Wei-Hsin Lin (Philip Lin)

Prof. Gaston Sanchez

Stat 154

6 Dec. 2017

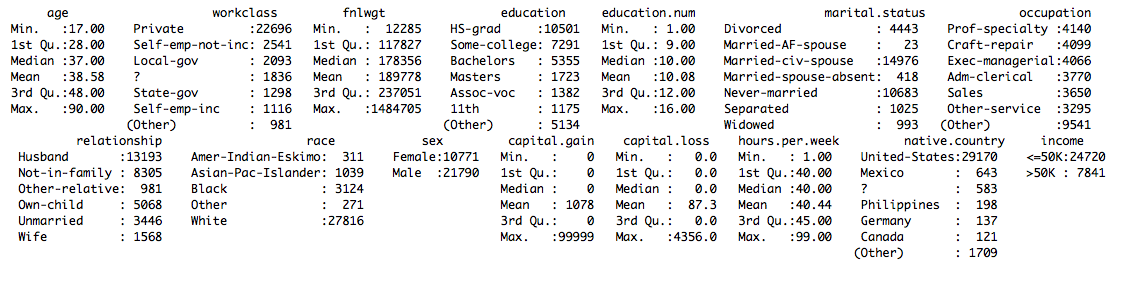
Final Project Report

**Introduction:**

Income level has become a topic people pay a lot of attention to. In this project, I will analyze people’s income level, below or above 50,000, from census income data set denoted by Ronny Kohavi and Barry Becker to the UCI Machine Learning Repository. The features I will use to for the analysis are age, work class, final weight, marital status, occupation, relationship in the family, race, sex, capital gain, capital loss, working hours per week, and native country. The goal of this project is to use the feature to build a reasonable model in order to be able to determine if a new observation with all the features stated above has income less than or more than 50,000. To accomplish the goal, I am going to utilize classification tree model, bagging tree, and random forest, and, in the end, I will pick a model that fits the data the best to consider as the final model.

**Data Cleaning and EDA:**

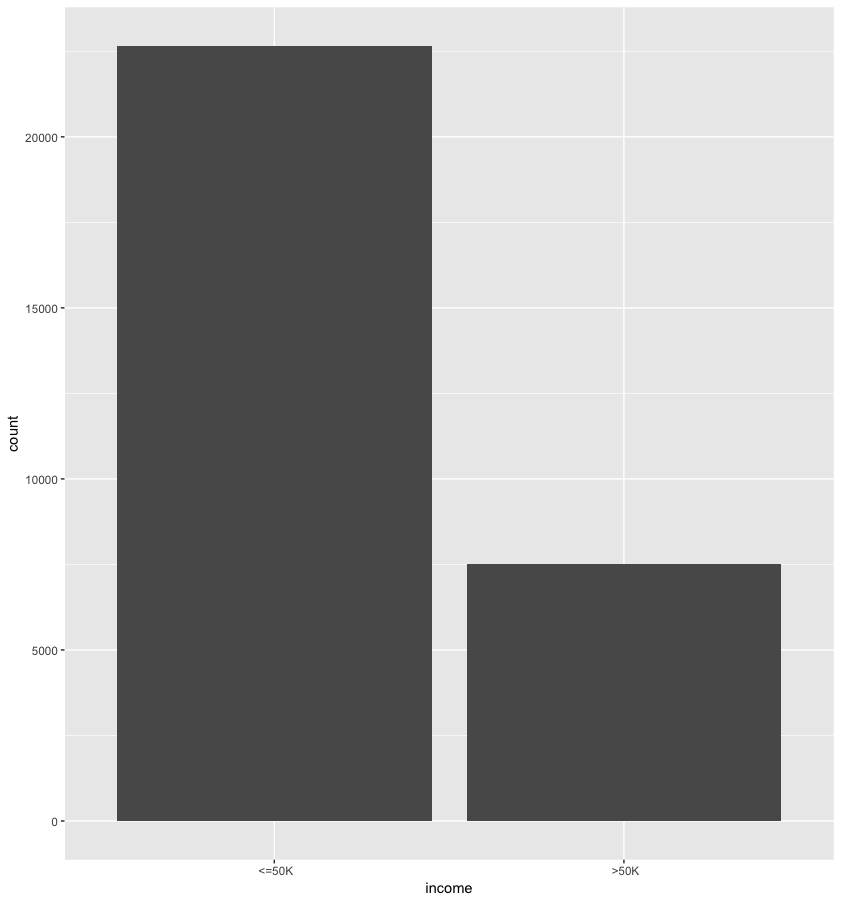
I will begin with preprocessing (i.e. data cleaning) and exploratory data analysis. First of all, I know that the dataset has 32561 observations by simply check the number of rows in the dataset. Then, by looking at the summary statistics (figure shown below) of the training data, we can see there are some “?”, which indicates missing values, in some of the variable. As a result, I will need to investigate further to handle these missing values.

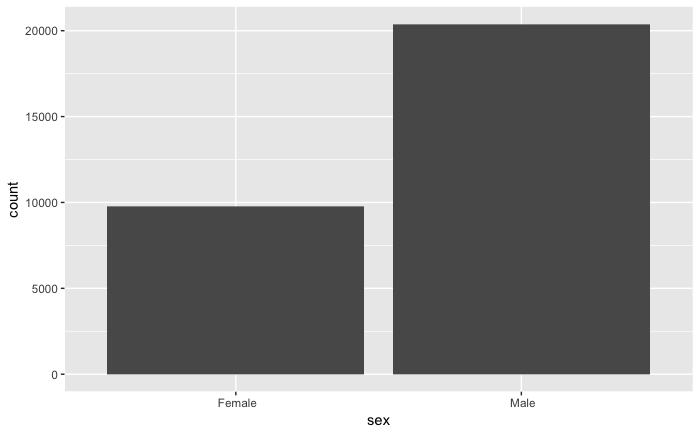


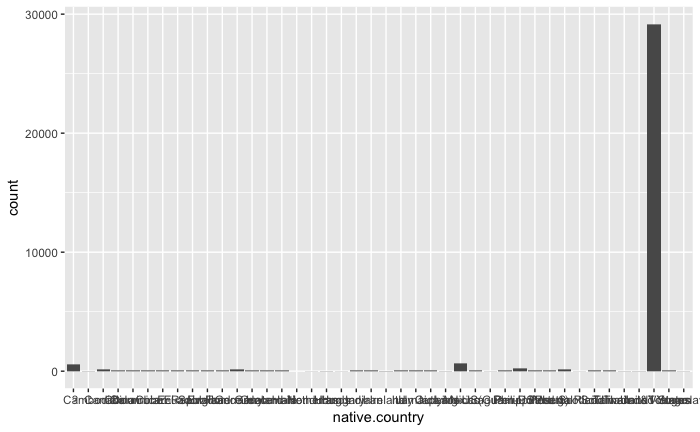
By looking at the summary statistic and the unique values in categorical variables, there seems to be no missing values in variables age, fnlwgt, education, education-num, marital-status, relationship, race, sex, hours-per-week, and income. Therefore, for these variables, we do not need to handle the missing values. Also, I found that there are 29849 of the data in capital gain equal to 0, and 31042 of the data in capital loss equal to 0. Therefore, I decide to omit these two features for building the model later on since they may not provide useful information since most of the data are missing. Furthermore, I noticed that the missing values seems to be non-systematic (i.e. data missing just due to chance), and the number observations having missing values is not large (1836 in workclass, 1843 in occupation, and 583 in native country) comparing to the total number of observations 32561. Accordingly, I choose to simply remove these individuals from the dataset.

Some people bin the variables to maybe make their classification models run more efficiently. However, since discretizing variables may lose some information, I just decide not to do so.

In addition, I change the scale of all the numeric variables into standard unit (i.e. subtract by their mean and divide by their standard deviation). By doing so, all the variables have the same scale, and doing to can make the variable comparisons much easier.

To further explore the variables, a good approach is to visualize the distributions of each variables. I would like to plot the numeric variables with density plots and categorical features with boxplots. First of all, I look at the plot for income, and notice that the number of people who earn less than 50K is way more than the number of people who earn more than in this dataset. I may need to take this fact into account when I build the model and/or make the conclusion. 

Then, we can look at the plot for sex. We can see that there are more males than females in the dataset. This may raise up a question: combining the fact that there are more people earn less that 50K than more than 50K, does that mean males earn less than females? I will say no. Before we do any data analysis, we can guarantee anything. Also, this dataset may not be collected at random, so there may be a lot of different factors that we have not explore yet that may affect the income. 

We can see that most of the individuals in the dataset are from the United State since the dataset came from US Census data. 

Now, we can further look at the correlation between explanatory variables and response variable. We can see that although basically all the variables are not highly correlated to the response variable. However, since I am going to build the model with not just one variable, I can still use all these variables to build the classification models.

[1] "The correlation between age and income is 0.242"

[1] "The correlation between workclass and income is 0"

[1] "The correlation between fnlwgt and income is -0.009"

[1] "The correlation between education and income is 0.079"

[1] "The correlation between education.num and income is 0.3353"

[1] "The correlation between marital.status and income is -0.1935"

[1] "The correlation between occupation and income is 0.0516"

[1] "The correlation between relationship and income is -0.251"

[1] "The correlation between race and income is 0.0717"

[1] "The correlation between sex and income is 0.2167"

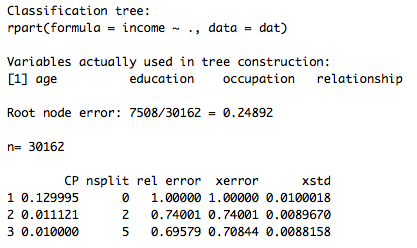
[1] "The correlation between hours.per.week and income is 0.2295"

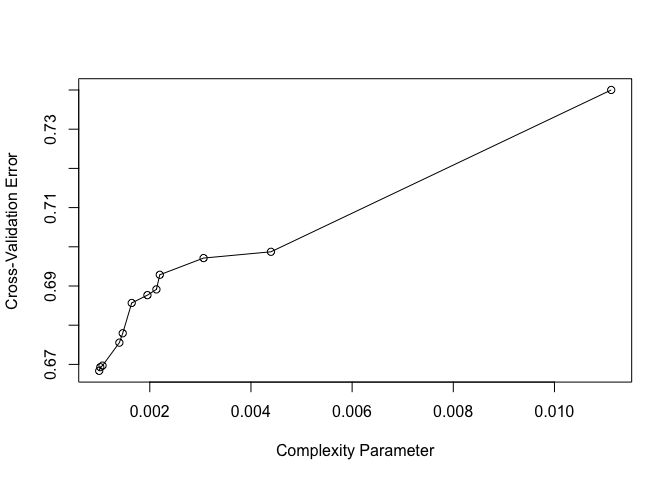
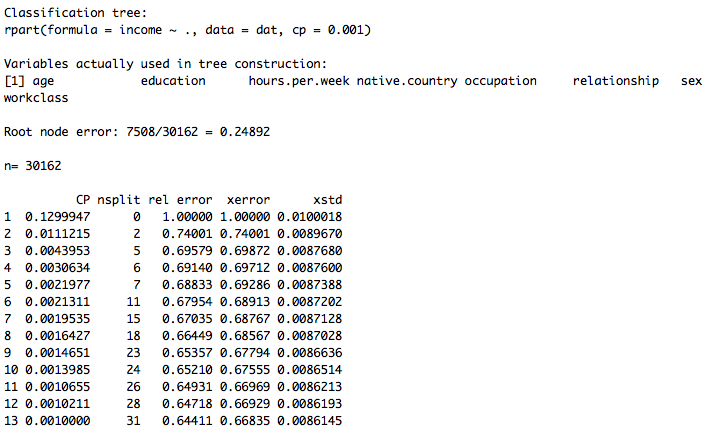
[1] "The correlation between native.country and income is 0.0233"

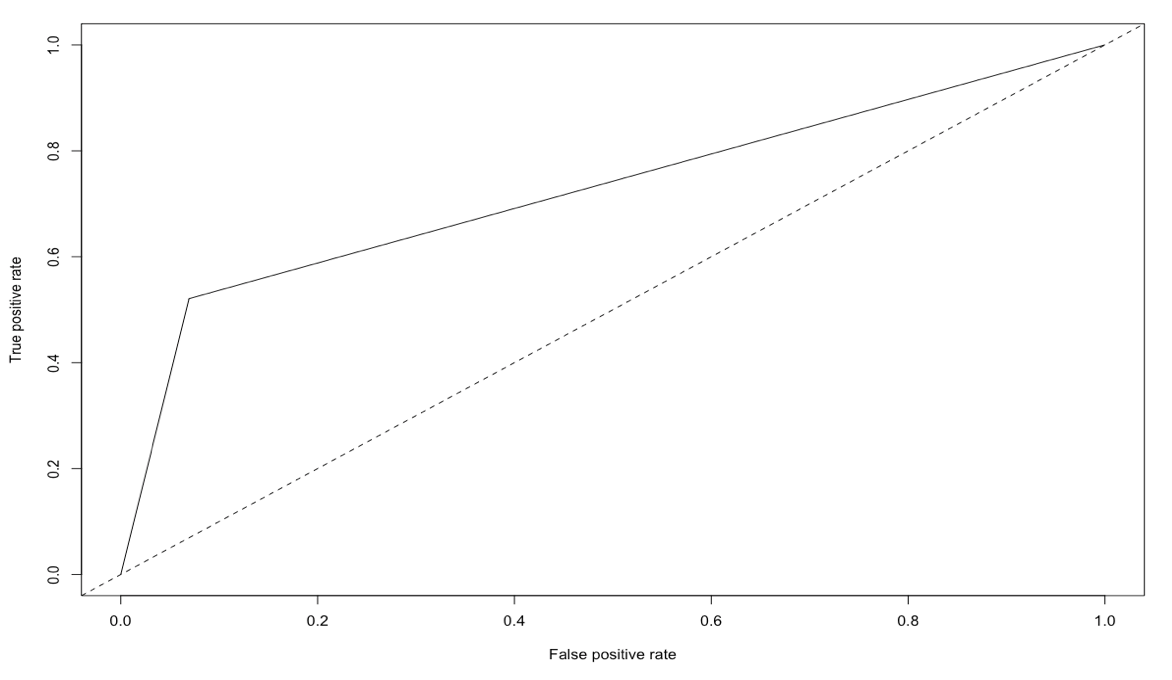
**Model Building:**

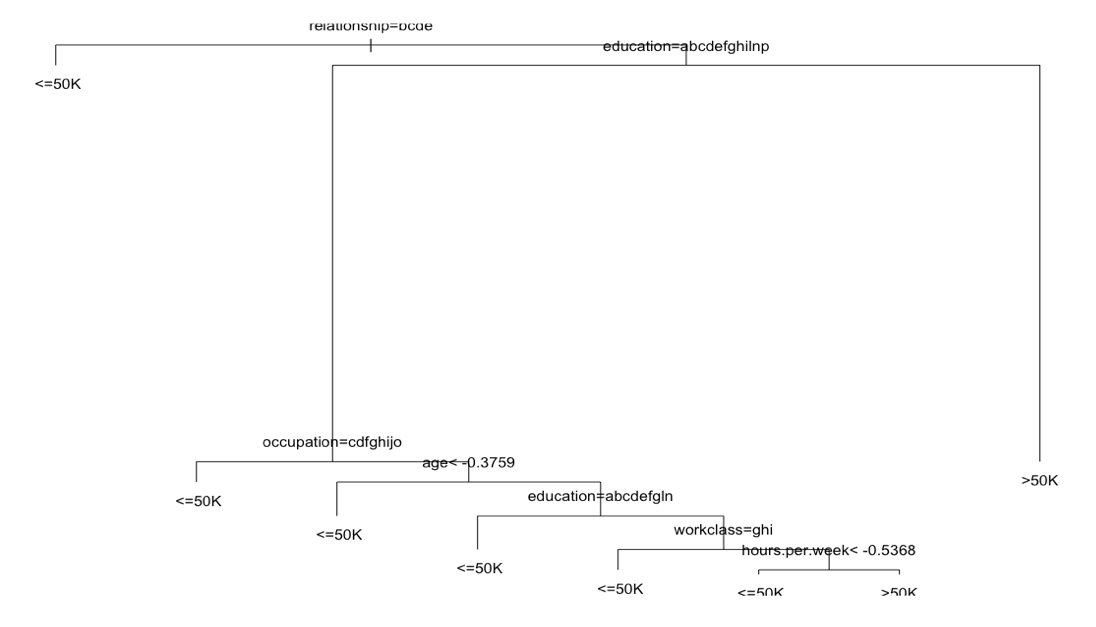
**Classification Tree**

In this part, I will use the R package “rpart” to build the classification model, and I will try to find out the best hyper parameter in order to fit the model more to the data. At the same time, I will need to prevent from overfitting the model.

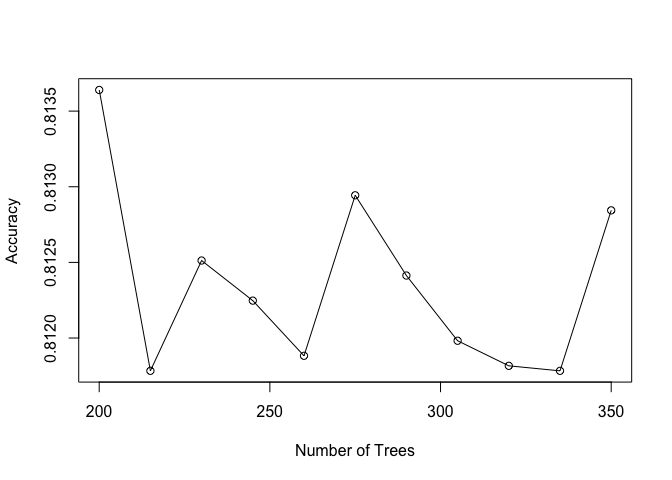
First of all, I fit the classification tree with default values in rpart. The default complexity parameter is 0.01. And, within rpart, I internally use cross validation to test on complexity parameter. We can see the complexity parameter table shown below. The way to read the cross-validation errors is that we look at the column “xerror” and multiply it be root node error (0.24892 in this case). However, we can simply look for the minimum values from “xerror” column since multiply by a scaler greater than 1 does not affect the sorted order of numbers. 

Knowing that there are actually only two complexity parameters that I can really choose since the one with no split is not useful, I redo the cross-validation process by setting the complexity parameter to 0.001 instead of default 0.01. Now, we can see from the table shown below, all the complexity parameters less than 0.003 have almost no difference in cross-validation errors (by taking the “xerror” column times root node error). As a result, I can choose any of them which will give me around the same cross-validation error. Furthermore, in order to prevent from overfitting, I need to choose a complexity parameter that does not give a high number of splits. Therefore, I decide to choose 0.0022 as my complexity parameter which gives me around 7 splits. 

One way to measure whether a model is good is to look at the ROC (Receiver Operating Characteristic) curve (shown below) and the AUC (area under curve). The model selected has AUC of 0.7258, which is not so good, but it’s not so bad as well. 

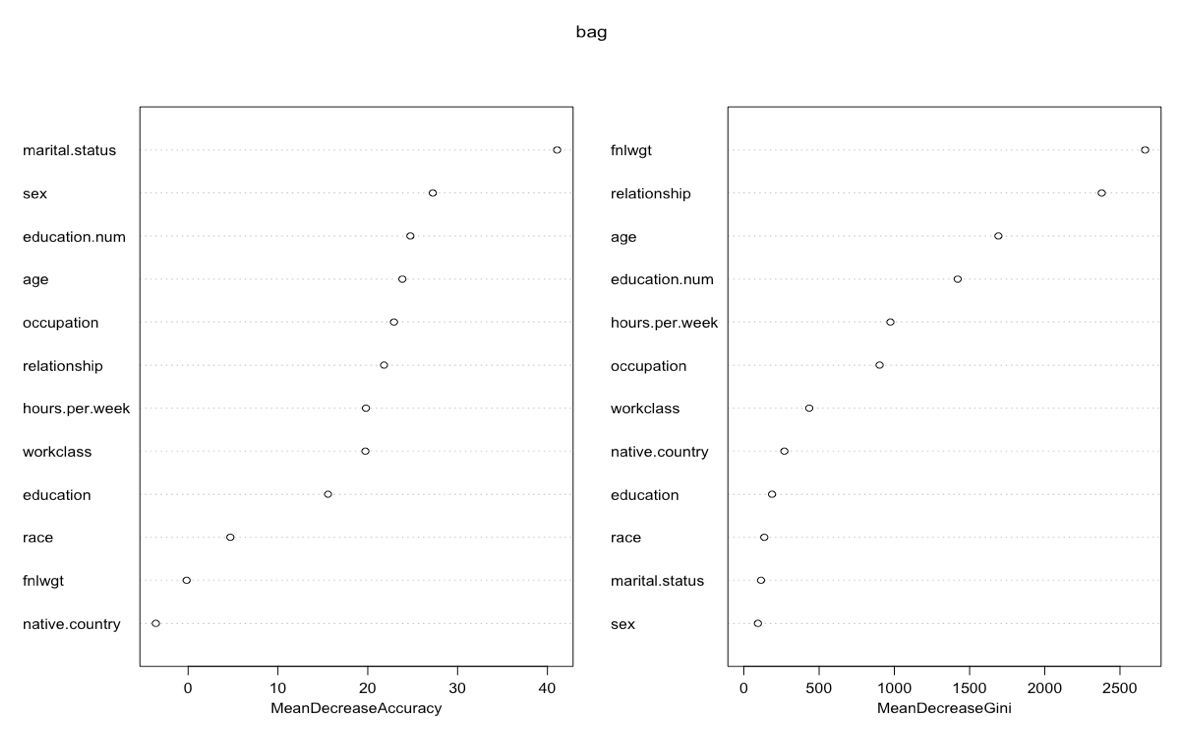
To summarize, the classification tree model I select has complexity parameter of 0.0022 with 7 splits. It has an in-sample accuracy of around 83%. 

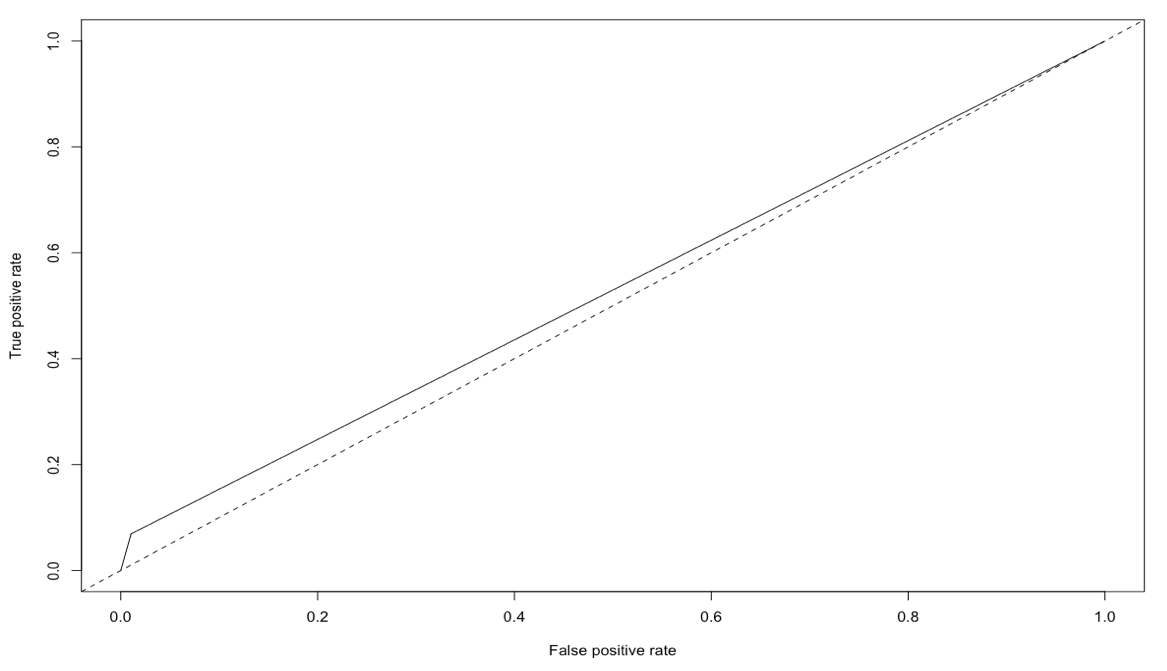
**Bagged Tree**

The idea is that build a random forest with all the features, but I need to choose the number of trees in the forest via parameter tuning. The way I choose to tune the hyperparameter is to utilize cross-validation. First of all, a reason range for a random forest is 200 to 350, in my opinion. After running cross-validation with sequence 200, 215, …, 350, 200 gives me the lowest cross-validation error although they are all really close (plot of cv accuracy shown below). The reason that the cross-validation errors are close may be random forest models predict the results base on majority votes. 

We can also look at the importance of each variable to the model. By looking at mean decrease accuracy, the 5 most important features and its variable importance statistics are shown below. ../../../../../Desktop/bag_MDA.pn

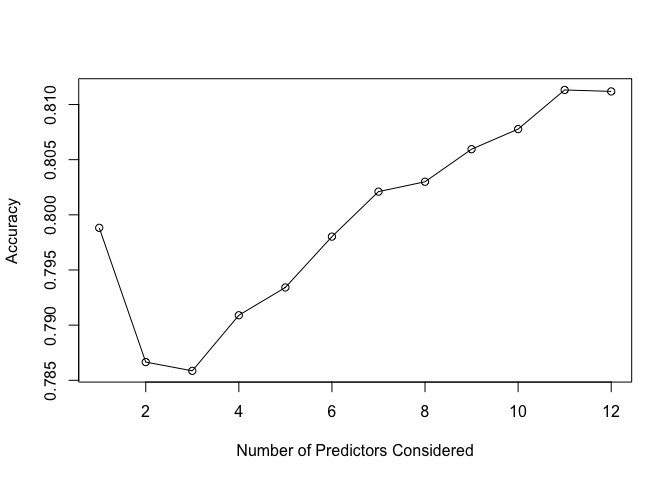
However, by looking at mean decrease gini, the 5 most important features and its variable importance statistics are shown below.../../../../../Desktop/bag_gini.pn

We can also see a more explicit plot below to see the importance statistics. 

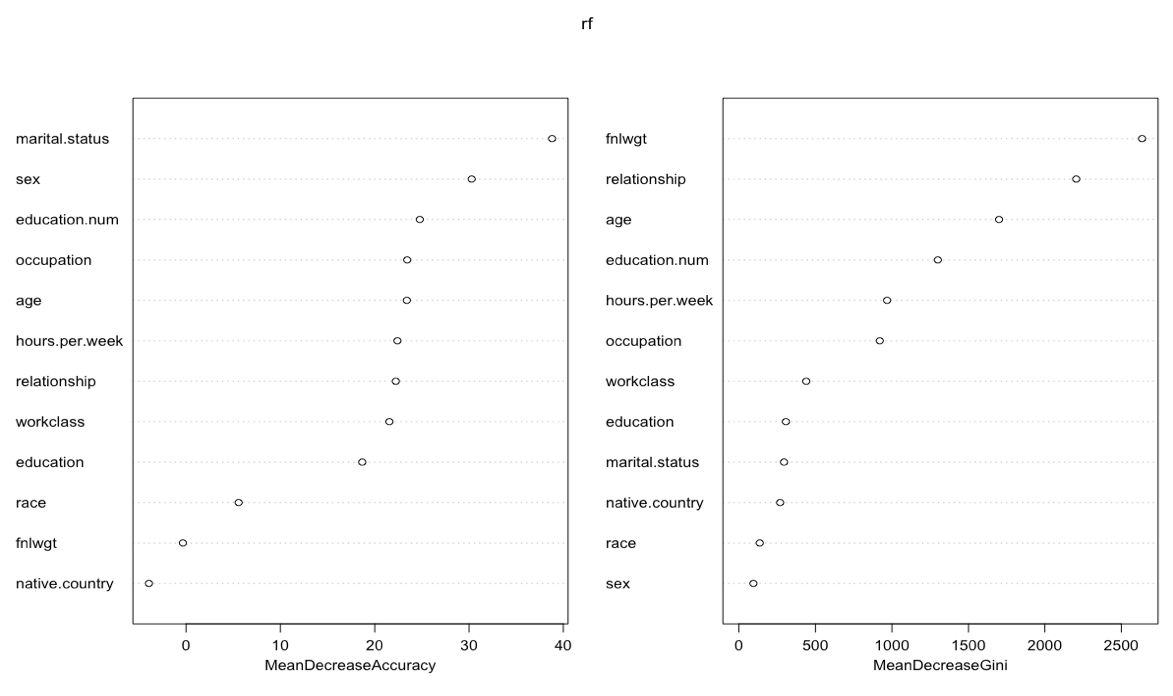
Again, ROC curve and its AUC is a way to measure a model. The bagged tree model selected has AUC of 0.5294. 

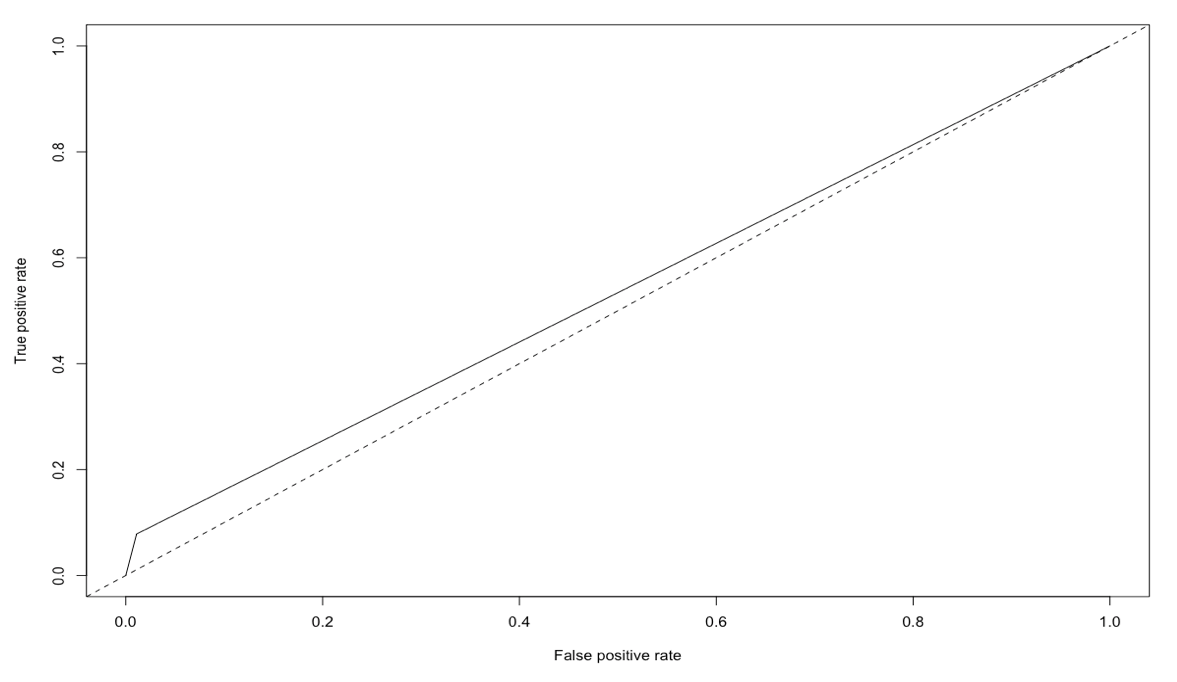
In conclusion, the bagging tree model selected has a structure of 200 trees, and it achieves an accuracy of 76%.

**Random Forest**

There are two hyperparameters to tune which are number of trees in the forest and number of features used to construct the trees. Since I have already tuned the number of trees while working with bagging tree, I will just keep the number of trees the same as before. Again, I utilize cross-validation to tune the number of features, and the one gives me the highest cross-validation accuracy is to use 11 features (plot of cv accuracy shown below). 

We can also see how important is a variable to this model. By looking at mean decrease accuracy, the 5 most important features and its variable importance statistics are shown below. ../../../../../Desktop/rf_MDA.pnHowever, by looking at mean decrease gini, the 5 most important features and its variable importance statistics are shown below. ../../../../../Desktop/rf_gini.pn

We can also see the plot for the importance statistics. 

One way to measure whether a model is good is to look at the ROC curve (shown below) and the AUC. The model has AUC of 0.5337. 

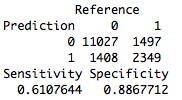
In summary, the random forest model chosen has structures of 200 trees, and uses 11 variables to build each tree. It has an in-sample arruracy of 76%.

Model Selection:

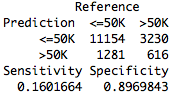
In conclusion, I have a classification tree model, a bagging tree model, and a random forest model, and want to pick a model that best describes the data. In order to measure how good a model is better, I use the models trained above to predict on the test set, which has the same explanatory variables. Since I did not build the model from any observation in the test set, the measure has lower bias on performance measures.

First of all, we use the models to predict the income from the test set, and we can look at the confusion matrix, specificity, and sensitivity from each model.

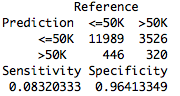
Classification Tree:



Bagged Tree:



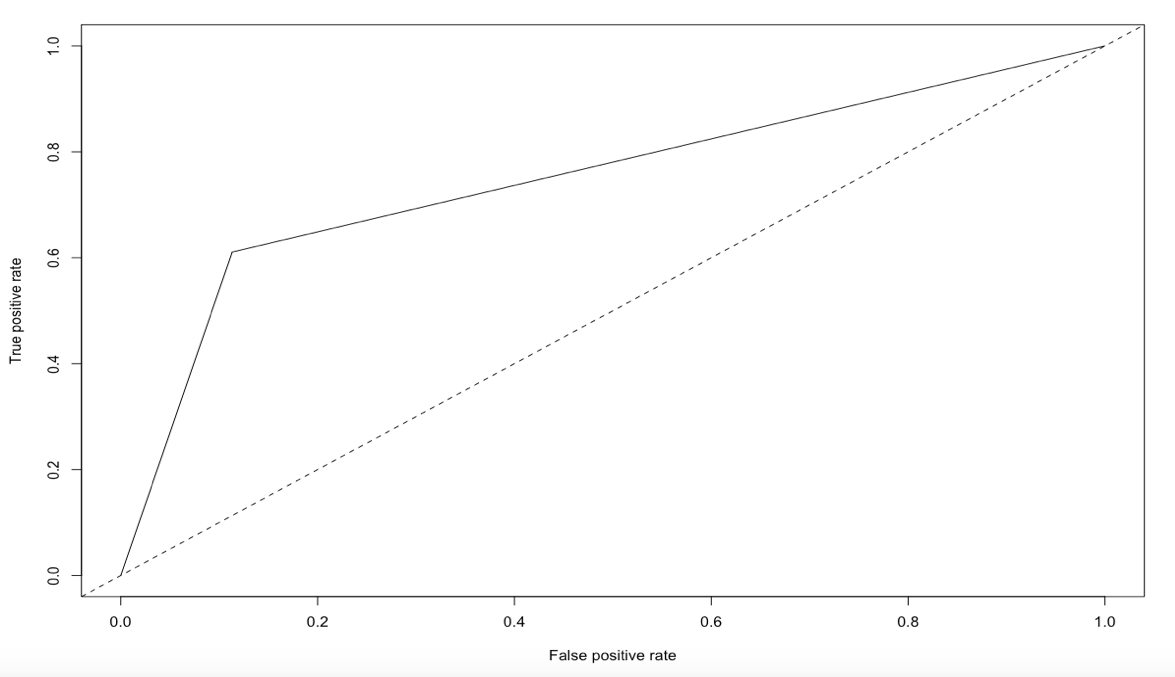
Random Forest:



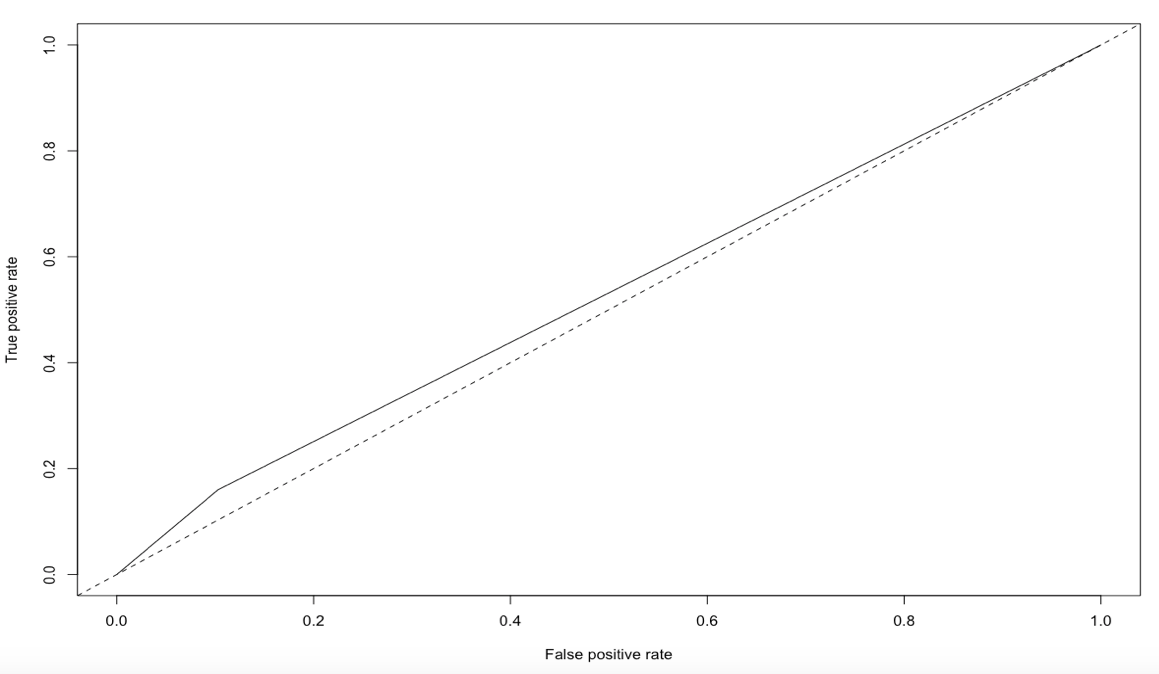
Additionally, we can also simply look at the test set accuracy. Classification tree model achieves an accuracy of 82.16%, bagged tree models has accuracy of 72.29%, and random forest has 75.6% accuracy.

Furthermore, we can look at the ROC curve and AUC.

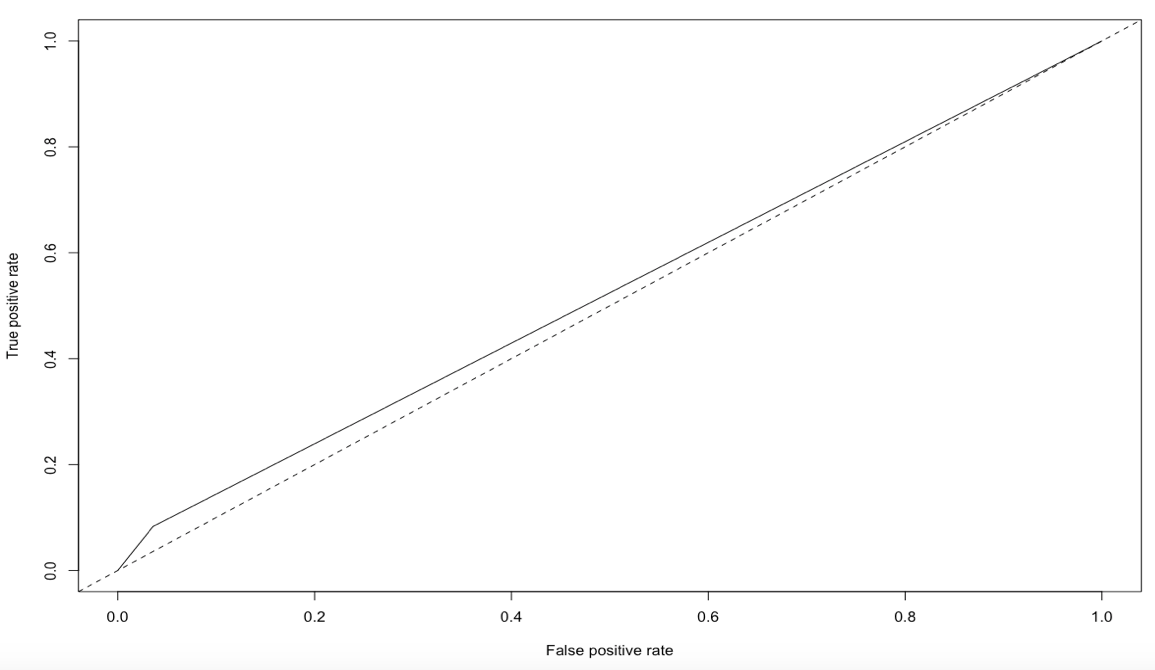
Classification Tree:



Bagged Tree:



Random Forest:



../../../../../Desktop/AUC_test.pn

In conclusion, by comparing confusion matrix, specificity, sensitivity, testing accuracy, ROC curve, and AUC, I will select the classification tree with complexity parameter of 0.0022 as my final model to fit this dataset.