Homework 5

Charles Hwang

Professor Matthews

STAT 388-001

12 November 2019

Exercise 1

```
rm(list=ls())
train <- read.csv(file="/Users/newuser/Desktop/Notes/Undergraduate/STAT 338 - Predictive Analytics/titar
test <- read.csv(file="/Users/newuser/Desktop/Notes/Undergraduate/STAT 338 - Predictive Analytics/titani
train <- train[,-c(4,9,11)] # Removing categorical variables with large number of unique values ("Name"
test <- test[,-c(3,8,10)]
train$Sex <- as.factor(train$Sex)
test$Sex <- as.factor(test$Sex)
train$Embarked <- as.factor(train$Embarked)
test$Embarked <- as.factor(test$Embarked)
library(car)
library(gbm)
library(glmnet)
library(ISLR)
library(plotmo)
library(randomForest)
library(rpart)
set.seed(811)
```

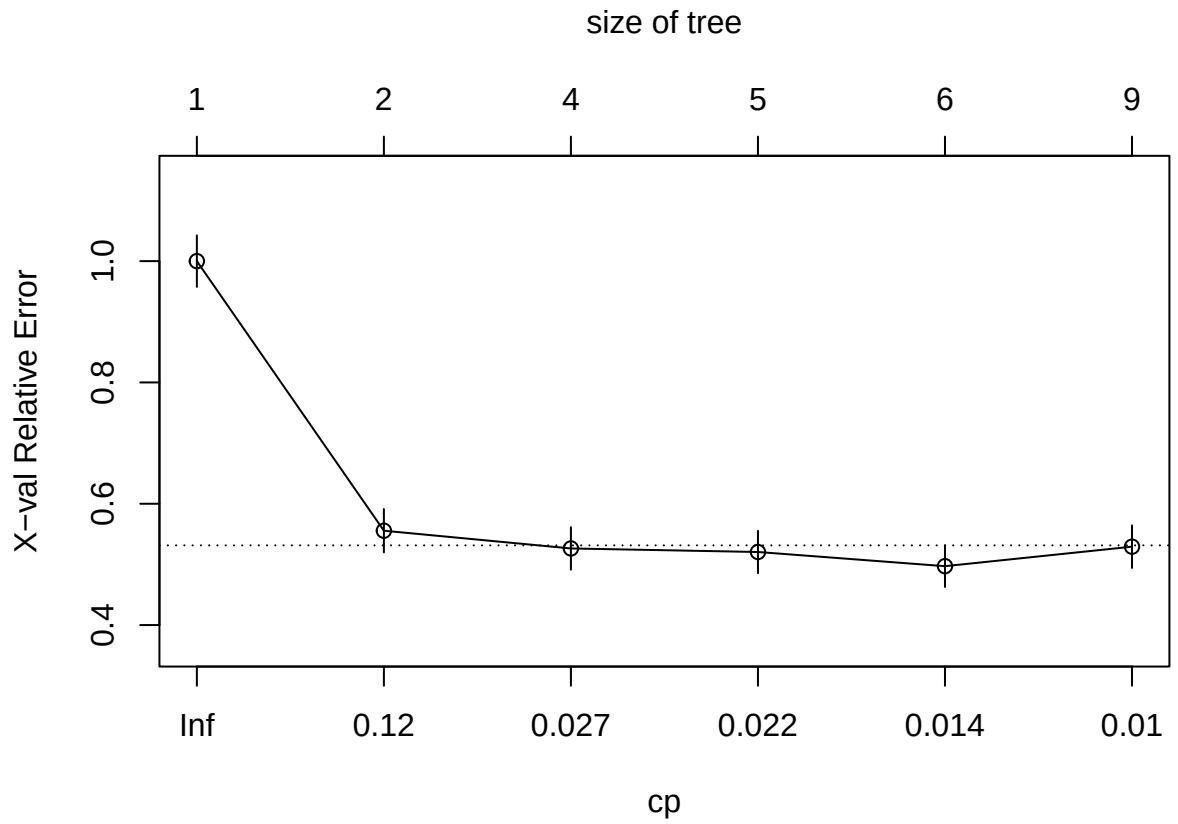
Exercise 2

```
tree <- rpart(Survived~.,method="class",data=train)
printcp(tree)
```

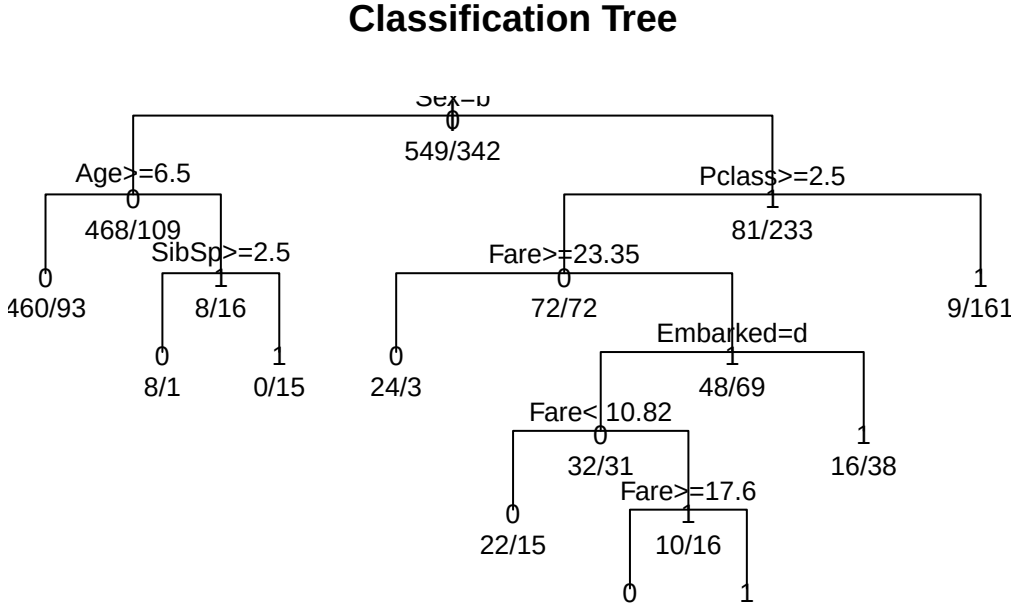
```
##
## Classification tree:
## rpart(formula = Survived ~ ., data = train, method = "class")
##
## Variables actually used in tree construction:
## [1] Age      Embarked Fare     Pclass   Sex      SibSp
##
## Root node error: 342/891 = 0.38384
##
## n= 891
##
##         CP nsplit rel error  xerror     xstd
## 1 0.444444      0   1.00000 1.00000 0.042446
## 2 0.030702      1   0.55556 0.55556 0.035750
```

```
## 3 0.023392      3   0.49415 0.52632 0.035043
## 4 0.020468      4   0.47076 0.52047 0.034897
## 5 0.010234      5   0.45029 0.49708 0.034295
## 6 0.010000      8   0.41813 0.52924 0.035116
```

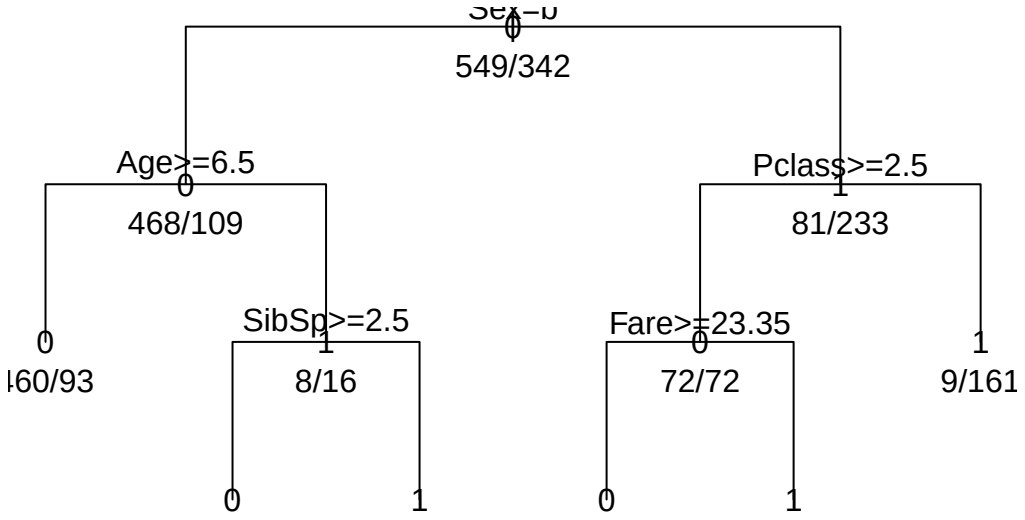
```
plotcp(tree)
```




```
plot(tree,uniform=TRUE,main="Classification Tree")
text(tree,use.n=TRUE,all=TRUE,cex=.8)
```



```
ptree <- prune(tree,cp=tree$cptable[which.min(tree$cptable[,"xerror"]),"CP"])
plot(ptree,uniform=TRUE,main="Pruned Classification Tree")
text(ptree,use.n=TRUE,all=TRUE)
```




```
treepred <- predict(tree,test,type="class")
table(treepred)
```

```
## treepred
##   0   1
## 288 130
```

```
cat("Cross validation error:",min(tree$cptable[,"xerror"]))
```

```
## Cross validation error: 0.497076
```

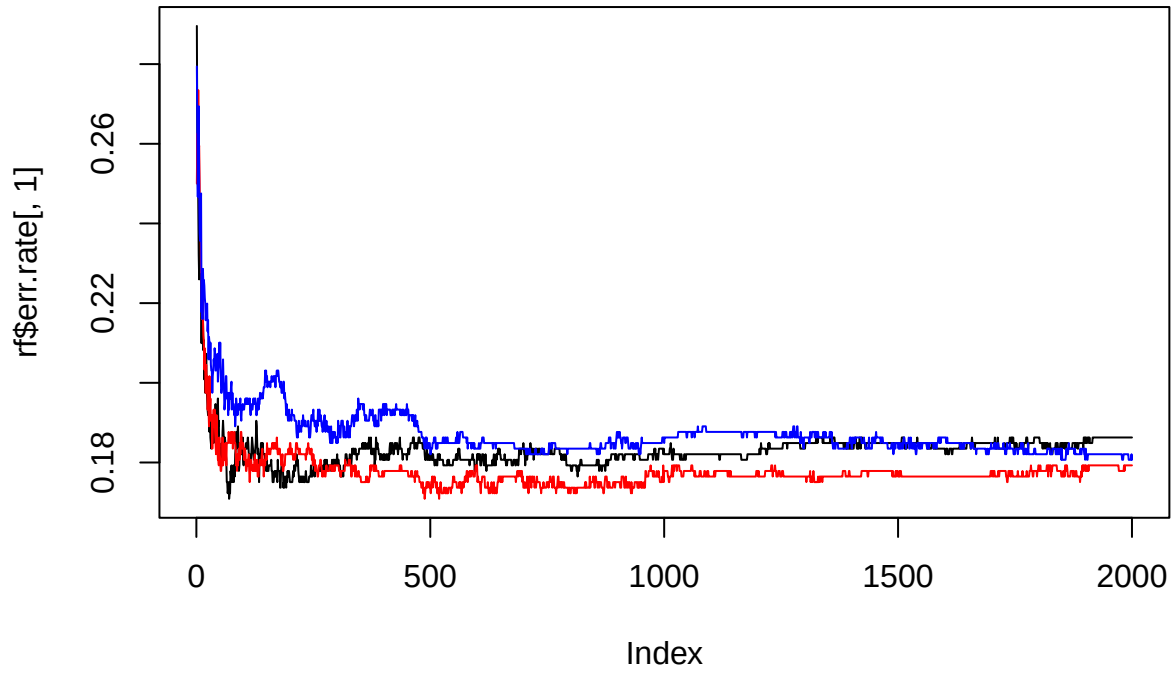
Exercise 3

```
rf <- randomForest(as.factor(Survived)~.,data=train[!is.na(train$Age),],ntree=2000,importance=TRUE)
rf
```

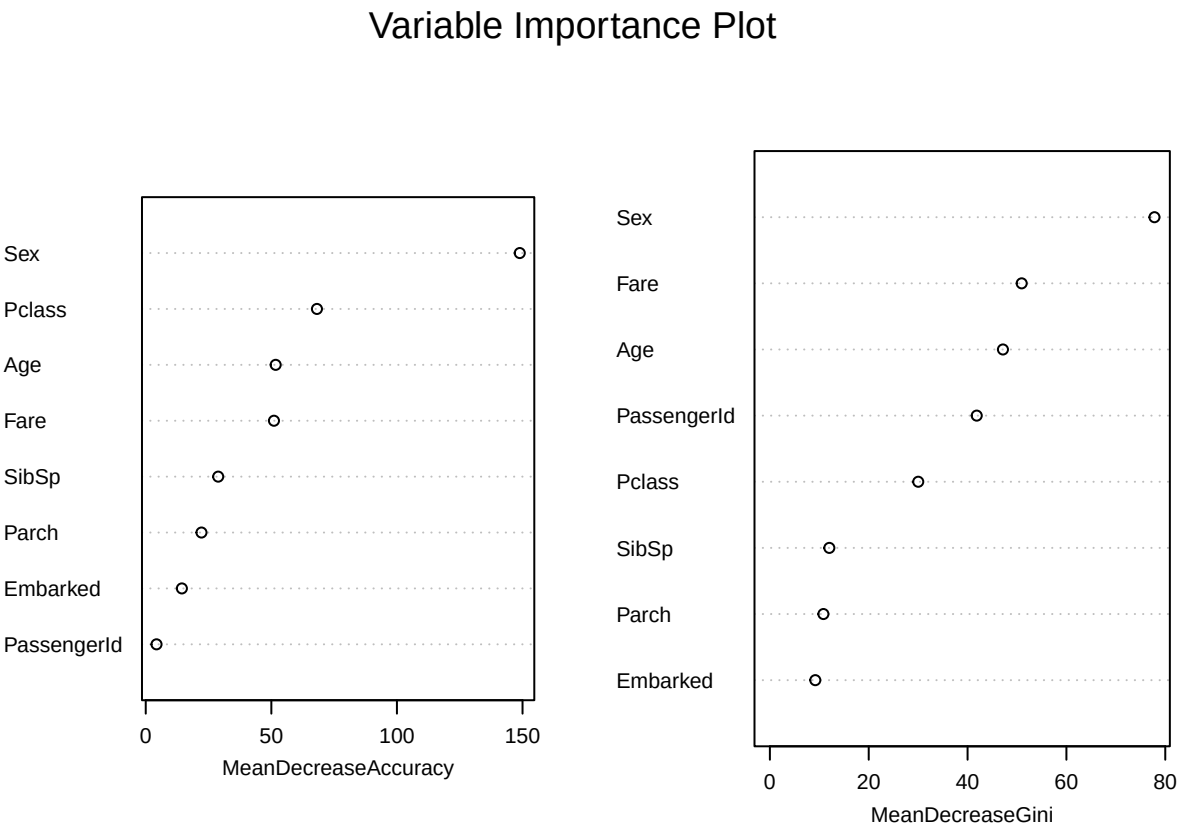
```
##
## Call:
##  randomForest(formula = as.factor(Survived) ~ ., data = train[!is.na(train$Age),      ], ntree = 2000
##                Type of random forest: classification
##                      Number of trees: 2000
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 18.63%
## Confusion matrix:
##     0   1 class.error
## 0 382  42   0.0990566
## 1  91 199   0.3137931
```

```
# The out-of-bag cross-validation error is 17.79%.
rf3 <- randomForest(as.factor(Survived)~.,data=train[!is.na(train$Age),],ntree=2000,importance=TRUE,mtry
rf6 <- randomForest(as.factor(Survived)~.,data=train[!is.na(train$Age),],ntree=2000,importance=TRUE,mtry
```

```
plot(rf$err.rate[,1],type="l")
points(1:2000,rf3$err.rate[,1],type="l",col="red")
points(1:2000,rf6$err.rate[,1],type="l",col="blue")
```




```
varImpPlot(rf,main="Variable Importance Plot",cex=.7)
```

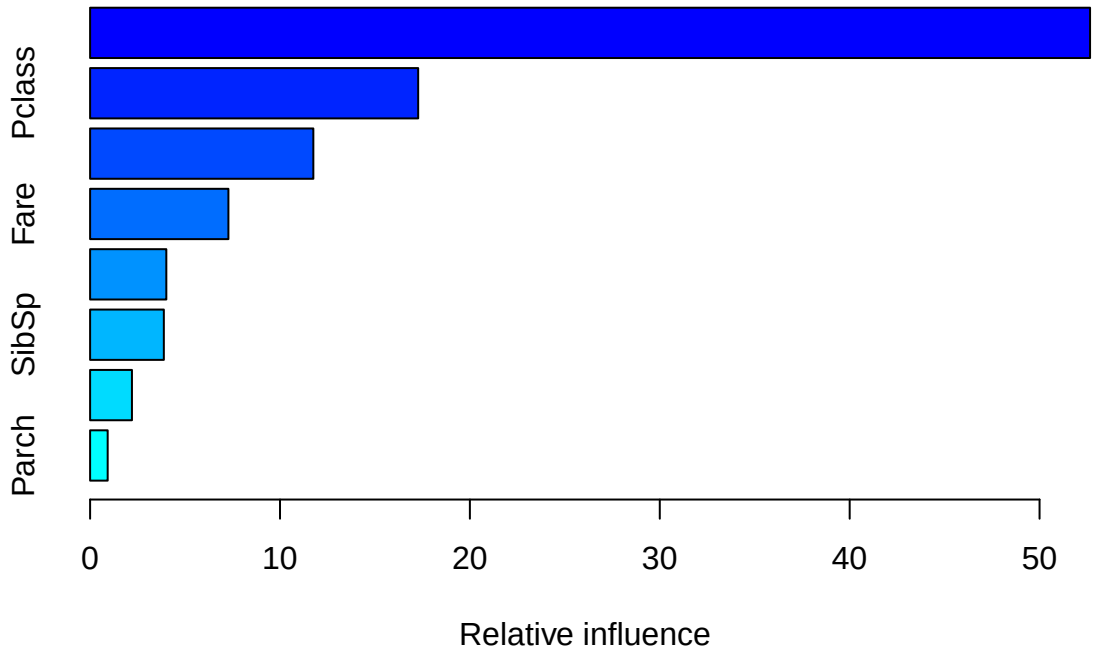


```
test <- rbind(train[1,-2],test) # See Stack Overflow for more information: https://stackoverflow.com/a/
test <- test[-1,]
rfpred <- predict(rf,test,type="class")
table(rfpred)
```

```
## rfpred
##   0   1
## 212 119
```

```
summary(gbm(Survived~.,data=train[!is.na(train$Age),]))
```

```
## Distribution not specified, assuming bernoulli ...
```




```
##                    var    rel.inf
## Sex                Sex 52.6596638
## Pclass          Pclass 17.2761214
## Age                Age 11.7555255
## Fare              Fare  7.2836242
## PassengerId PassengerId  4.0143068
## SibSp            SibSp  3.8826045
## Embarked      Embarked  2.2028040
## Parch            Parch  0.9253498
```

```
rfpreddf <- as.data.frame(rep(0,length(rfpred))) # Exporting predictions to CSV file for Kaggle submiss
rfpreddf$PassengerId <- test$PassengerId
rfpreddf$Survived <- rfpred
rfpreddf$`rep(0, length(rfpred))` <- NULL
Export(as.data.frame(rfpreddf),"Charles Hwang Submission.csv")
```

```
## Loading required namespace: rio
```

```
# Kaggle score: 290/418 (69.37799043% - nice!)
```