Pset-6-Katia-Williams

Katia Williams

4/25/2018

```r
train_raw <- read.csv("/Users/katiawilliams/154/training.csv")
test_new <- read.csv("/Users/katiawilliams/154/test_new.csv")
```

```r
train <- train_raw[, c(2:14)]
head(train)
```

```
##   income age capital_gain capital_loss hours_per_week      workclass
## 1      0  39          Low         None             40      Other-gov
## 2      0  50         None         None             13  Self-Employed
## 3      0  38         None         None             40        Private
## 4      0  53         None         None             40        Private
## 5      0  28         None         None             40        Private
## 6      0  37         None         None             40        Private
##      education marital.status      occupation   relationship  race     sex
## 1    Bachelors  Never-Married  Administration  Not-in-family White    Male
## 2    Bachelors        Married      Management        Husband White    Male
## 3  HS-Graduate    Not-Married     Blue-Collar  Not-in-family White    Male
## 4      Dropout        Married     Blue-Collar        Husband Black    Male
## 5    Bachelors        Married    High-Service           Wife Black  Female
## 6      Masters        Married      Management           Wife White  Female
##          native.country
## 1         United-States
## 2         United-States
## 3         United-States
## 4         United-States
## 5 South-America-Frontier
## 6         United-States
```

```r
test <- test_new[,c(3:15)]
head(test)
```

```
##   income         age capital_gain capital_loss hours_per_week    workclass
## 1      0 -1.02897097         None         None    -0.07888642      Private
## 2      0 -0.05742062         None         None     0.75010635      Private
## 3      0 -0.35635919         None         None    -0.90787919      Private
## 4      0 -1.10370561         None         None    -0.07888642      Private
## 5      0  1.21306829         None         None    -2.56586474      Private
## 6      0 -0.20688990         None         None    -0.07888642  Federal-gov
##      education marital.status      occupation   relationship  race     sex
## 1      Dropout  Never-Married    High-Service      Own-child Black    Male
## 2  HS-Graduate        Married     Blue-Collar        Husband White    Male
## 3      Dropout  Never-Married         Service  Not-in-family White    Male
## 4  HS-Graduate  Never-Married         Service      Unmarried White  Female
## 5      Dropout        Married     Blue-Collar        Husband White    Male
## 6    Bachelors        Married  Administration        Husband White    Male
##   native.country
## 1  United-States
## 2  United-States
```

```
## 3  United-States
## 4  United-States
## 5  United-States
## 6  United-States
```

```r
nrow(train)
```

```
## [1] 30155
```

EDA

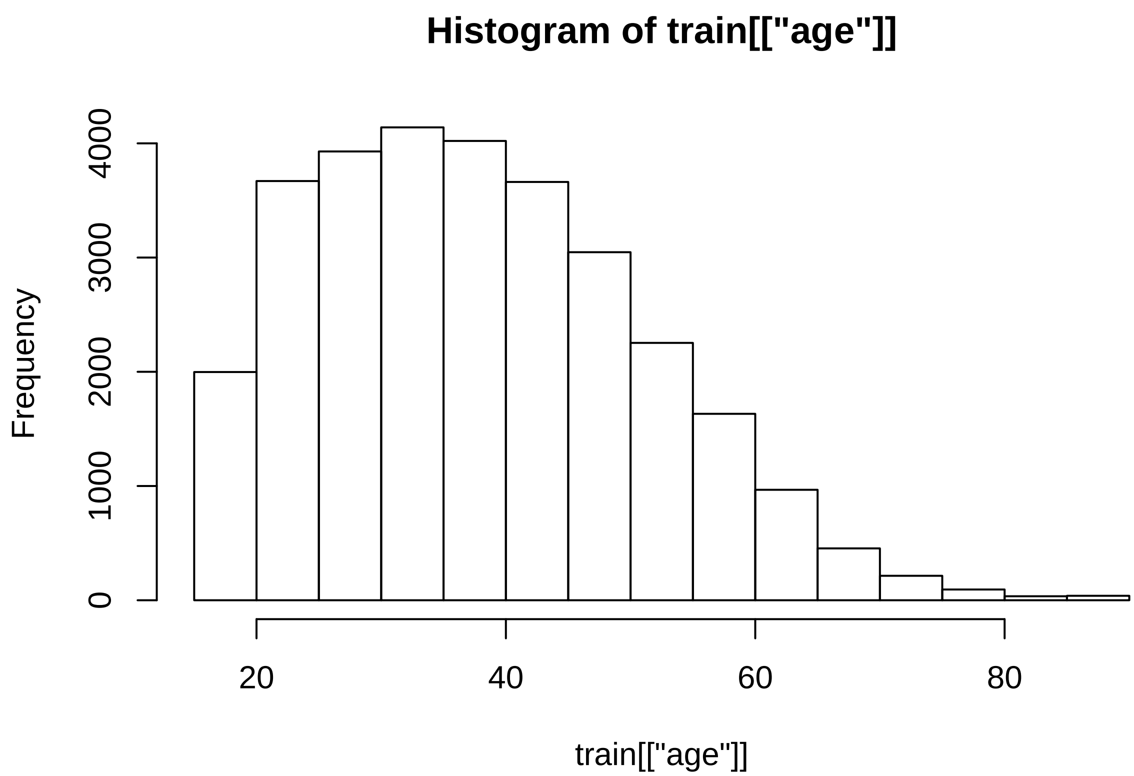
```r
#Summary
summary(train)
```

```
##      income            age         capital_gain capital_loss
##  Min.   :0.0000   Min.   :17.00   High: 1090   High:  686  
##  1st Qu.:0.0000   1st Qu.:28.00   Low : 1448   Low :  734  
##  Median :0.0000   Median :37.00   None:27617   None:28735  
##  Mean   :0.2489   Mean   :38.43                            
##  3rd Qu.:0.0000   3rd Qu.:47.00                            
##  Max.   :1.0000   Max.   :90.00                            
##                                                            
##  hours_per_week           workclass             education    
##  Min.   : 1.00    Federal-gov  :  942    Associates : 2315  
##  1st Qu.:40.00    Not-Working  :   14    Bachelors  : 5044  
##  Median :40.00    Other-gov    : 3345    Doctorate  :  374  
##  Mean   :40.93    Private      :22281    Dropout    : 3739  
##  3rd Qu.:45.00    Self-Employed: 3573    HS-Graduate:16514  
##  Max.   :99.00                           Masters    : 1627  
##                                          Prof-School:  542  
##        marital.status            occupation           relationship  
##  Married      :14086    Administration:3720    Husband       :12463  
##  Never-Married: 9725    Blue-Collar   :9906    Not-in-family : 7724  
##  Not-Married  : 5518    High-Service  :4035    Other-relative:  888  
##  Widowed      :  826    Management    :3991    Own-child     : 4465  
##                         Sales         :3584    Unmarried     : 3209  
##                         Service       :4919    Wife          : 1406  
##                                                                     
##           race            sex                        native.country 
##  Amer-Indian:  286    Female: 9776   United-States          :27497  
##  Asian      :  895    Male  :20379   South-America-Emerging:   970  
##  Black      : 2817                   Western-Developed      :   466  
##  Other      :  231                   Asia-Emerging          :   273  
##  White      :25926                   South-America-Frontier:   242  
##                                      Asia-Frontier          :   199  
##                                      (Other)                :   508
```

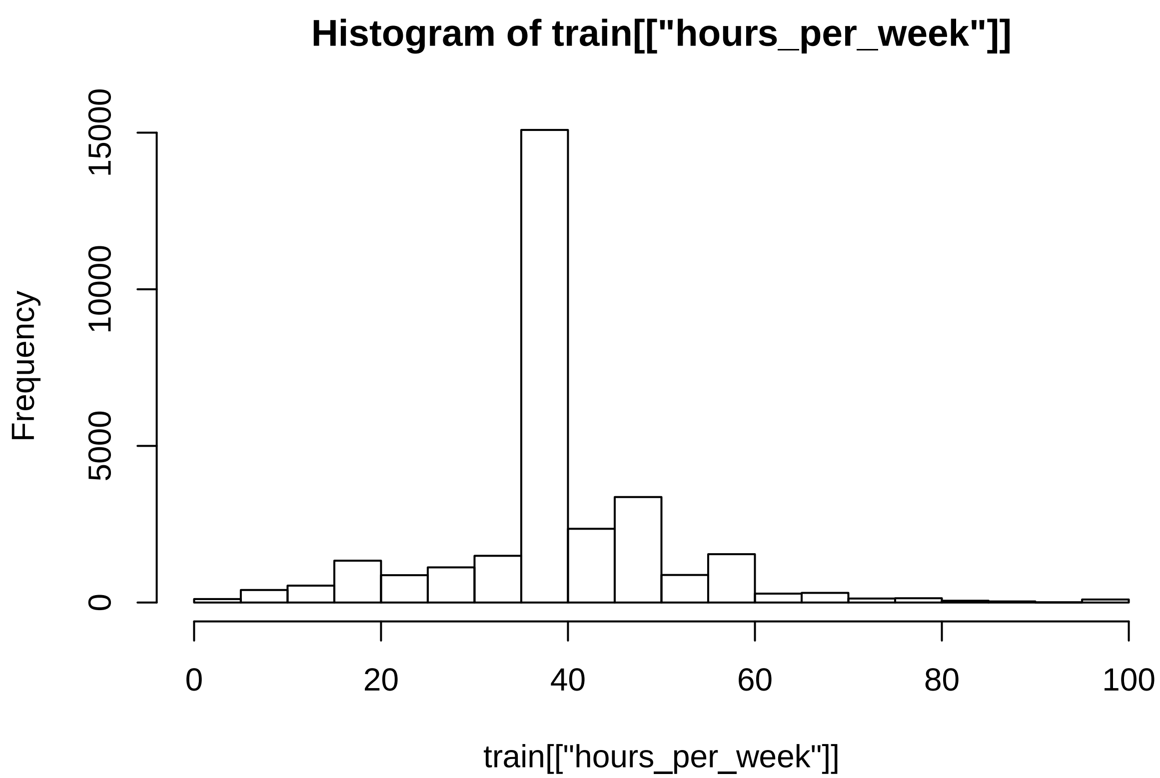
```r
nrow(train)
```

```
## [1] 30155
```

```r
#Histogram(s)
hist(train[['age']])
```




```r
hist(train[['hours_per_week']])
```

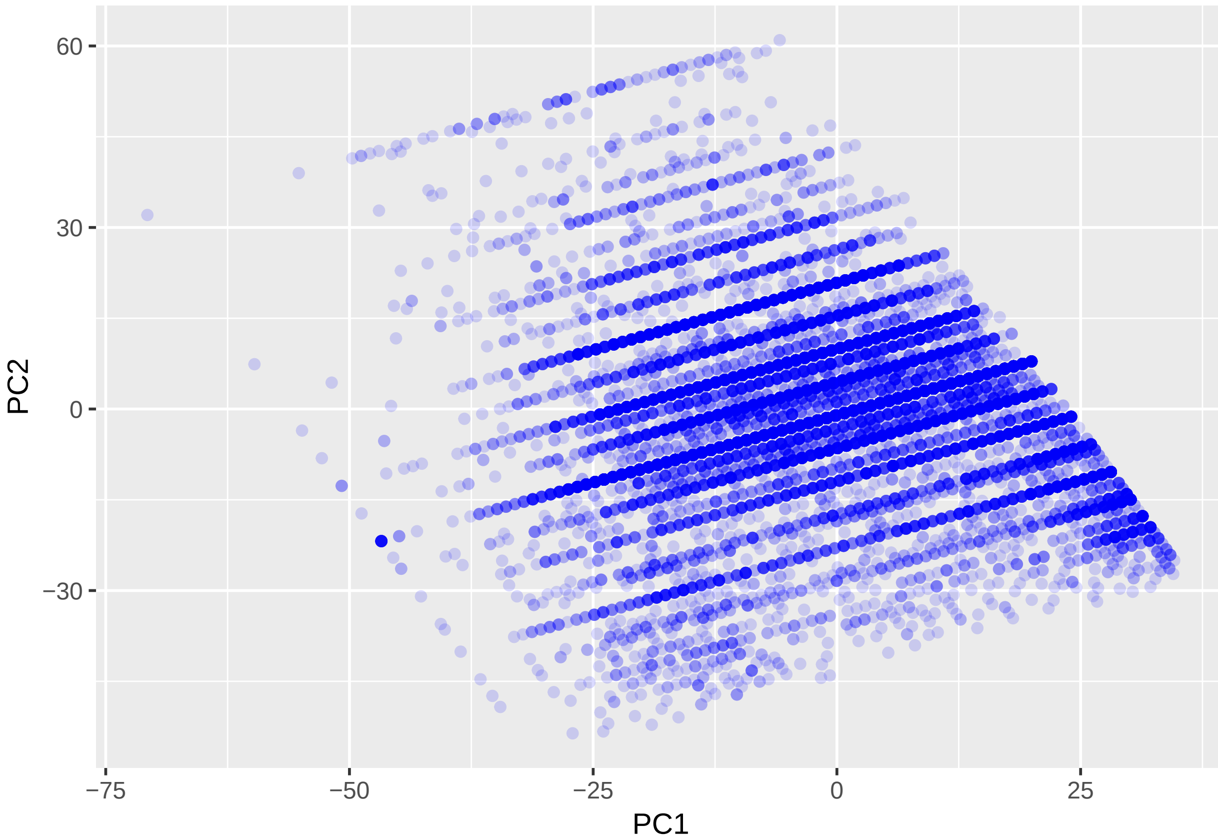



```r
cor(train[, c(2,5)])
```

```
##                     age hours_per_week
## age            1.000000       0.102033
```

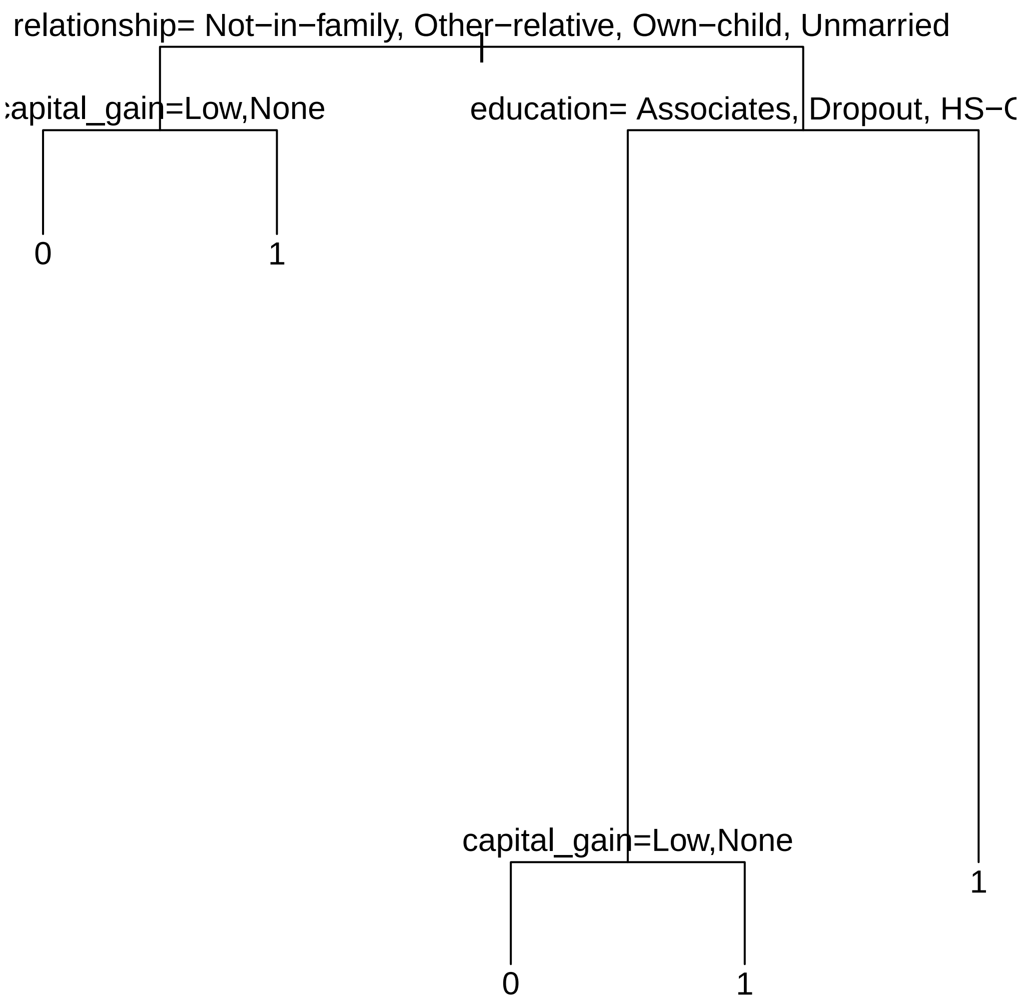
```
## hours_per_week 0.102033       1.000000
```

```
PCAtrain <- prcomp(train[, c(2,5)])
ggplot(aes(x=PC1, y=PC2), data = data.frame(PCAtrain$x) ) + geom_point(alpha=.15, col='blue')
```

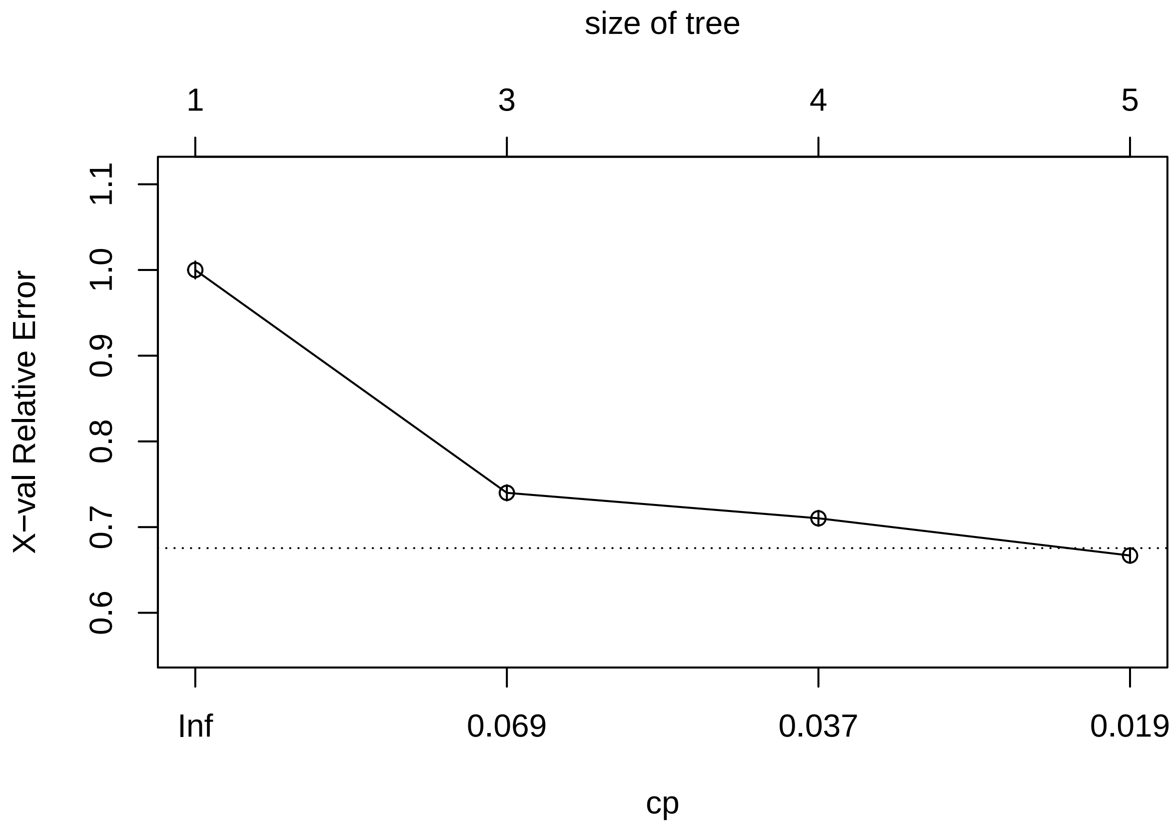


Build a Classification Tree

```
#Fit a classification tree
Ctree <- rpart(income ~ ., data=train, method='class')
plot(Ctree)
text(Ctree, pretty=0)
```




```
#Make plots and describe the steps you took to justify choosing optimal tuning parameters.
plotcp(Ctree)
```



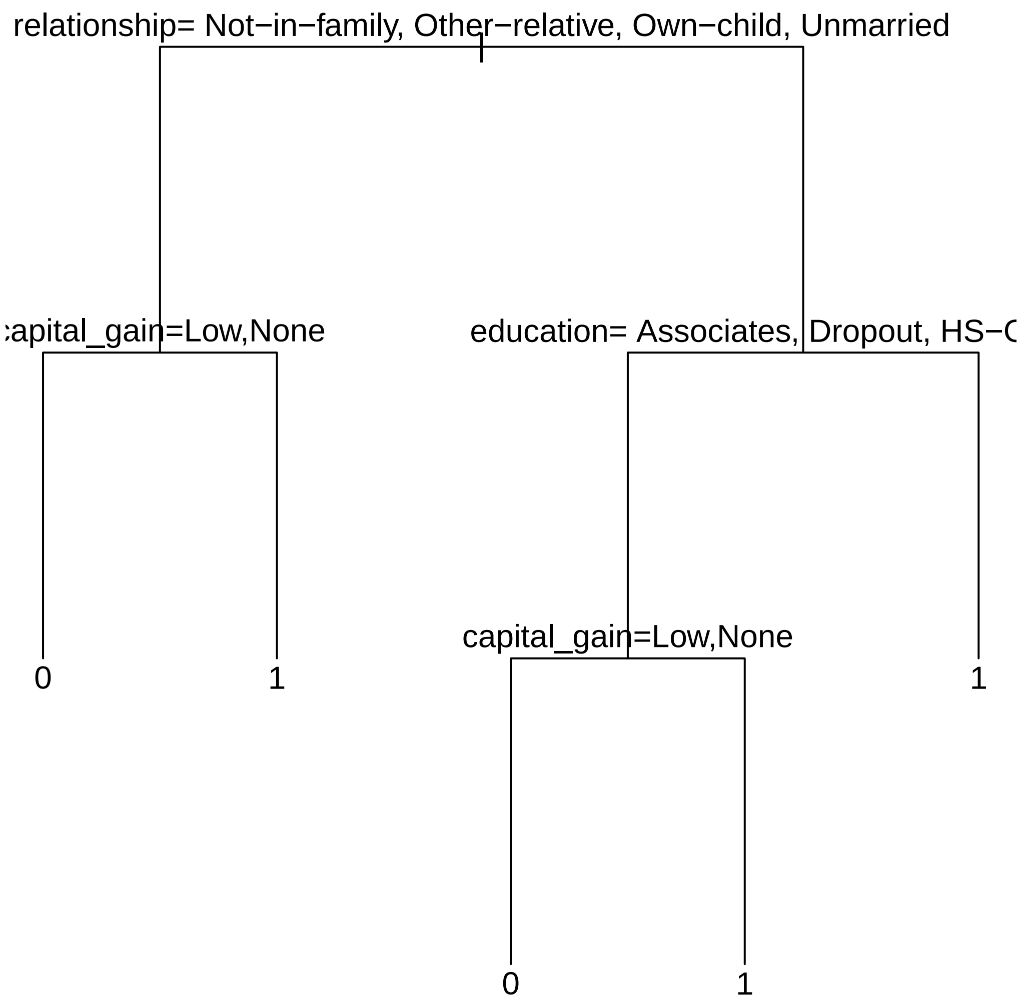

```
printcp(Ctree)
```

```
##
## Classification tree:
## rpart(formula = income ~ ., data = train, method = "class")
##
## Variables actually used in tree construction:
## [1] capital_gain education    relationship
##
## Root node error: 7507/30155 = 0.24895
##
## n= 30155
##
##         CP nsplit rel error  xerror      xstd
## 1 0.130012      0   1.00000 1.00000 0.0100024
## 2 0.036899      2   0.73998 0.73998 0.0089673
## 3 0.036233      3   0.70308 0.71027 0.0088252
## 4 0.010000      4   0.66684 0.66684 0.0086072
```

```
prunedCtree <- prune(Ctree, cp = Ctree$cptable[which.min(Ctree$cptable[,"xerror"]),"CP"])
```

I chose to cut the tree off based on the minimum cross validated error **Note: some of the code above finding the min using the which function was found online

```
plot(prunedCtree,uniform=TRUE)
text(prunedCtree, pretty=0)
```



The tree wasn't actually pruned, since the lowest cross validation error happened when the tree had all of its leaves.

```r
#Report your 5 (or 6 or 7) important features (could be either just 5, or 6 or 7), with their variable
pruned.tree.sum <- summary(prunedCtree)
```

```
## Call:
## rpart(formula = income ~ ., data = train, method = "class")
##   n= 30155
##
##           CP nsplit rel error    xerror        xstd
## 1 0.13001199      0 1.0000000 1.0000000 0.010002350
## 2 0.03689889      2 0.7399760 0.7399760 0.008967337
## 3 0.03623285      3 0.7030771 0.7102704 0.008825232
## 4 0.01000000      4 0.6668443 0.6668443 0.008607157
##
## Variable importance
##   relationship marital.status      education   capital_gain            sex
##             27             27             11              9              9
##     occupation            age hours_per_week
##              7              6              4
##
## Node number 1: 30155 observations,    complexity param=0.130012
##   predicted class=0  expected loss=0.2489471  P(node) =1
```

```
##     class counts: 22648  7507
##    probabilities: 0.751 0.249
##   left son=2 (16286 obs) right son=3 (13869 obs)
##   Primary splits:
##       relationship   splits as  RLLLLR,  improve=2277.196, (0 missing)
##       marital.status splits as  RLLL,    improve=2244.433, (0 missing)
##       capital_gain   splits as  RLL,     improve=1220.325, (0 missing)
##       education      splits as  LRRLLRR, improve=1192.818, (0 missing)
##       occupation     splits as  LLRRLL,  improve=1038.684, (0 missing)
##   Surrogate splits:
##       marital.status splits as  RLLL,     agree=0.993, adj=0.984, (0 split)
##       sex            splits as  LR,       agree=0.691, adj=0.328, (0 split)
##       age            < 33.5 to the left,  agree=0.645, adj=0.229, (0 split)
##       hours_per_week < 43.5 to the left,  agree=0.604, adj=0.138, (0 split)
##       occupation     splits as  LRRRLL,   agree=0.600, adj=0.129, (0 split)
##
## Node number 2: 16286 observations,    complexity param=0.03689889
##   predicted class=0  expected loss=0.06963036  P(node) =0.5400763
##     class counts: 15152  1134
##    probabilities: 0.930 0.070
##   left son=4 (15989 obs) right son=5 (297 obs)
##   Primary splits:
##       capital_gain   splits as  RLL,      improve=486.48960, (0 missing)
##       education      splits as  LLRLLRR,  improve=142.13350, (0 missing)
##       occupation     splits as  LLRRLL,   improve=116.27470, (0 missing)
##       hours_per_week < 42.5 to the left,  improve=108.13670, (0 missing)
##       age            < 28.5 to the left,  improve= 69.90175, (0 missing)
##
## Node number 3: 13869 observations,    complexity param=0.130012
##   predicted class=0  expected loss=0.459514  P(node) =0.4599237
##     class counts:  7496  6373
##    probabilities: 0.540 0.460
##   left son=6 (9719 obs) right son=7 (4150 obs)
##   Primary splits:
##       education      splits as  LRRLLRR,  improve=900.0575, (0 missing)
##       occupation     splits as  LLRRRL,   improve=765.5709, (0 missing)
##       capital_gain   splits as  RLL,      improve=473.2348, (0 missing)
##       age            < 33.5 to the left,  improve=218.4852, (0 missing)
##       hours_per_week < 41.5 to the left,  improve=186.4766, (0 missing)
##   Surrogate splits:
##       occupation     splits as  LLRRLL,      agree=0.792, adj=0.306, (0 split)
##       capital_gain   splits as  RLL,         agree=0.717, adj=0.054, (0 split)
##       capital_loss   splits as  RLL,         agree=0.706, adj=0.018, (0 split)
##       native.country splits as  RRRRLLLLLLL, agree=0.706, adj=0.017, (0 split)
##       race           splits as  LRLLL,       agree=0.703, adj=0.006, (0 split)
##
## Node number 4: 15989 observations
##   predicted class=0  expected loss=0.05297392  P(node) =0.5302272
##     class counts: 15142   847
##    probabilities: 0.947 0.053
##
## Node number 5: 297 observations
##   predicted class=1  expected loss=0.03367003  P(node) =0.009849113
##     class counts:    10   287
```

```
##    probabilities: 0.034 0.966
##
## Node number 6: 9719 observations,    complexity param=0.03623285
##   predicted class=0  expected loss=0.3418047  P(node) =0.3223014
##     class counts:  6397  3322
##    probabilities: 0.658 0.342
##   left son=12 (9435 obs) right son=13 (284 obs)
##   Primary splits:
##       capital_gain   splits as  RLL,       improve=237.46540, (0 missing)
##       occupation     splits as  RLRRRL,    improve=172.67000, (0 missing)
##       education      splits as  R--LR--,   improve=164.36250, (0 missing)
##       age            < 35.5 to the left,   improve=131.17600, (0 missing)
##       hours_per_week < 41.5 to the left,   improve= 56.57807, (0 missing)
##
## Node number 7: 4150 observations
##   predicted class=1  expected loss=0.2648193  P(node) =0.1376223
##     class counts:  1099  3051
##    probabilities: 0.265 0.735
##
## Node number 12: 9435 observations
##   predicted class=0  expected loss=0.3226285  P(node) =0.3128834
##     class counts:  6391  3044
##    probabilities: 0.677 0.323
##
## Node number 13: 284 observations
##   predicted class=1  expected loss=0.02112676  P(node) =0.009418007
##     class counts:     6   278
##    probabilities: 0.021 0.979
```

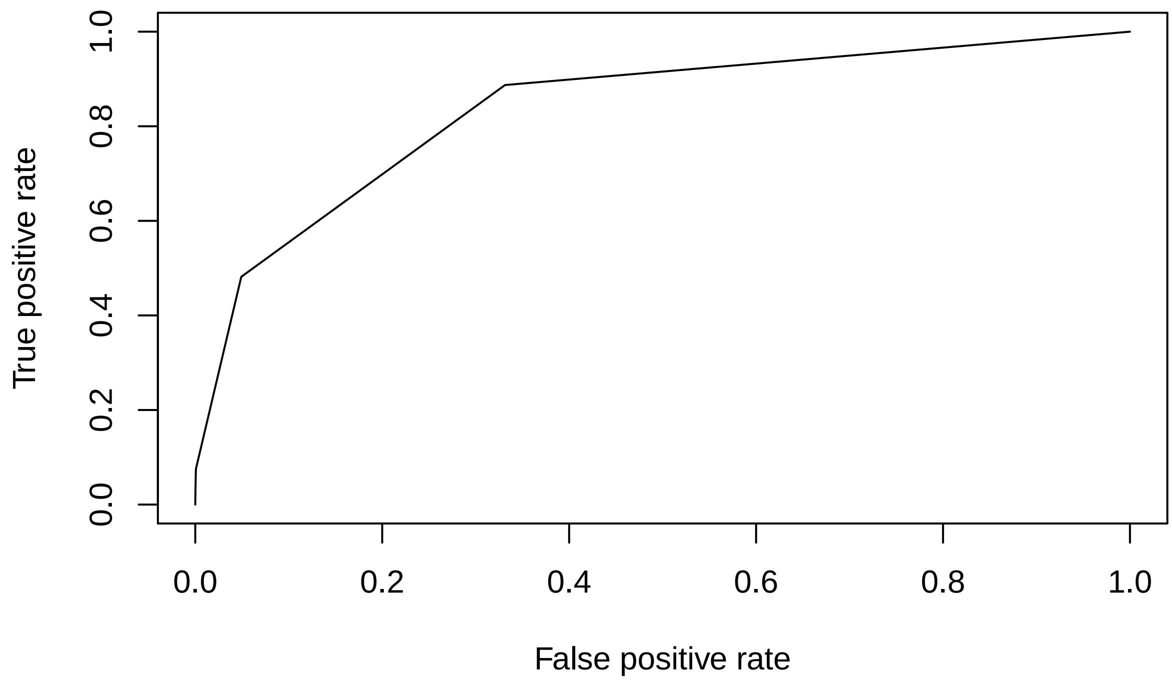
```
var_imp_df <- data.frame(pruned.tree.sum$variable.importance)
top_seven <- data.frame('Predictor' = rownames(var_imp_df)[c(1:5)], 'Variable_Importance' = var_imp_df[c
top_seven
```

```
##        Predictor Variable_Importance
## 1   relationship           2277.1955
## 2 marital.status           2241.5656
## 3      education            900.0575
## 4   capital_gain            772.7533
## 5            sex            746.5865
```

```
#Report the training accuracy rate
acc <- data.frame(predict(prunedCtree))
acc['X1'] <- as.integer(acc['X1'] > .5)
acc['class'] <- train['income']
acc['correct'] <- as.integer(acc['class'] == acc['X1'])
Ctree_accuracy <- sum(acc['correct'])/nrow(acc)
Ctree_accuracy
```

```
## [1] 0.833991
```

```
#Plot the ROC curve, and report its area under the curve (AUC) statistic.
pred <- prediction(predict(prunedCtree, type = "prob")[, 2], train$income)
plot(performance(pred, 'tpr', 'fpr'))
```




```
as.numeric(performance(pred, 'auc')@y.values)
```

```
## [1] 0.8375429
```

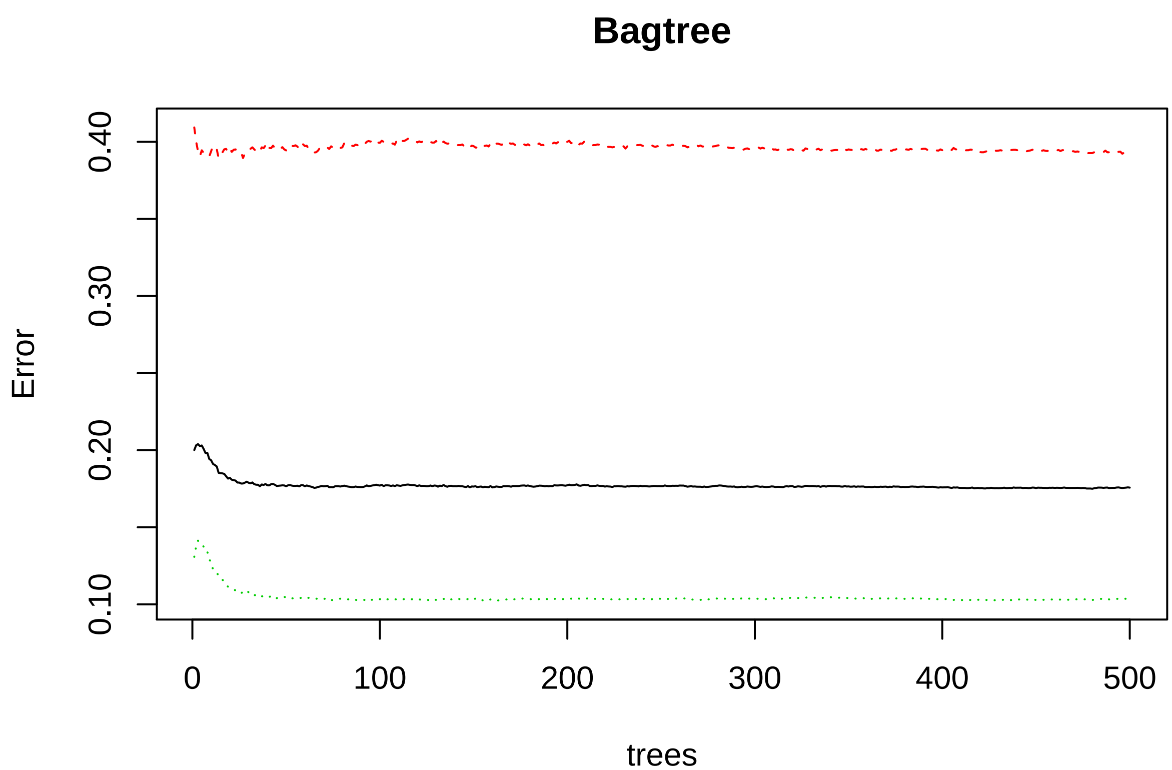
***Note: Instruction on how to use ROC found online

Build a Bagged Tree

```
#Train a Random Forest classifier
trainF <- train
incomeF <- ifelse(train$income == 0, "Low", "High")
trainF <- data.frame(train[-1], incomeF)
Bagtree <- randomForest(factor(incomeF) ~., data=trainF, mtry=12, importance=TRUE)
Bagtree
```

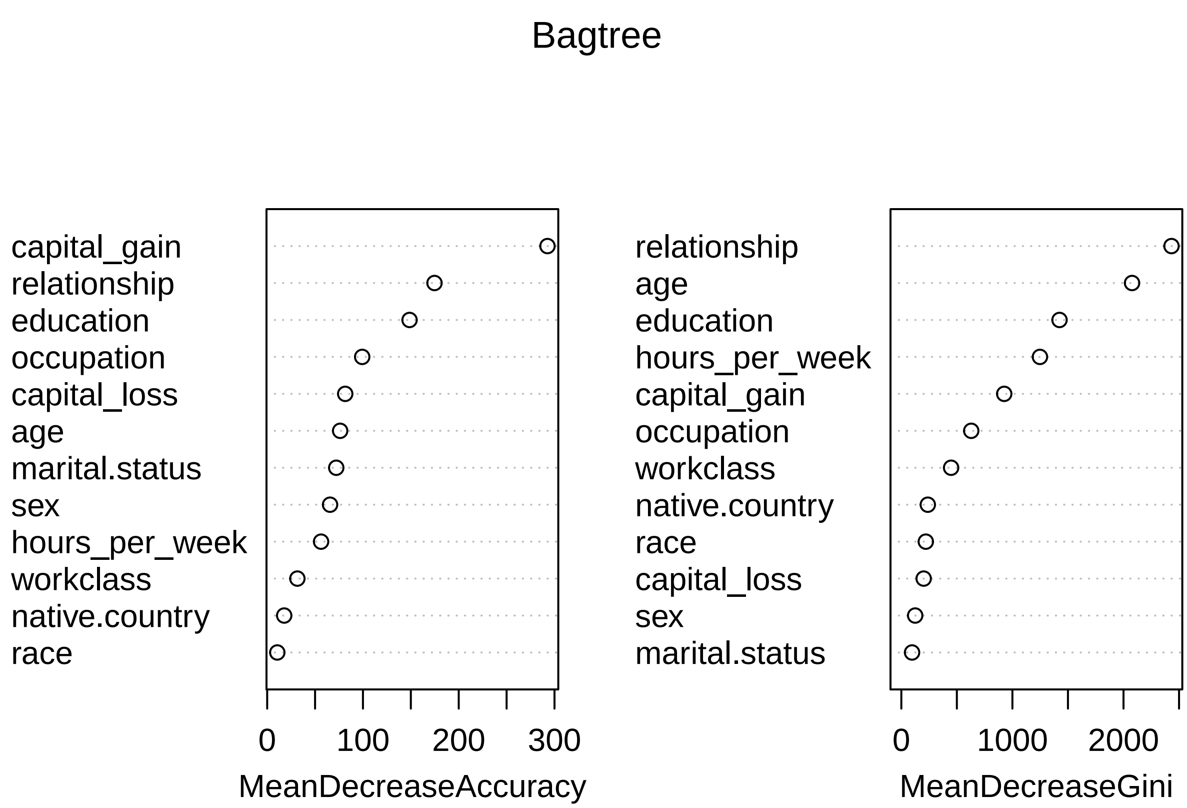
```
##
## Call:
##  randomForest(formula = factor(incomeF) ~ ., data = trainF, mtry = 12,      importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 12
##
##         OOB estimate of  error rate: 17.57%
## Confusion matrix:
##      High   Low class.error
## High 4554  2953   0.3933662
## Low  2346 20302   0.1035853
```

```
#Make plots and describe the steps you took to justify choosing optimal tuning parameters.
plot(Bagtree)
```



Not tuned. Error is lowest with max trees

```r
#Report your 5 (or 6 or 7) important features (could be either just 5, or 6 or 7), with their variable
varImpPlot(Bagtree)
```



The most important features are: * Capital gain * Relationship * Age * Education * Occupation * Hours Per

Week * Capital loss

```r
Bagimp_df <- data.frame(importance(Bagtree))
Bagimp_df <- Bagimp_df[order(-Bagimp_df$MeanDecreaseAccuracy),]

Bagimp_df <- data.frame(rownames(Bagimp_df)[c(1:7)], Bagimp_df$MeanDecreaseAccuracy[c(1:7)])
names(Bagimp_df) <- c('Variable', 'Mean Decrease in Accuracy')
Bagimp_df
```

```
##         Variable Mean Decrease in Accuracy
## 1   capital_gain                 292.67046
## 2   relationship                 174.70226
## 3      education                 148.65745
## 4     occupation                  99.19384
## 5   capital_loss                  81.45669
## 6            age                  76.16911
## 7 marital.status                  72.11716
```

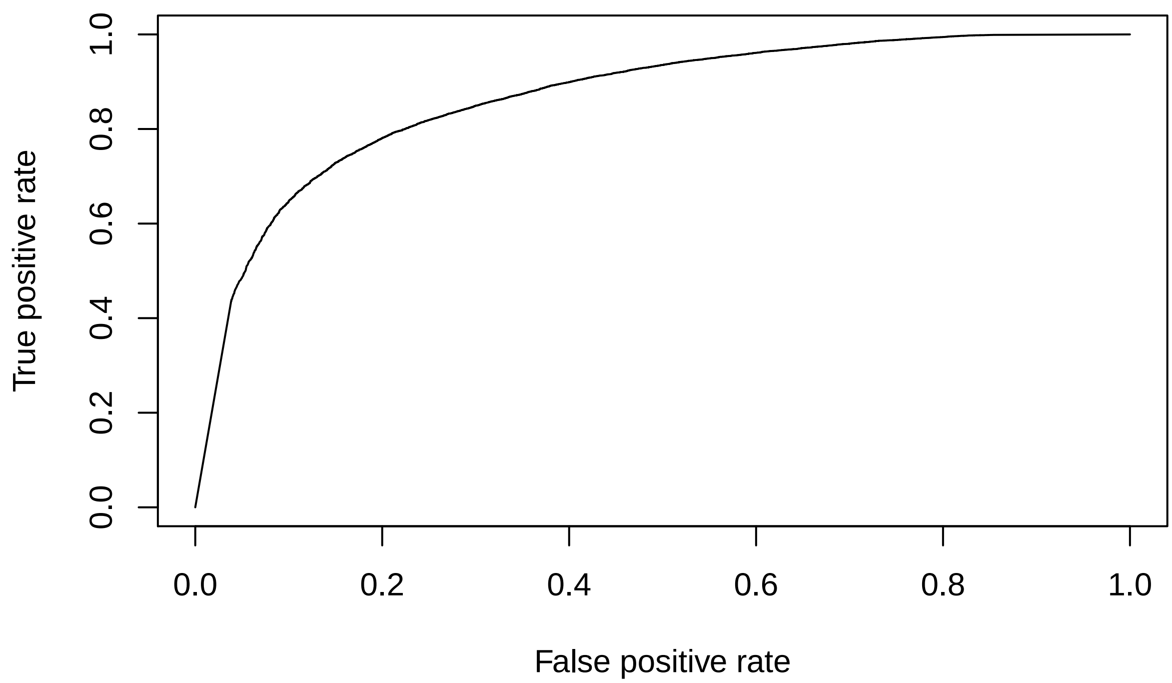
```r
#Report the training accuracy rate
sum(diag(Bagtree$confusion))/nrow(train)
```

```
## [1] 0.8242746
```

```r
#Plot the ROC curve, and report its area under the curve (AUC) statistic.
Bagpred <- prediction(predict(Bagtree, type = "prob")[, 2], trainF$incomeF)
performance(Bagpred, "auc")@y.values[[1]]
```

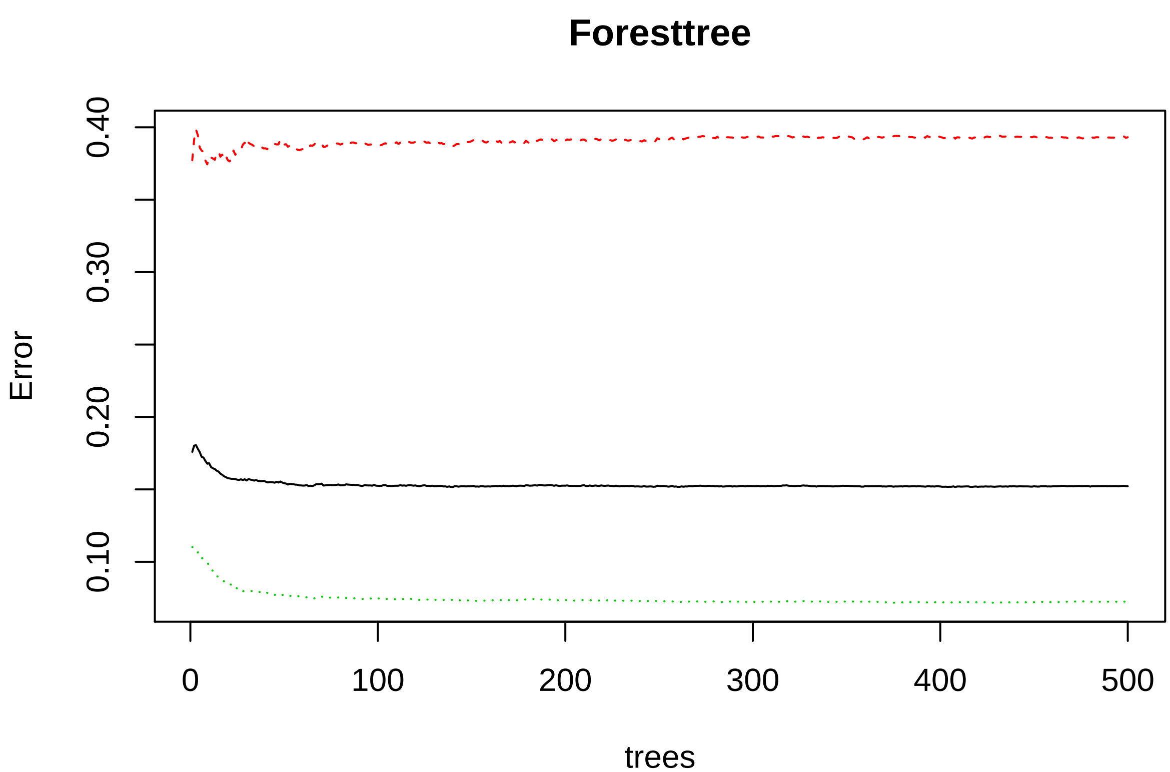
```
## [1] 0.8664682
```

```r
plot(performance(Bagpred, "tpr", "fpr"))
```




```r
#Train a Random Forest classifier (see examples in ISL chapter 8, and APM chapter 14)
Foresttree <- randomForest(factor(incomeF) ~., data=trainF, importance=TRUE)
```

```r
#Make plots and describe the steps you took to justify choosing optimal tuning parameters.
plot(Foresttree)
```



Not tuned. Error is lowest with max trees.

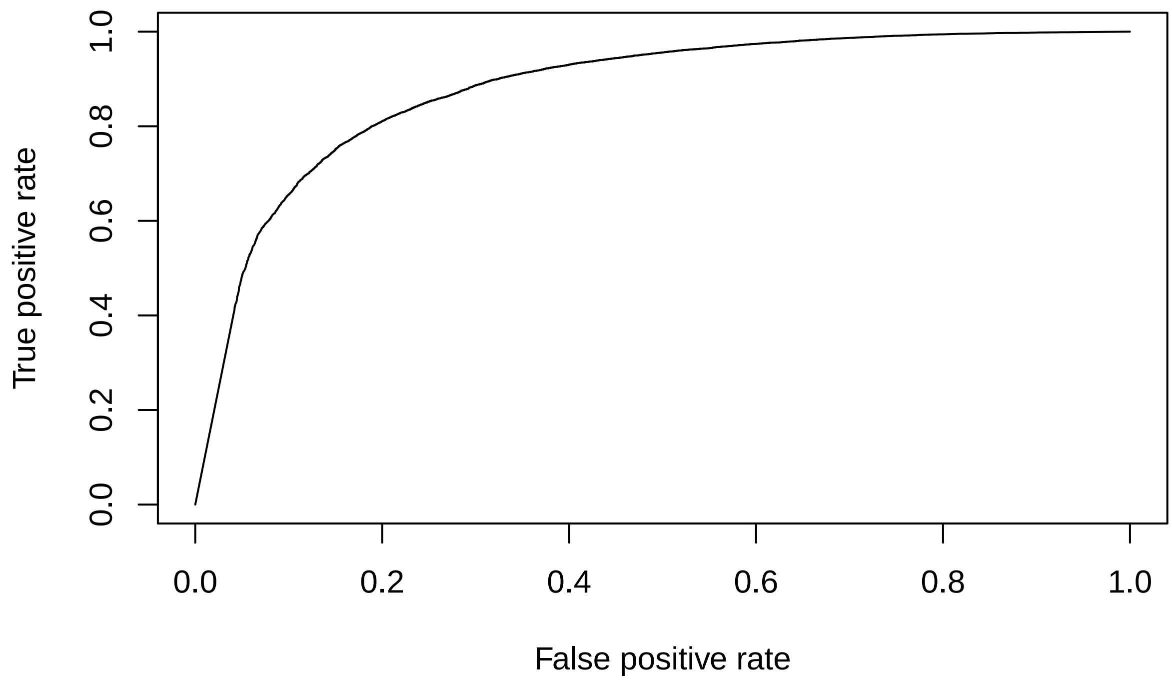
```r
#Report your 5 (or 6 or 7) important features (could be either just 5, or 6 or 7), with their variable
Forestimp_df <- data.frame(importance(Foresttree))
Forestimp_df <- Forestimp_df[order(-Forestimp_df$MeanDecreaseAccuracy),]

Forestimp_df <- data.frame(rownames(Forestimp_df)[c(1:7)], Forestimp_df$MeanDecreaseAccuracy[c(1:7)])
names(Forestimp_df) <- c('Variable', 'Mean Decrease in Accuracy')
Forestimp_df
```

```
##          Variable Mean Decrease in Accuracy
## 1    capital_gain                 230.98612
## 2       education                 109.14471
## 3      occupation                  93.49707
## 4             age                  85.63432
## 5    capital_loss                  68.61806
## 6  hours_per_week                  61.38844
## 7  marital.status                  49.24326
```

```r
#Report the training accuracy rate
sum(diag(Foresttree$confusion))/nrow(train)
```

```
## [1] 0.8478196
```

```r
#Plot the ROC curve, and report its area under the curve (AUC) statistic.
Forestpred <- prediction(predict(Foresttree, type = "prob")[, 2], trainF$incomeF)
plot(performance(Forestpred, 'tpr', 'fpr'))
```




```
as.numeric(performance(Forestpred, 'auc')@y.values)
```

```
## [1] 0.8797547
```

Model Selection

```
#Validate your best supervised classifier on the test set.
#Foresttree had the best accuracy and AUC

#Included to use for bagging and RF, which had income as factored characters
incomeF <- ifelse(test$income == 0, "Low", "High")
testF <- data.frame(test[-1], incomeF)
#testF <- testF[-8578,]
#testF <- droplevels(testF)
```

```
sum(as.character(predict(Foresttree, testF)) == testF$incomeF)/nrow(testF)
```

```
## [1] 0.7913679
```

It's a pretty good accuracy :)

```
#Compute the confusion matrix

Confusionize <- function(predictions, truevals) {
  return(table(data.frame('Prediction' = as.character(predictions), 'True' = truevals)))
}
#Random Forest Confusion Matrix
ForestConf <- Confusionize(predict(Foresttree, testF), testF$incomeF)

ForestConf
```

```
##             True
```

```
## Prediction  High   Low
##       High   572    14
##       Low   3128 11346
```

```r
#Bagging Confusion Matrix
Bagpredictions <- predict(Bagtree, testF)
BagConf <- Confusionize(Bagpredictions, testF$incomeF)
BagConf
```

```
##            True
## Prediction  High   Low
##       High   501    39
##       Low   3199 11321
```

```r
#Tree confusion matrix
tree_probs <- data.frame(predict(prunedCtree, test))
tree_probs['prediction'] <- as.integer(tree_probs['X1'] > .5)
tree_probs['true'] <- test$income
treeConf <- table(tree_probs[, c(3,4)])
treeConf
```

```
##            true
## prediction     0     1
##          0 10773  1944
##          1   587  1756
```

In this case 0 is Low and 1 is High (if I renamed the columns, I wasn't sure how to keep the "prediction" and "true" labels)

```r
#Using the class "over 50K a year" as the positive event, calculate the Sensitivity or True Positive Ra

#True Positive: Random Forests
ForestConf[1,1]/sum(ForestConf[,1])
```

```
## [1] 0.1545946
```

```r
#True Positive: Bagging
BagConf[1,1]/sum(BagConf[,1])
```

```
## [1] 0.1354054
```

```r
#Ture Positive: Normal Tree
treeConf[2,2]/sum(treeConf[,2])
```

```
## [1] 0.4745946
```

```r
#True Negative Rate: Random Forests
ForestConf[2,2]/sum(ForestConf[,2])
```

```
## [1] 0.9987676
```

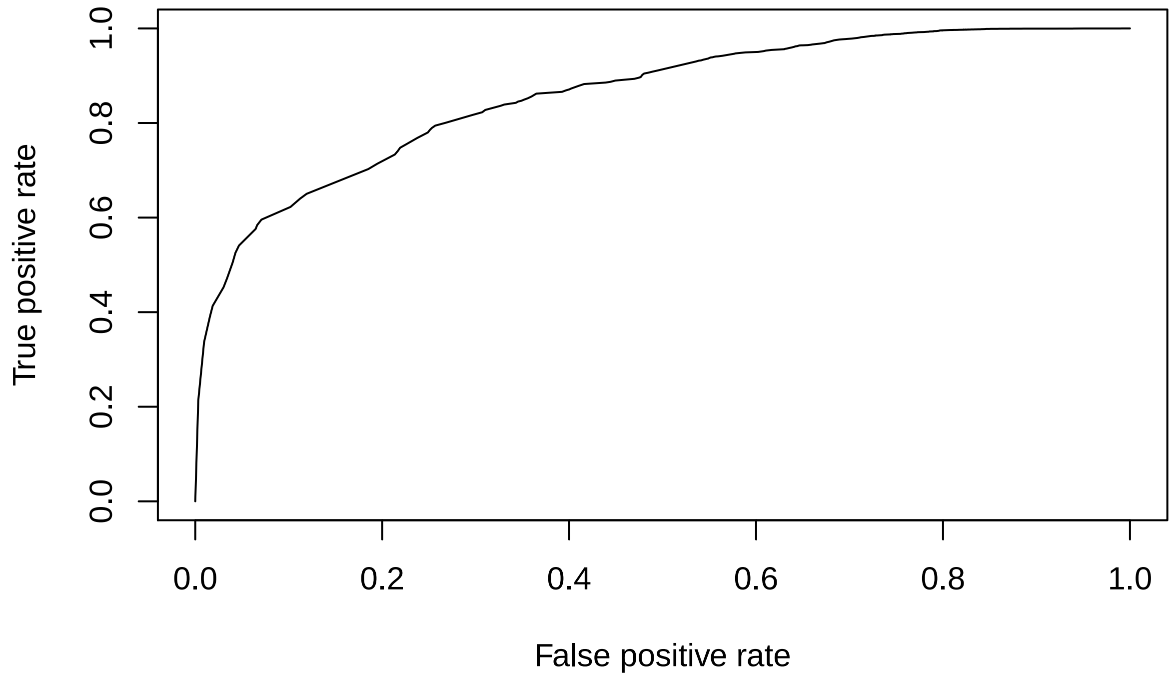
```r
#True Negative Rate: Bagging
BagConf[2,2]/sum(BagConf[,2])
```

```
## [1] 0.9965669
```

```r
#True Negative Rate: Normal Tree
treeConf[1,1]/sum(treeConf[,1])
```

```
## [1] 0.9483275
```

```r
#ROC Curves: Forests
Forestpred <- prediction(predict(Foresttree, testF, type = "prob")[, 2], testF$incomeF)
plot(performance(Forestpred, 'tpr', 'fpr'))
```



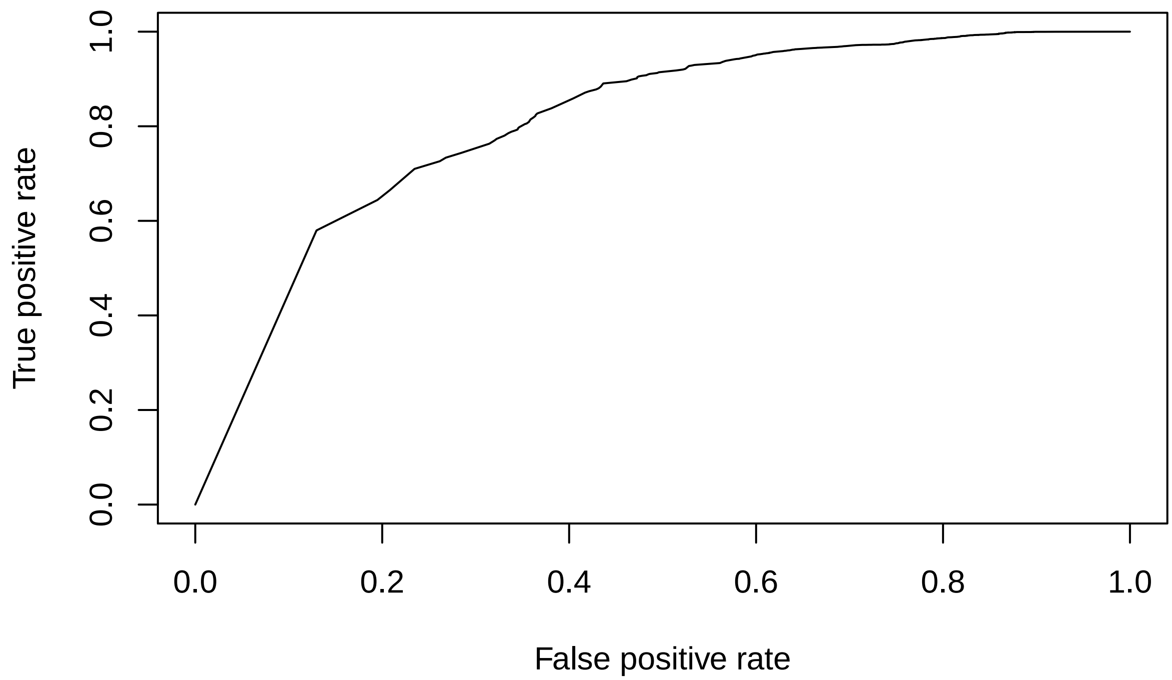

```r
as.numeric(performance(Forestpred, 'auc')@y.values)
```

```
## [1] 0.8577164
```

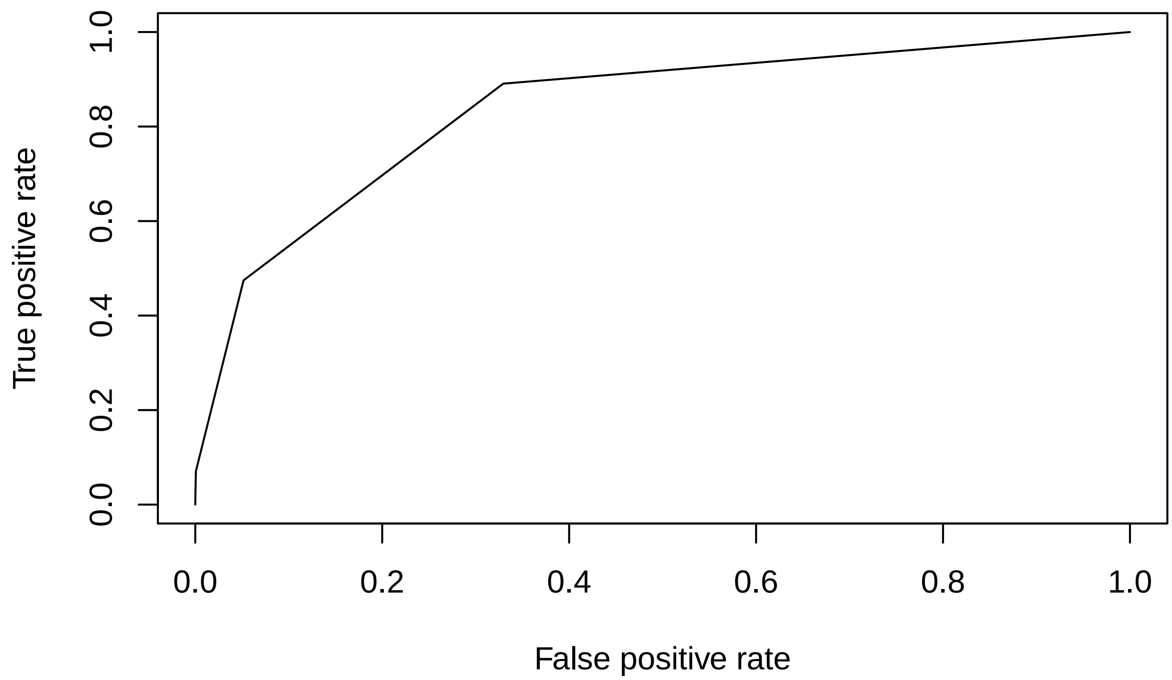
```r
#ROC curves: Bagging
Bagpred <- prediction(predict(Bagtree, testF, type = "prob")[, 2], testF$incomeF)
performance(Bagpred, "auc")@y.values[[1]]
```

```
## [1] 0.8081377
```

```r
plot(performance(Bagpred, "tpr", "fpr"))
```




```r
#ROC curves: normal tree
pred <- prediction(predict(prunedCtree, test, type = "prob")[, 2], test$income)
plot(performance(pred, 'tpr', 'fpr'))
```




```r
as.numeric(performance(pred, 'auc')@y.values)
```

```
## [1] 0.8374965
```