Predicting Income Level From Census Data - Supervised Machine Learning Project

Tanar Hernandez-Wroblewski (3920188)

Introduction

Governments around the world are interested in developing an adequate solution to address the prominent disparity of wealth within the world's population. Tackling this issue consists of the creation of policy and programs to stifle the rise in widespread poverty. One of the first pieces to this puzzle is to determine the root causes of this wealth gap and discover a target on which to focus our efforts. In the 2010 census of the United States, the real median household income was found to be $49,445. By some measure, households who have an income that exceeds $50,000 can be considered as having an above average level of income. In this project, various methods of supervised machine learning will be applied to a census dataset to develop predictive models that classify an individual as making either above or below $50,000. Through the exploration and analysis of this dataset, we aim to identify the most effiecient predictors and thus determine what features are distinct to the group of indivduals whose income level exceeds this "average level". We hope that with this information we could potentially be able to apply these methods to future data in order to learn where to most accurately allocate the policy and programs that combat poverty.

Data

The dataset used for this analysis is the Census Adult Income Data Set from the UCI Machine Learning Repository. This dataset was extracted by Barry Becker from the 1994 Census database and given to the UCI Machine Learning Repository. Despite being somewhat outdated, it is still presumable that the learning and predictive methods performed and resulting conclusions that are arrived at using this dataset could be applied to more recent census data to perform similar analysis.

This dataset contains 32,560 observations with 15 attributes including:

Age: the age of the individual (integer greater than 0)

Workclass: the category of work the individual performs (factor w/9 levels: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.)

Final Weight: the amount of people the census creators believe the individual entry represents (integer)

Education: the highest level of education obtained by the individual (factor w/ 16 levels: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool)

Education Years: the number of years of school attended by the individual (integer)

Marital Status:the marital status of the individual (factor w/ 7 levels: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse)

Occupation: the occupation of the individual (factor w/ 15 levels: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces)

Relationship: the relationship the individual holds to other (factor w/ 6 levels: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.)

Race: the individual's racial identification (factor w/ 5 levels: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black)

Sex: the biological sex of the individual (factor w/ 2 levels: Male, Female)

Capital Gain: capital gains for the individual (integer)

Capital Loss: capital losses for the individual (integer)

Hours Per Week: the hours per week worked by the individual (integer)

Native Country: the individual's native country (factor w/ 42 levels)

50K: the response variable: binary output for whether or not an indiviual's income is above or below 50 thousand (factor w/ 2 levels: <=50K (0), >50K (1))

Preprocessing the data

After reading in the dataset and assigning more easily callable variable names, we evaluate the overview of the dataset and structure of the predictors. The response variable was turned into a binary response named "fifty" to aid in the performance of calssification. The outcome is a 0 if the individual has an income below $\$50,000$ and a 1 if the individual has an income above $\$50,000$. As no missing entries or glaringly problematic issues were found up to this point, we can move forward with the exploratory analysis.

Exploratory Analysis

First we will examine the breakdown of observations that lie within each of the binary outcomes, a household income level of either above 50K or below 50K.

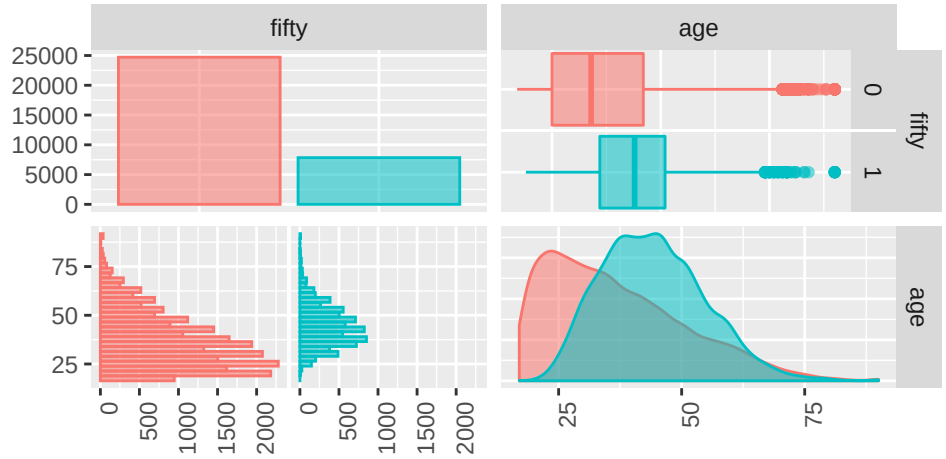
Table 1: Frequency of observations with incomes either above (1) or below (0) the $50k threshold

Outcome	Frequency
0	24719
1	7841

From the table, we observe that 75.92% of the observations in the full dataset are classified as having an income below $\$50,000$ while 24.08% lie above this income threshold. We will try to preserve this breakdown in the creation of the training and test sets.

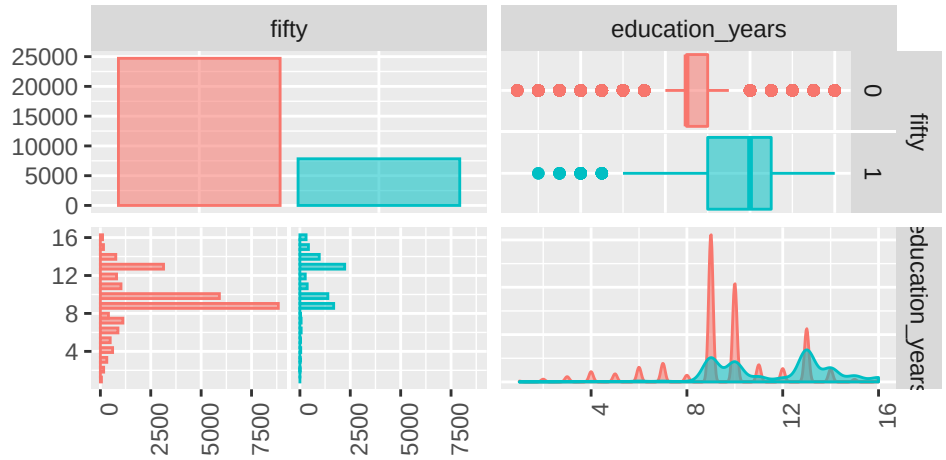
We will now investigate the dataset features through exploratory methods of selected predictors in an attempt to gain insight on how to possibly reduce model complexity through the identification of redundant, unecessary, or problematic predictors. In addition, we will test for possible colinearity between the continuous predictors. The predictors of numeric data type will be examined first through the creation of boxplots, density plots, and histograms of the predictors with respect to the response variable "fifty".

Age



The boxplot and density plot indicate that individuals whose income falls above the 50K threshold tend to be slightly older. This is a reasonably believable outcome since as people get older, they gain more experience and therefore command a higher wage. Given this information, it is safe to presume that age will be a fairly strong predictor of income level and should be included in the predictive models.

Education Years

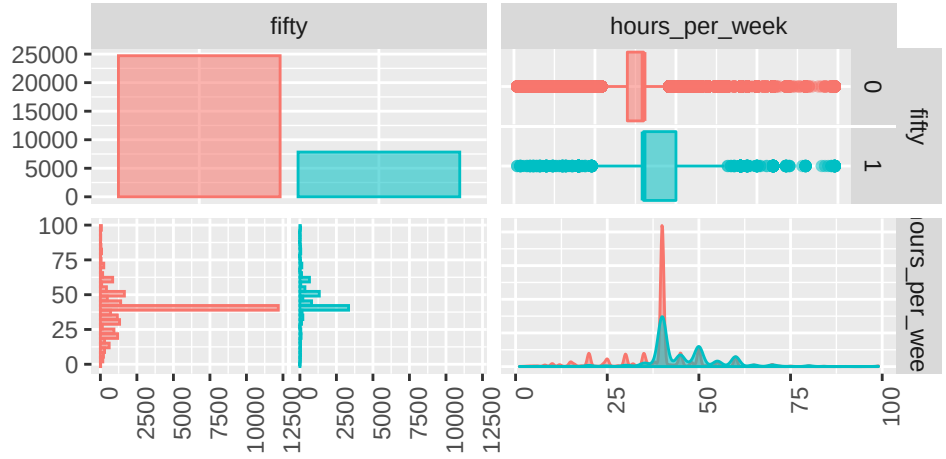


The plots have implications similar to the age variable. Individuals with higher income tend to have more years of education. As shown in the boxplot, the mean value appears to be about 4 to 5 years greater in the blue plot correspinding to an income level above $\$50K$. The histogram shows that an extremely small number of the above 50K individuals reported having lower than 8 years of education. Again, this predictor variable seems highly relevant to the model. The "education" predictor, which has a factor data type, reiterates the same information contained within this predictor so we will exclude it's exploratory analysis for simplicity.

Capital Gain/Loss

Both the Capital Gain and Capital Loss variables can take on an infinite range of values, therefore exploratory graphics are difficult to interpret due to their skewedness. However, their effect on the models will still be investigated using p-values and cross validation during the model creation process.

Hours Per Week



As indicated in the histogram and density plot, most of the observations are close to standard 40 hours per week. However, the observations that fall into the high income level are skewed slightly to a higher number of hours per week. This means that people who log more hours of work per week tend to be classified as being in the above 50K income level at a higher rate. All of these observations point towards hours per week being a strong predictor of income level so it will be included in the models.

Final Weight

The distribution of final weight is practically identical between each of the two binary responses. The only difference being due to there being a greater proportion of the total observations that fall into the lower income level classification category. Since this is the only discernable discrepency in the distributions, it seems likely that final weight will not be a significantly strong predictor of income level.

We will now move forward with the exploratory analysis of a selection of interesting predictors with categorical responses.

Marital Status

Table 2: Proportion of observations in each marital status category and income level

	0	1
Divorced	0.1610097	0.0590486
Married-AF-spouse	0.0005259	0.0012753
Married-civ-spouse	0.3351268	0.8534626
Married-spouse-absent	0.0155346	0.0043362
Never-married	0.4122740	0.0626196
Separated	0.0387961	0.0084173
Widowed	0.0367329	0.0108405

The table shows that a large majority, 85.35%, of the individuals in the higher income level fall into the distinction, "Married Civilian Spouse". Just under half of the individuals within the lower income level responded "Never married" compared to only 6.26% of higher income individuals sharing this distinction. These large discrepencies coupled with the heavy propotions of certain marital statuses within each classification point towards this being a highly significant predictor of income level.

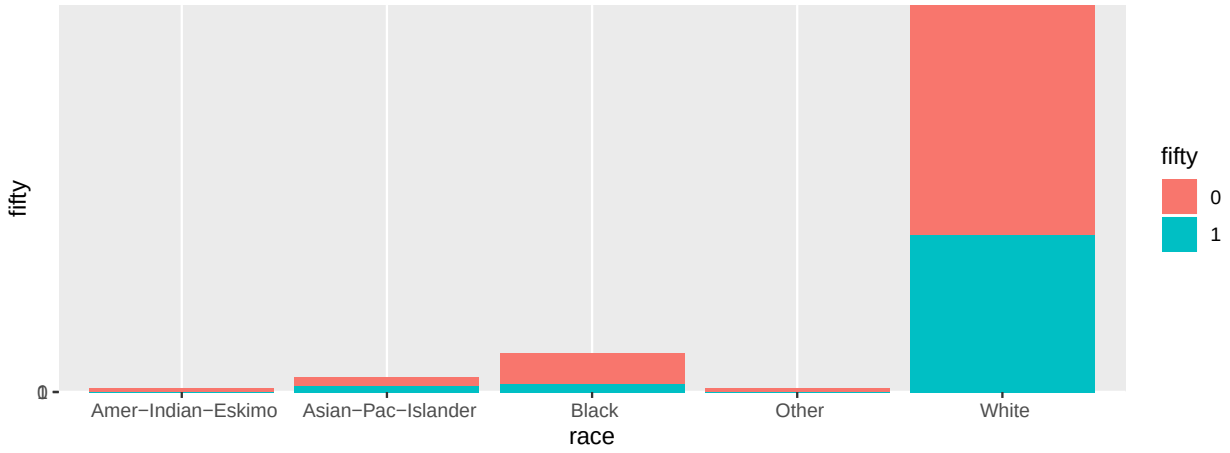
Occupation

Table 3: Proportion of observations in each occupational sector and income level

	0	1
?	0.0668312	0.0243591
Adm-clerical	0.1319633	0.0646601
Armed-Forces	0.0003236	0.0001275
Craft-repair	0.1282414	0.1184798
Exec-managerial	0.0848740	0.2509884
Farming-fishing	0.0355597	0.0146665
Handlers-cleaners	0.0519438	0.0109680
Machine-op-inspct	0.0708767	0.0318837
Other-service	0.1277560	0.0174723
Priv-house-serv	0.0059873	0.0001275
Prof-specialty	0.0922772	0.2370871
Protective-serv	0.0177192	0.0269098
Sales	0.1078927	0.1253667
Tech-support	0.0260933	0.0360923
Transport-moving	0.0516607	0.0408111

The two most common occupations for individuals above the income level threshold are shown to be "Executive-Managerial" 25.1% and "Professional-Specialty" 23.7%. The proportions of individuals holding the same occupations in the lower income level are less than half as large. Although not to the same extent, similar disparity arises when comparing the some of the more commonly held occupations in the below 50K income level to the proportions of the above 50K level. It makes intuitive sense that one's occupation has a palpable impact on their income and the table backs up this claim.

Race



From the figure, we can see that white is by far the most commonly observed race. The stacked barplot still seems to imply that races other than white belong to the below 50K income level category at higher rates. Although the information contained in this predictor seems like it will be useful in classification, the sheer volume of observations that are white may convolute some of the predictive power of this variable.

Sex

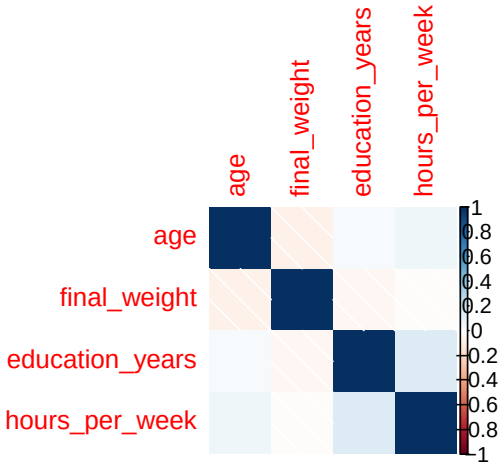
Table 4: Proportion of observations of each sex and income level

	0	1
Female	0.3880416	0.1503635
Male	0.6119584	0.8496365

Again it seems that there were more far male respondents in this particular census dataset. Nonetheless, the table suggests that the above 50K income level is comprised of a greater proportion of males than the below 50K level is. It remains to be seen how great the effect of having more total male than female observations will have on the models.

Correlation Detection

We will now look at the relationships between the continuous predictors in order to indentify and mitigate any potential issues of collinearity. This will be done using a simple correlation plot.



The plot indicates the continuous predictors all share small correlations with eachother, ruling out any colinearity and confirming that it is safe to proceed with the model creation.

Methods

Training and Testing Sets

Before building the predictive models, the dataset will be split into a training set and a testing set. The training set used to fit and train the models will be a random selection of 70% (22,792 obs.) of the observations from the full dataset. The remaining 30% (9,768 obs.) of the data will be withheld during the model creation processes and used in the test set. The test set's function is to act as unseen data on which to test model accuracy.

Logistic Regression

The first classifier that will be developed and implemented is logistic regression, a variation of a generalized linear model that uses the log-odds, or logit, link function defined below.

$$logit(p) = ln(\frac{1-p}{p}) = x^T\beta \iff P(Y=1|x) = \frac{e^{x^T\beta}}{1-e^{x^T\beta}} \tag{1}$$

where,

$$x^T\beta = \beta_0 + x_1\beta_1 + x_2\beta_2 + ... + x_n\beta_n \tag{2}$$

A baseline logistic regression model will be fit including all available predictors. This particular model's performance will be scrutinized through the calculation of a confusion matrix depicting the true positive, true negative, false positive and false negative rates. The elements of the confusion matrix will then be used to perform predictions based on a probability threshold of 0.5, also known as a majority rule. The resulting accuracy and misclassification error rate will be calculated from these prediction values.

In attempt to acheive a more accurate prediction, a Reciever Operating Characteristic Curve (ROC) will be plotted to identify the appropriate tradeoff bewteen the true positive rate and false positive rate by selecting an optimal probability threshold. The area under the curve of the ROC will also be calculated and used as

another metric of overall model performance. Lastly, new predictions will be done using the found opitmal probability threshold and used to calculate the new accuracy and misclassification error rate to compare to the metrics found with the majority rule. Finally, we will call a summary of the best performing model and note the subset of predictors with the smallest p-values. We define these as the most significant predictors in the model.

A second logistic regression model will be fit using only the subset of the most significant predictors to try and reduce model complexity and bias to improve the model performance when applied to unseen data in the test set. The analysis of the model fitted to the most significant predictors will be identical to that performed on the baseline model to assure accurate comparisons.

Once each model has been created and their respective probability thresholds have been found, we will identify the model that produces the lowest misclassification error rate on the training data. This best performing model will then be used to make new classification predictions using the unseen testing set. The accuracy and misclassification error rate will be noted for comparison to the other methods.

Decision Trees

The second classification method to be used is the decision tree or more specifically the classification tree. Decision trees use a greedy approach to recursive binary splitting. In other words it selects the regions $R_{j_n} = (X|X_i < s)$ and $R_{j_{n+1}} = (X|X_i \geq s)$ of the feature space, variables, and cutpoints that give the biggest decrease in region impurtiy.

Similarly to the logistic regression model, we will begin by fitting the classification tree using a large quantity of predictors. This particular model will exclude the variable "native_country" since it contains more than 32 factor levels, breaking the tree function's cardinality constraint. Once fit to the training data, the model's accuracy and misclassification error rate will be computed with respect to predictions on the test set. These metrics are considered our baseline for this classification method.

Two methods to prevent overfitting are then used to reduce the baseline model's complexity. First, Pruning is implemented to determine an optimal split count under the pruning criteria. Second, K-Fold Cross Validation is performed giving it's own optimal split count. Two new trees will be created from these suggested split counts and their accuracy and misclassification rates are then calculated on the test set and compared the the baseline model performance.

Bootstrap Aggregation and Random Forest

The final models to be fit will be of the ensemble learning variety. In other words, instead of only using a single classification model, an ensemble method uses the combination of many classification models in an attempt to improve overall performance. First, Bagging (Bootsrap Aggregation) will be implemented. The intuition behind bagging is that averaging multiple independent estimates should reduce the variance of the overall estimate. In a classification case, we will use the majority rule defined as $\hat{f}_{comb}(x) = mode(\hat{f}^1(x), \hat{f}^2(x), ..., \hat{f}^B(x))$. However, since we only have the training set data available, we will use psuedo replicate trees using the bootsrap. Large trees will be built, without pruning, that will overfit the data. These are averaged to create a final prediction from the B overfitted trees. The error of the bagged tree will be examined using a plot of the classification errors and OOB (Out Of Bag) error.

A Random Forest model is then developed. The Random Forest method is essentially bagging applied to a decision tree with the added caveat that it will try to decorrelate each of the bootstrap trees. Each time a new split is chosen, it will split one of the randomly sampled predictors chosen on $m = \sqrt{p}$ randomly chosen variables, as opposed to the $m = p$ chosen in the bagging model. The 10 most important predictors will be plotted in a variable importance plots based on the mean increase in accuracy and gini.

Both the Bagging and Random Forest trees are then evaluated on the testing set, having their accuracies and misclassification error rates noted and compared to the previously developed calssification models.

Discussion (Model Building)

Logistic Regression

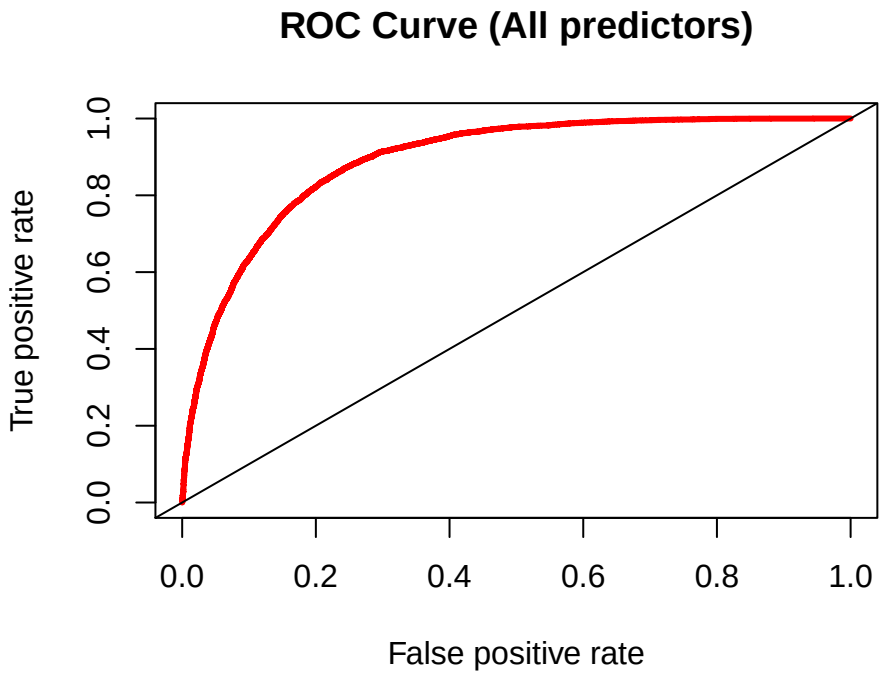
We begin by creating a baseline logistic regression model using all available predictors with the training set. Calling a summary of the fit gives a z-value for each predictor to measure it's significance within the fit. The set of predictors with the highest levels of significance are noted as they will be used to fit the following logistic regression model. Excluded due to the large amount of output, the fitted model is used to perform predictions with respect to a threshold of 0.5. With these predictions, we calculate the misclassification error rate to the training data to be 16.23377% corrresponding to an accuracy of 83.76623%.

A confusion matrix is then created using the training data predictions in order to calculate the true postive and false postive rate.

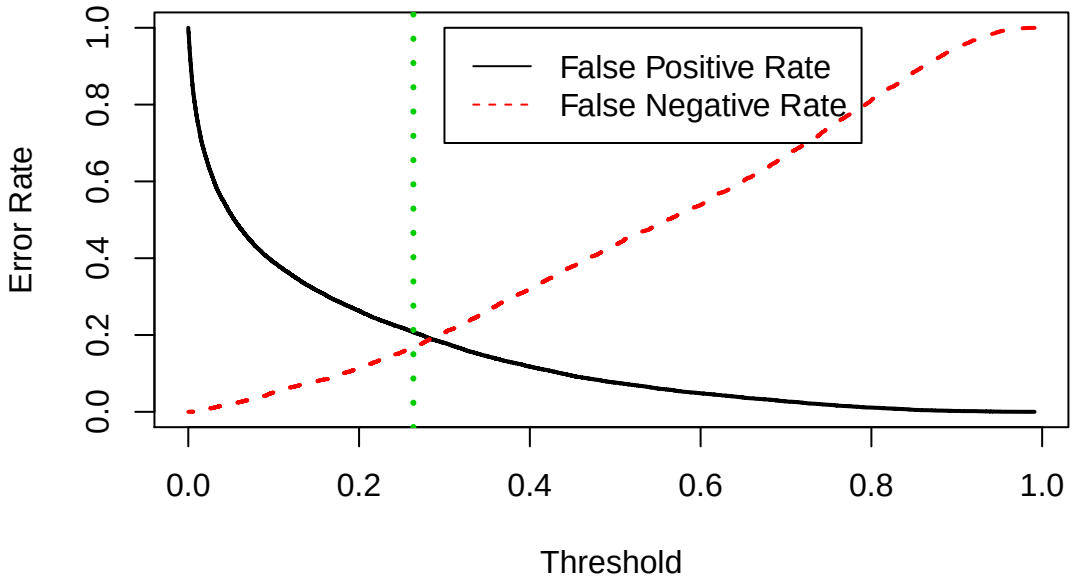
Table 5: Confusion Matrix for Logistic Regression Model 1

	0	1
0	15987	1316
1	2384	3105

From the confusion matrix, $TPR = \frac{3105}{2384+3105} = 56.5676\%$ and $FPR = \frac{1316}{1316+15987} = 7.6056\%$. Using these values we can plot the ROC Curve to identify the appropriate tradeoff between the true and false positive rates as well as calculate and overall score of the model as the area under the curve metric.



The AUC is found to be 0.89147 out of a maximum score of 1.0. This model seems to be fairly accurate. To find the optimal threshold value, the tpr and fpr will be plotted onto a graph of the threshold against the error rate.



The graph shows that the optimal threshold value is at the probability 0.2636124. Using this new optimal threshold value, the same analysis is performed as with the 0.5 threshold.

Table 6: Confusion Matrix for Logistic Regression Model 1 (Optimal Threshold)

	0	1
0	13722	3581
1	917	4572

From the new confusion matrix, we calculate that $TPR = 83.3121\%$ and $FPR = 20.6958\%$. With an accuracy 80.2694% and misclassification error 19.7306%.

As the construction of the second logistic regression model follows identical steps to the previous model fitted to all of the predictors, this section will be less verbose. However, plots will still be shown and results compared to the performance of the prior model.

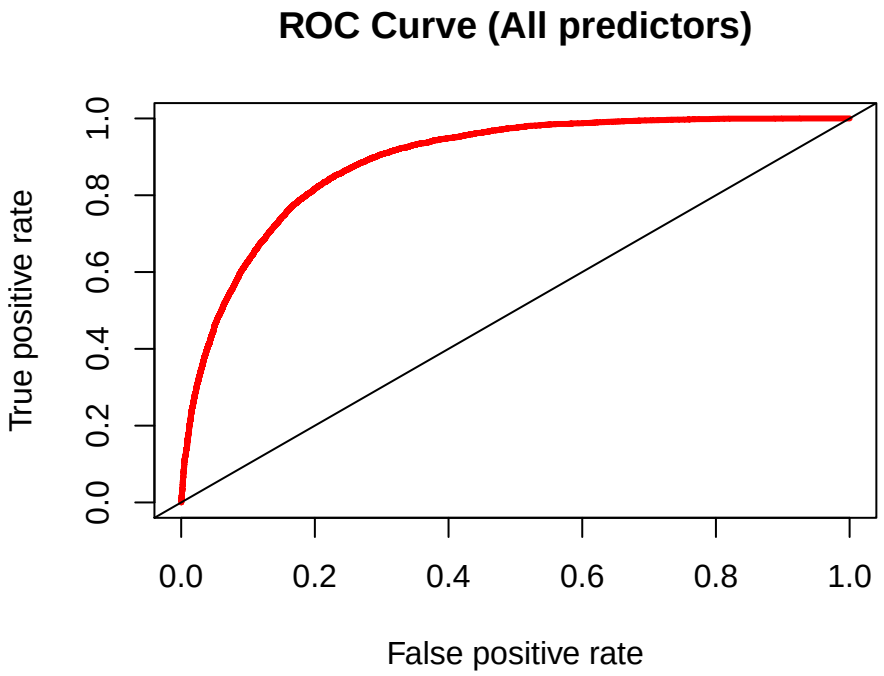
Mentioned earlier, this model will be fitted to the training data using only the most significant predictors in the last model. These 8 predictors are: age, workclass, final_weight, education, marital_status, occupation, relationship, and hours_per_week. The confusion matrix for this model fit to the training data is given as,

Table 7: Confusion Matrix for Logistic Regression Model 2

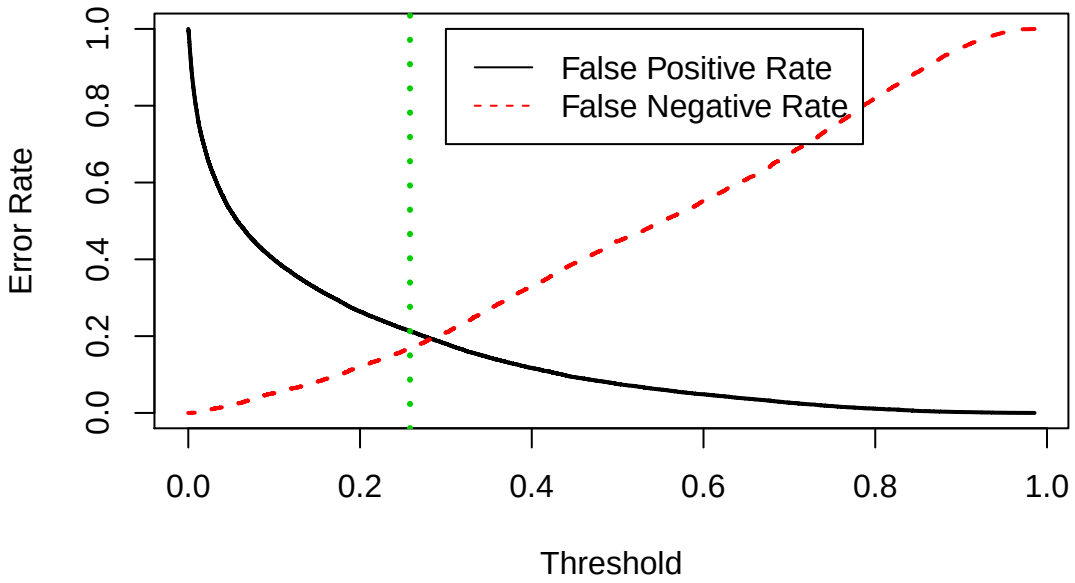
	0	1
0	16001	1302
1	2454	3035

Which gives $TPR = 55.2924\%$, $FPR = 7.5247\%$, $Accuracy = 83.5205\%$, and a misclassification error rate

16.4795%.



The AUC in this model comes out to be 0.8884319, slightly underperforming the baseline model fitted to all predictors. We will move forward with plotting and finding the optimal probability threshold.



The plot shows the optimal probability threshold to be 0.2581948. The confusion matrix calculated using this optimal cutoff value is,

Table 8: Confusion Matrix for Logistic Regression Model 2 (Optimal Cutoff)

	0	1
0	13627	3676
1	919	4570

The values in this confusion matrix are used to calculate the accuracy to be 79.8394% and the misclassification error rate to be 20.1606%.

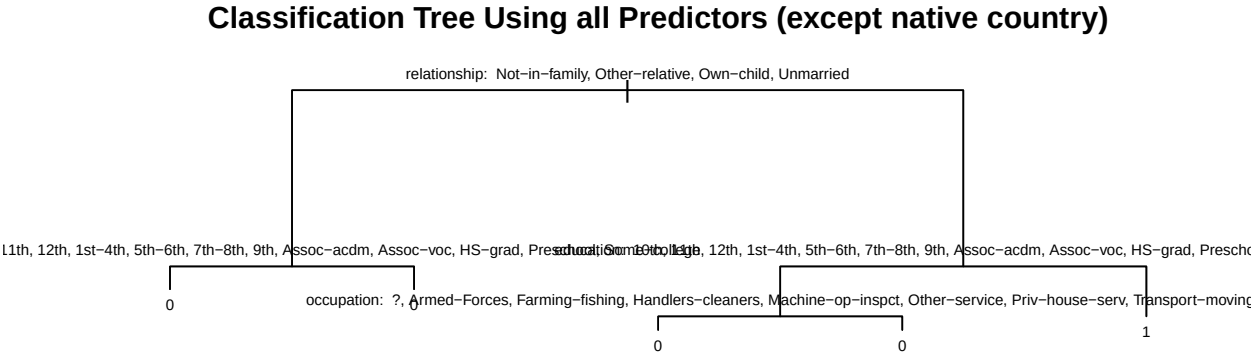
We now select the best performing variant from each of the two logistic regression models fitted to the training data and evaluate their performance on the the testing set. The first model gives an accuracy of 79.3407% and misclassification error rate 20.6593%. The second model gives an accuracy of 78.81859% and misclassification error rate 21.18141%

Although intuitively the model fitted only to the most significant predictors seems like it should have performed better, the model fitted to all available predictors had an accuracy around 1 superior. Therefore, when comparing the performance of all models developed in this project. The Logistic Regression model fitted to all available predictors will be used.

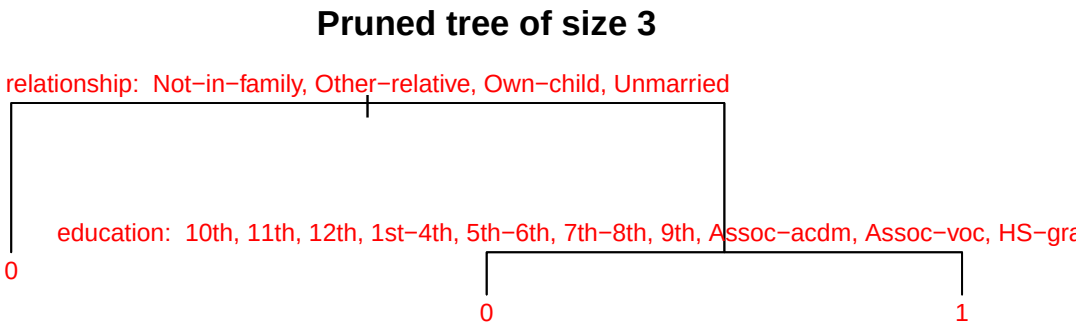
Decision Trees

We begin by fitting a decision tree to the training data, examing the summary and plotting the initial fit. As mentioned above the predictor native_country will be excluded from this model.

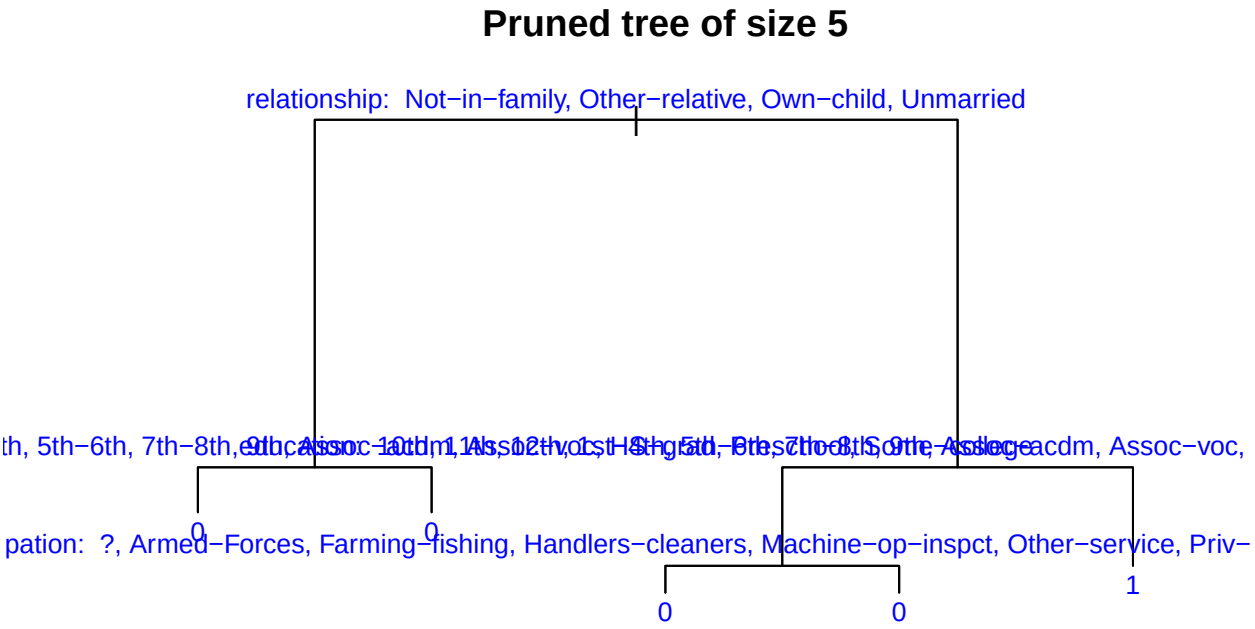
```
##
## Classification tree:
## tree(formula = fifty ~ . - native_country, data = Adult_train)
## Variables actually used in tree construction:
## [1] "relationship" "education"    "occupation"
## Number of terminal nodes:  5
## Residual mean deviance:  0.7824 = 17830 / 22790
## Misclassification error rate: 0.1786 = 4071 / 22792
```



The inital tree has an accuracy of 78.24% with misclassification error rate 17.86%. It contains cuts on the variables “relationship”, “education”, and “occupation” resulting in 5 terminal nodes.



Pruning was performed to improve model performance resulting in a suggested split count of 3 on variables “relationship” and “education”. Pruning results in an improved accuracy, raising it to 82.7% with corresponding misclassification error rate 17.86%. Next, we use K-Fold Cross Validation with K = 10 to develop a third tree.



The resulting split count is 3, on variables relationship, education, and occupation with 5 terminal nodes. The cross validated tree has an accuracy 78.24% and misclassification error rate 17.86%. The accuracy and misclassification error rate are similar to the inital, unadultered tree, which is due to the split count and

number of terminal nodes being identical. Since the pruned tree has the best accuracy, we will call this our best classifier in the method of decision trees, it's performance on the test set will be evaluated.

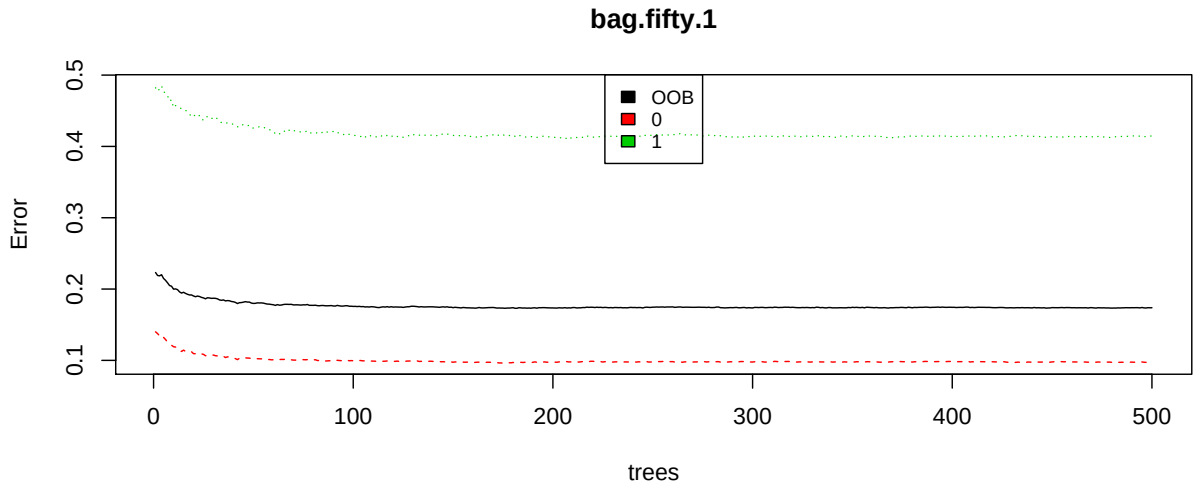
Table 9: Confusion Matrix for Pruned Decision Tree (test set)

	0	1
0	7015	1387
1	401	965

The corresponding missclassification error rate is 18.30467%.

Bootstrap Aggregation and Random Forest

Finally, we move to develop the ensemble learning methods. The bagging model will be fitted first using the random forest function with parameter $m = p$ or mtry $= 11$.

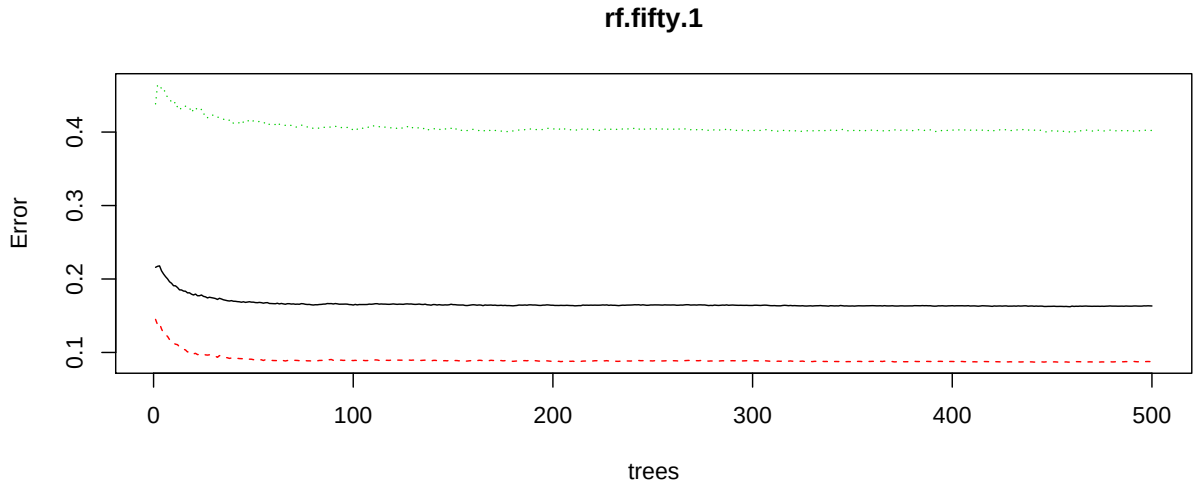



```
##
## Call:
##  randomForest(formula = fifty ~ . - native_country, data = Adult_train,      mtry = 11, importance =
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 11
##
##         OOB estimate of  error rate: 17.38%
## Confusion matrix:
##       0    1 class.error
## 0 15618 1685  0.09738196
## 1  2276 3213  0.41464748
```

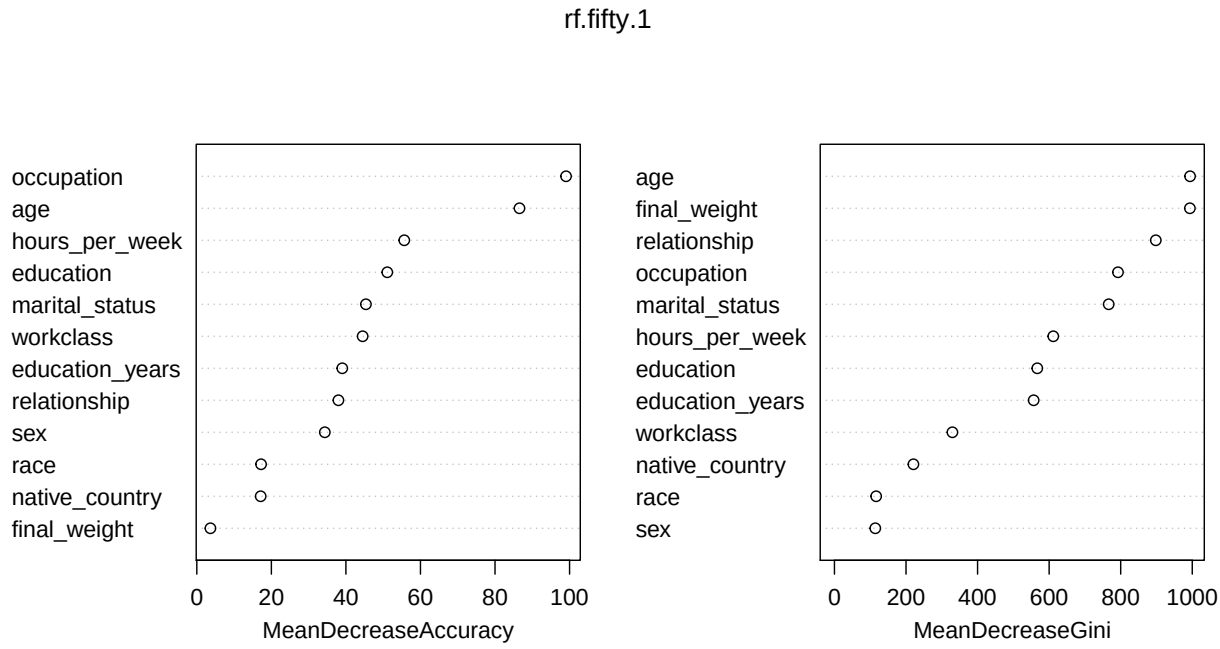
The bagged model has an out of bag error rate of 17.3% and an error rates 9.594% and 41.59% for the clasifications of being above or below 50K respectively.

When applied to the testing data, we see a missclassification error rate 17.34%.

Next we construct the random forest model using $m = \sqrt{p}$.



The random forest has an out of bag error rate at 16.34%, below and above 50K classification error rates 8.8% and 40.12% respectively. We will plot a variable importance plot to develop further insight on the random forest model.



With respect to a mean decrease in accuracy, occupation, age and hours per week are found to be the most important predictors in the random forest model. In contrast, in the case of a mean decrease in gini, age, final_weight, and relationship are the most important variables. With this insight, we can move to evaluate the random forest's performance on the testing set.

When applied to the testing set, the confusion matrix results in a misclassification error rate of 16.216%.

Conclusions

Final Model

We will consider the Random Forest to be the final classification model. It had the lowest misclassification error rate when applied to the test set, coming in at 16.21622% with a corresoinding figure for accuracy 83.78378%. From the final model, we found some of the most important predictors of an individual's income level from the dataset to be Age, Occupation, and Relationship. This is interesting because these were not particularly influential in the development of the first, logistic regression, model.

Limitations

One limitation to the project was the dataset. Since this dataset is semi-outdated, there is some information that it does not contain that would be of much interest in the current geopolitical climate. For instance, almost all of the observation in the dataset have the "race" factor white. This is especially concerning since we know that one of the strongest predictors of one's wealth in real life is their race.

Future Research Directions

As mentioned, the data used to train the classifiers is not the most up to date. More granular and recent data could be utilized to further train the models in order to acheive results more relevant to the present.

Appendix of Project Code

```
#Loading necessary packages
library(caTools)
library(tidyverse)
library(ggplot2)
library(tree)
library(maptree)
library(ROCR)
library(rpart)
library(rpart.plot)
library(ggthemes)
library(GGally)
library(kableExtra)
library(broom)
library(corrplot)
library(caret)
library(partykit)
library(randomForest)
library(gbm)
library(glmnet)
library(stats)
library(kableExtra)
#Reading in dataset and assigning variable names
adult <- read.csv("C:/Users/tanar/Desktop/adult.data")
names(adult) <- c('age', 'workclass','fnlwgt', 'education', 'education-num', 'marital-status', 'occupati

#Convert to tibble
Adult <- as_tibble(adult)
Adult

#Rename some variable names
colnames(Adult)[3] <- "final_weight"
colnames(Adult)[5] <- "education_years"
colnames(Adult)[6] <- "marital_status"
colnames(Adult)[11]<- "capital_gain"
colnames(Adult)[12] <- "capital_loss"
colnames(Adult)[13] <- "hours_per_week"
colnames(Adult)[14] <- "native_country"
colnames(Adult)[15] <- "fifty_thousand"
head(adult)
dim(adult)
str(adult)

#Check for missingness
apply(is.na(as.matrix('adult')),2, sum)

#Turn output into binary variable
Adult <- Adult %>% mutate(fifty = factor(ifelse(Adult$fifty_thousand == levels(Adult$fifty_thousand)[1]

#Breakdown of individuals either above or below the threshold of 50K
fif_table <- with(Adult, table(fifty),2)
fif_table
#Frequency of observations in each response labeling
```

```
knitr::kable(fif_table,"pandoc" ,caption = 'Frequency of observations with incomes either above (1) or
#Age ggpairs graphs
Adult %>%
  dplyr::select(fifty, age) %>%
  ggpairs(aes(col = fifty, fill = fifty, alpha = 0.6),
          upper = list(combo = 'box'),
          diag = list(discrete = 'barDiag'),
          lower = list( combo = 'facethist')) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

#Education Years
Adult %>%
  dplyr::select(fifty, education_years) %>%
  ggpairs(aes(col = fifty, fill = fifty, alpha = 0.6),
          upper = list(combo = 'box'),
          diag = list(discrete = 'barDiag'),
          lower = list( combo = 'facethist')) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
#Remove capital_loss, capital_gain, fifty_thousand
Adult$capital_gain <- NULL
Adult$capital_loss <- NULL
Adult$fifty_thousand <- NULL

#Hours per week exploratory plots
Adult %>%
  dplyr::select(fifty, hours_per_week) %>%
  ggpairs(aes(col = fifty, fill = fifty, alpha = 0.6),
          upper = list(combo = 'box'),
          diag = list(discrete = 'barDiag'),
          lower = list( combo = 'facethist')) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
#GGpairs graphs of final weight
Adult %>%
  dplyr::select(fifty, final_weight) %>%
  ggpairs(aes(col = fifty, fill = fifty, alpha = 0.6),
          upper = list(combo = 'box'),
          diag = list(discrete = 'barDiag'),
          lower = list( combo = 'facethist')) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
marital_table <- prop.table(with(Adult,table(marital_status,fifty)),2);marital_table #Most of the above
#Prop table using marital status compared to response
knitr::kable(marital_table,"pandoc" ,caption = "Proportion of observations in each marital status categ
#Stacked barplot of workclass
attach(Adult)
ggplot(Adult, aes(fill=fifty, x=workclass, y =fifty)) +
  geom_bar(position="stack", stat="identity")
#Stacked barplots of education
ggplot(Adult, aes(fill=fifty, x=education, y =fifty)) +
  geom_bar(position="stack", stat="identity")
#Create prop table of occupation pred
occu_table <- prop.table(with(Adult, table(occupation,fifty)),2)
occu_table
#Prop table using marital status compared to response
```

```r
knitr::kable(occu_table,"pandoc" ,caption = "Proportion of observations in each occupational sector and
#Stacked barplot of race
ggplot(Adult, aes(fill=fifty, x=race, y =fifty)) +
  geom_bar(position="stack", stat="identity")
#Stacked barplot of sex
ggplot(Adult, aes(fill=fifty, x=sex, y =fifty)) +
  geom_bar(position="stack", stat="identity")
#Create prop table of sex
sex_table <- prop.table(with(Adult, table(sex,fifty)),2)
#Make markdown kable table
knitr::kable(sex_table,"pandoc" ,caption = "Proportion of observations of each sex and income level")
#Correlation Plot
correlat<- cor(Adult[c(1,3,5,11)])
corrplot(correlat, method = "shade")
# Split the data into a training and testing set
set.seed(112358)
id <- sample.split(Adult$fifty, SplitRatio = .7)
Adult_train = subset(Adult, id == TRUE)
Adult_test = subset(Adult, id == FALSE)
##FITTED WITH ALL PREDICTORS##
glm_1 <- glm(fifty~ ., data = Adult_train, family = 'binomial')
summary(glm_1) #logistic model including all predictors
#Confusion matrix for total fit
phat_1 <- predict(glm_1,type='response') #Creating the fitted values
probs_1 <- round(phat_1, digits = 2);probs_1
yhat_1 <- phat_1 > 0.5
yhat_1
#yhat_1 outputs 0 and 1
yhat_1 <- as.factor(ifelse(yhat_1 == FALSE,0,1))
cm_1 <- table(obs=Adult_train$fifty,pred=yhat_1)
cm_1
#missclassification rate
mr_1 <- mean(yhat_1 != Adult_train$fifty);mr_1 #0.1623377
#Make markdown kable table of the confusion matrix
knitr::kable(cm_1,"pandoc" ,caption = "Confusion Matrix for Logistic Regression Model 1")
#TPR and FPR
tpr_1 <- cm_1[2,2]/(cm_1[2,1]+cm_1[2,2])
fpr_1 <- cm_1[1,2]/(cm_1[1,1]+cm_1[1,2])
tpr_1 #0.5656768
fpr_1 #0.07605618
#Create and plot the ROC Curve
pred_1 <- prediction(phat_1, Adult_train$fifty)
perf_1 <- performance(pred_1, 'tpr', 'fpr')
plot(perf_1, col = 2, lwd = 3, main = 'ROC Curve (All predictors)');abline(0,1)
# Calculate AUC
auc_1 = performance(pred_1, "auc")@y.values
auc_1 #0.8914733

#Find optimal threshold values
# FPR
fpr_1 = performance(pred_1, "fpr")@y.values[[1]]
cutoff = performance(pred_1, "fpr")@x.values[[1]]
# FNR
```

```r
fnr_1 = performance(pred_1,"fnr")@y.values[[1]]
# Plot
matplot(cutoff, cbind(fpr_1,fnr_1), type="l",lwd=2, main= "Optimal Threshold", xlab="Threshold",ylab="E
# Add legend to the plot
legend(0.3, 1, legend=c("False Positive Rate","False Negative Rate"),
       col=c(1,2), lty=c(1,2))
rate_1 = as.data.frame(cbind(Cutoff=cutoff, FPR=fpr_1, FNR=fnr_1))
rate_1$distance = sqrt((rate_1[,2])^2+(rate_1[,3])^2)
index = which.min(rate_1$distance)
best_1 = rate_1$Cutoff[index]#0.2636124
# Plot
matplot(cutoff, cbind(fpr_1,fnr_1), type="l",lwd=2, xlab="Threshold",ylab="Error Rate")
# Add legend to the plot
legend(0.3, 1, legend=c("False Positive Rate","False Negative Rate"),
       col=c(1,2), lty=c(1,2))
# Add the best value
abline(v=best_1, col=3, lty=3, lwd=3)
#Best cutoff
phat_1_best <- predict(glm_1,type='response')
probs_1_best <- round(phat_1_best, digits = 2);probs_1_best
yhat_1_best <- phat_1_best > best_1
yhat_1_best <- as.factor(ifelse(yhat_1_best == FALSE,0,1))
#confusion matrix
cm_1_best <- table(obs=Adult_train$fifty,pred=yhat_1_best)
cm_1_best
#TPR and FPR
tpr_1_best <- cm_1_best[2,2]/(cm_1_best[2,1]+cm_1_best[2,2])
fpr_1_best <- cm_1_best[1,2]/(cm_1_best[1,1]+cm_1_best[1,2])
tpr_1_best #0.8331208
fpr_1_best #0.2069583
#accuracy and missclassification rate
accuracy.1.best <-(cm_1_best[1,1] + cm_1_best[2,2])/sum(cm_1_best)
accuracy.1.best #0.8026939
mr_1_best <- mean(yhat_1_best != Adult_train$fifty);mr_1_best
mr_1_best <- 1 - accuracy.1.best; mr_1_best #0.1973061
#Make markdown kable table of the confusion matrix
knitr::kable(cm_1_best,"pandoc" ,caption = "Confusion Matrix for Logistic Regression Model 1 (Optimal T
#fit the logistic regression model with only the most significant predictors
glm_2 <- glm(fifty~ age + workclass + final_weight + education + marital_status + occupation + relations
summary(glm_2)
#Confusion matrix for total fit
phat_2 <- predict(glm_2,type='response') #Creating the fitted values
probs_2 <- round(phat_2, digits = 2)
yhat_2 <- phat_2 > 0.5
yhat_2 <- as.factor(ifelse(yhat_2 == FALSE,0,1))
cm_2 <- table(obs=Adult_train$fifty,pred=yhat_2)
cm_2
#TPR and FPR
tpr_2 <- cm_2[2,2]/(cm_2[2,1]+cm_2[2,2])
fpr_2 <- cm_2[1,2]/(cm_2[1,1]+cm_2[1,2])
tpr_2 #0.552924
fpr_2 #0.07524707
#Misclassification Rate
```

```r
mr_2 <- mean(yhat_2 != Adult_train$fifty);mr_2 #0.1647947
#Make markdown kable table of the confusion matrix
knitr::kable(cm_2,"pandoc" ,caption = "Confusion Matrix for Logistic Regression Model 2")
#Create and plot the ROC Curve
pred_2 <- prediction(phat_2, Adult_train$fifty)
perf_2 <- performance(pred_2, 'tpr', 'fpr'); plot(perf_2, col = 2, lwd = 3, main = 'ROC Curve (All predi
# Calculate AUC
auc_2 = performance(pred_2, "auc")@y.values#0.8884319

#Find optimal threshold values
# FPR
fpr_2 = performance(pred_2, "fpr")@y.values[[1]]
cutoff = performance(pred_2, "fpr")@x.values[[1]]
# FNR
fnr_2 = performance(pred_2,"fnr")@y.values[[1]]
# Plot
matplot(cutoff, cbind(fpr_2,fnr_2), type="l",lwd=2, xlab="Threshold",ylab="Error Rate")
# Add legend to the plot
legend(0.3, 1, legend=c("False Positive Rate","False Negative Rate"),
       col=c(1,2), lty=c(1,2))
rate_2 = as.data.frame(cbind(Cutoff=cutoff, FPR=fpr_2, FNR=fnr_2))
rate_2$distance = sqrt((rate_2[,2])^2+(rate_2[,3])^2)
index = which.min(rate_2$distance)
best_2 = rate_2$Cutoff[index]#0.2581948
# Plot
matplot(cutoff, cbind(fpr_2,fnr_2), type="l",lwd=2, xlab="Threshold",ylab="Error Rate")
# Add legend to the plot
legend(0.3, 1, legend=c("False Positive Rate","False Negative Rate"),
       col=c(1,2), lty=c(1,2))
# Add the best value
abline(v=best_2, col=3, lty=3, lwd=3)
#Accuracy of logistic model fitted with all predictors
accuracy.lr.2 <-(cm_2[1,1] + cm_2[2,2])/sum(cm_2)
#0.8352053
mr_2 <- 1 - accuracy.lr.2 #0.1647947

#Best cutoff
phat_2_best <- predict(glm_2,type='response')
probs_2_best <- round(phat_2_best, digits = 2)
yhat_2_best <- phat_2_best > best_2
yhat_2_best <- as.factor(ifelse(yhat_2_best == FALSE,0,1))
#confusion matrix
cm_2_best <- table(obs=Adult_train$fifty,pred=yhat_2_best)
cm_2_best
#TPR and FPR
tpr_2_best <- cm_2_best[2,2]/(cm_2_best[2,1]+cm_2_best[2,2])
fpr_2_best <- cm_2_best[1,2]/(cm_2_best[1,1]+cm_2_best[1,2])
tpr_2_best #0.8325742
fpr_2_best #0.2124487
#accuracy and missclassification rate
accuracy.2.best <-(cm_2_best[1,1] + cm_2_best[2,2])/sum(cm_2_best)
accuracy.2.best #0.7983942
mr_2_best <- mean(yhat_2_best != Adult_train$fifty);mr_2_best
```

```r
mr_2_best <- 1 - accuracy.2.best; mr_2_best #0.2016058
#Make markdown kable table of the confusion matrix
knitr::kable(cm_2_best,"pandoc" ,caption = "Confusion Matrix for Logistic Regression Model 2 (Optimal C
#Test the two models on the test data
#All predictors
phat_1_test <- predict(glm_1,Adult_test,type='response')
yhat_1_test <- phat_1_test > .5
yhat_1_test <- as.factor(ifelse(yhat_1_test == FALSE,0,1))
mr_1_test <- mean(yhat_1_test!=Adult_test$fifty); mr_1_test #0.206593

#Reduced model (only significant predictors)
phat_2_test <- predict(glm_2,Adult_test,type='response')
yhat_2_test <- phat_2_test > 0.5
yhat_2_test <- as.factor(ifelse(yhat_2_test == FALSE,0,1))
yhat_2_test
mr_2_test <- mean(yhat_2_test!=Adult_test$fifty); mr_2_test #0.2118141
#Fit classification tree with all predictors except native_country since it has more than 32 levels
tree.1 <- tree(fifty~. - native_country, data = Adult_train)
summary(tree.1)
#Plot the fit
plot(tree.1); text(tree.1, pretty = 0, cex = .6); title("Classification Tree Using all Predictors (excep
#Nicer plot
form <- as.formula("fifty ~ age + workclass + education + marital_status +
  occupation + relationship + race + sex  + hours_per_week")
mod.tree.1 <- rpart(form, data = Adult_train); mod.tree.1
plot(mod.tree.1)
text(mod.tree.1, use.n = TRUE, all = TRUE, cex = 0.7)
plot(as.party(mod.tree.1))
#Compute missclassification error rate
yhat.testset.1 <- predict(tree.1, Adult_test, type="class"); yhat.testset.1
# Obtain confusion matrix
tree.1.error <- table(yhat.testset.1, Adult_test$fifty); tree.1.error
# Test accuracy rate
tree.1.accuracy <- sum(diag(tree.1.error))/sum(tree.1.error); tree.1.accuracy #0.8260647

tree.1.mr <- 1 - tree.1.accuracy; tree.1.mr #0.1830467
#Pruning
prune.tree.1 <- prune.tree(tree.1, k = 0:20, method = "misclass")
# Best size
best.prune.tree.1 <- prune.tree.1$size[which.min(prune.tree.1$dev)]; best.prune.tree.1
#best tree determined by pruning
tree.1.pruned <- prune.misclass(tree.1, best=best.prune.tree.1)
# Plot pruned tree
plot(tree.1.pruned);text(tree.1.pruned, pretty=0, col = "red", cex = .8);title("Pruned tree of size 3")
summary(tree.1.pruned)
#K-fold cross validation
cv.tree.1 <- cv.tree(tree.1, FUN=prune.misclass, K=10)
# Print out cv
cv.tree.1
# Best size
best.cv.1 <- cv.tree.1$size[which.min(cv.tree.1$dev)]
best.cv.1
#best tree determined by cv
```

```r
tree.1.cv.best <- prune.misclass (tree.1, best=best.cv.1)
# Plot pruned tree
plot(tree.1.cv.best);text(tree.1.cv.best, pretty=0, col = "blue", cex = .8);title("Pruned tree of size 
summary(tree.1.cv.best)
#predict on test set
pred.tree.1.prune <- predict(tree.1.pruned, Adult_test, type="class")
# Obtain confusion matrix
err.tree.1.prune <- table(pred.tree.1.prune, Adult_test$fifty); err.tree.1.prune
# Test accuracy rate
tree.1.pruned.accuracy <- sum(diag(err.tree.1.prune))/sum(err.tree.1.prune)
#Missclassification error rate
mr.tree.1.pruned <- 1 - tree.1.pruned.accuracy; mr.tree.1.pruned #0.1830467
#Make markdown kable table of the confusion matrix
knitr::kable(err.tree.1.prune,"pandoc" ,caption = "Confusion Matrix for Pruned Decision Tree (test set)
#Bagging
bag.fifty.1 = randomForest(fifty ~ .-native_country, data=Adult_train, mtry=11, importance=TRUE)
#Plot
plot(bag.fifty.1)
legend("top", colnames(bag.fifty.1$err.rate),col=1:4,cex=0.8,fill=1:4)
bag.fifty.1
#Test on new data
yhat.bag.1 = predict(bag.fifty.1, newdata = Adult_test)
# Confusion matrix
bag.err.1 = table(pred = yhat.bag.1, truth = Adult_test$fifty)
test.bag.err.1 = 1 - sum(diag(bag.err.1))/sum(bag.err.1)
test.bag.err.1
#Grow random forest
#mtry is set to approximately = sqrt(11)
rf.fifty.1 = randomForest(fifty ~ ., data=Adult_train, mtry=3, ntree=500, importance=TRUE);
plot(rf.fifty.1)
yhat.rf.1 = predict (rf.fifty.1, newdata = Adult_test)
importance(rf.fifty.1)
varImpPlot(rf.fifty.1)
# Confusion matrix
rf.err.1 = table(pred = yhat.rf.1, truth = Adult_test$fifty)
test.rf.err.1 = 1 - sum(diag(rf.err.1))/sum(rf.err.1);test.rf.err.1 #0.1621622
```