficients
sc = data.frame(Standardized.Coeff = std.coef(mylog))
sc = cbind(Variable = row.names(sc), sc)
row.names(sc) = NULL
sc[order(sc$Standardized.Coeff),]
```

Function to perform Box Cox transformation:

```
trans <- function(x) {
  bc <- boxcox(x ~ 1, plotit = FALSE)
  lambda <- bc$x[which.max(bc$y)]
  x <- if (lambda == 0) {
    log(x)
  } else {
    (x^lambda - 1) / lambda
  }
  return(x)
}
```