Titanic Tree

The dataset /home/bigdata/7/titanic.csv contains information on the passengers of the titanic. The following facts are recorded for each passenger: Class, Gender, Age. Class indication which category of ticket they had bought. The last fact the dataset contains is whether or not the passenger survived. Approach the task as follows:

```
library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units
```

```
X <- read.csv('titanic.csv',colClasses=c("NULL",NA,NA,NA,"NULL"))
Y <- read.csv('titanic.csv',colClasses=c("NULL","NULL","NULL","NULL",NA))

## 75% of the sample size
smp_size <- floor(0.75 * nrow(X))

## set the seed to make your partition reproductible
set.seed(123)
train_ind <- sample(seq_len(nrow(X)), size = smp_size)
Xtrain <- X[train_ind, ]
Xtest <- X[-train_ind, ]

Ytrain <- Y[train_ind, ]
Ytest <- Y[-train_ind, ]
summary(Xtrain)
```

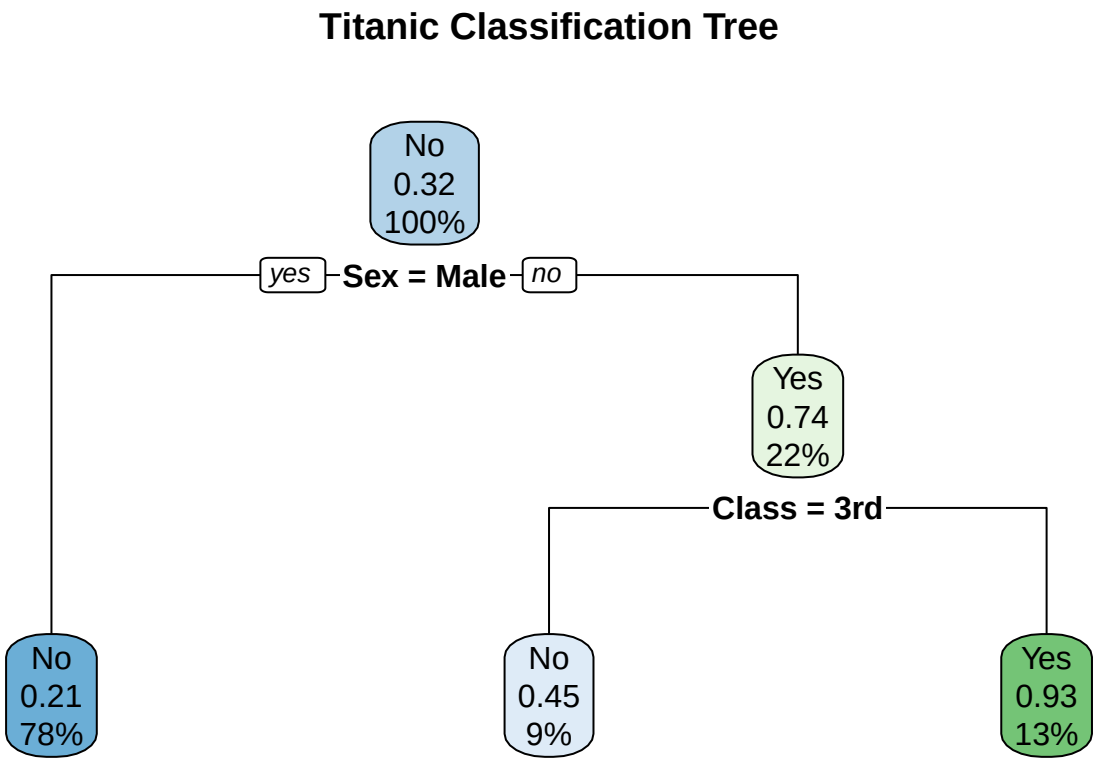
```
##   Class         Sex          Age
##  1st :248   Female: 355   Adult:1574
##  2nd :214   Male  :1295   Child:  76
##  3rd :533
##  Crew:655
```

Create a decision tree for all passengers and try to deduct useful rules for determining the survival or demise of a passenger.

```
#Classification (with two classes method="poisson" else "class")
library(rpart)
#plot tree
# install.packages('rpart.plot')
library(rpart.plot)

# has to be a dataframe to work with this package
Xtrain_df = as.data.frame(Xtrain)
```

```
# has to be a dataframe to work with this package
Xtest_df = as.data.frame(Xtest)
# Create a classification tree based on all inputs
tree = rpart(formula = Ytrain~., Xtrain_df, method="class")
rpart.plot(tree, main="Titanic Classification Tree")
```



It's apparent that the most dominant factors in survival is the gender followed by the Class. being a male in 3rd class have the highest death ratio.

```
#predictions
p = predict(tree, Xtest_df, type="class")
#calculate misclassification rate
1-sum(diag(table(Ytest,p)))/sum(table(Ytest,p))
```

```
## [1] 0.23049
```

```
# summarize accuracy
table(Ytest, p)
```

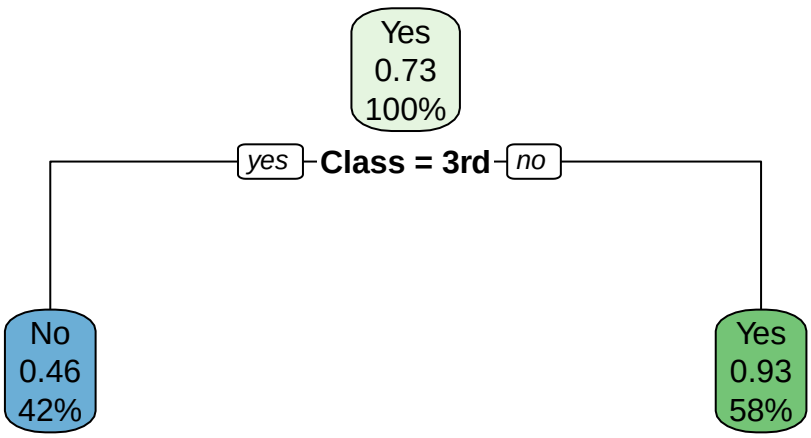
```
##       p
## Ytest  No Yes
##   No  368   6
##   Yes 121  56
```

The previous analysis could give the impression that during the chaos of a sinking ship people were well behaved and let himself die in favour of other. One factor that the classification tree doesn't take into consideration is the ratio of being a 3rd class. Below is a rerun of this same analysis but per Gender.

It shows a different point of view. Being female is apparently not enough, you also need to be a higher class person (pun intended). Also For males it also showed that age was a factor while it also shows that higher class adults were rescued over lower class childs. This would suggest that pretty harsh methods were used to prevent lower class people from reaching the lifeboats.

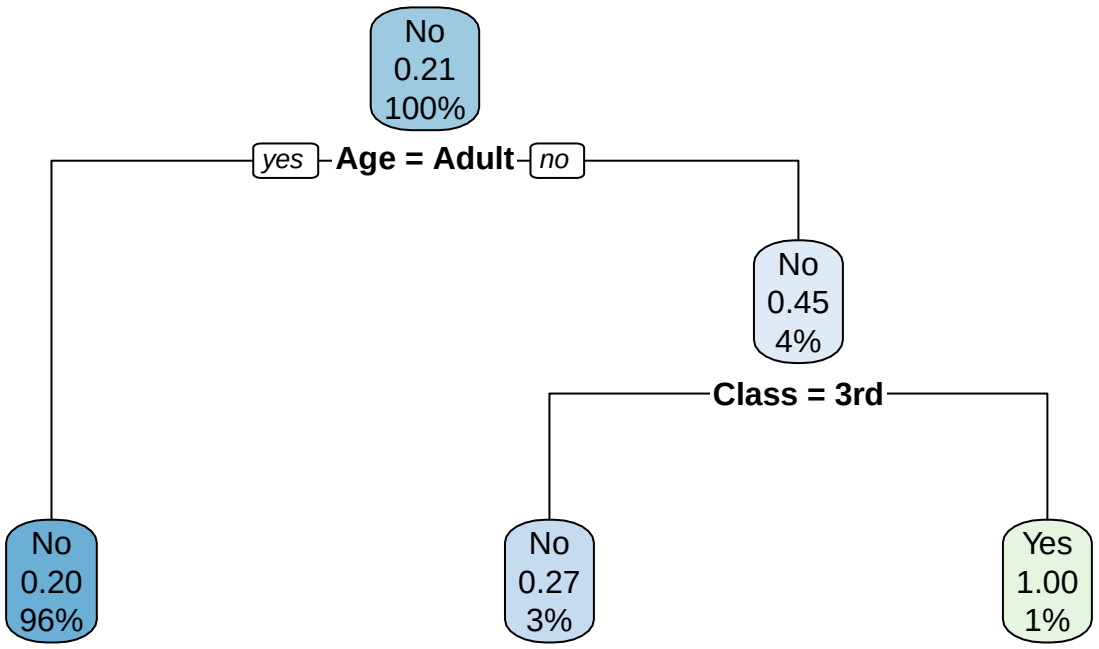
```
XFemales <- as.data.frame(read.csv('titanic.csv',colClasses=c("NULL",NA,NA,NA,"NULL")))
YFemales <- as.data.frame(read.csv('titanic.csv',colClasses=c("NULL","NULL",NA,"NULL",NA)))
XFemales <- XFemales[ which(XFemales$Sex == "Female"),]
YFemales <- YFemales[ which(YFemales$Sex == "Female"),]
drops <- c("Sex")
YFemales <- YFemales[ , !(names(YFemales) %in% drops)]
#YFemales <- subset(YFemales, select=-c(Sex))

tree = rpart(formula = YFemales~., XFemales, method="class")
rpart.plot(tree, main="Titanic Classification Tree Females")
```




```
XMales <- as.data.frame(read.csv('titanic.csv',colClasses=c("NULL",NA,NA,NA,"NULL")))
YMales <- as.data.frame(read.csv('titanic.csv',colClasses=c("NULL","NULL",NA,"NULL",NA)))
XMales <- XMales[ which(XMales$Sex == "Male"),]
YMales <- YMales[ which(YMales$Sex == "Male"),]
drops <- c("Sex")
YMales <- YMales[ , !(names(YMales) %in% drops)]
#YMales <- subset(YMales, select=-c(Sex))

tree = rpart(formula = YMales~., XMales, method="class")
rpart.plot(tree, main="Titanic Classification Tree Males")
```



• Execute k-fold-cross-validation to evaluate the accuracy of your decision tree1 • Compute the error rate of the predictions for k = (2, .., 10) and visualize false negative and false positive rates in a diagram. • Evaluate the quality of your predictions. How much training data is needed to yield some acceptable results?

```
AllData <- as.data.frame(read.csv('titanic.csv',colClasses=c("NULL",NA,NA,NA,NA)))
meanError = 0
for(n_folds in 2:10){
errorList = 0
folds_i <- sample(rep(1:n_folds, length.out = dim(X)))
for (k in 1:n_folds) {
    test_i <- which(folds_i == k)
    train_xy <- AllData[-test_i, ]
    test_xy <- AllData[test_i, ]
    Ytrain = train_xy$Survived
    drops <- c("Survived")
    XTrain <- train_xy[ , !(names(train_xy) %in% drops)]

    Ytest = test_xy$Survived
    drops <- c("Survived")
    Xtest <- test_xy[ , !(names(test_xy) %in% drops)]

    # Create a classification tree based on all inputs
    tree = rpart(formula = Ytrain~., XTrain, method="class")
    #predictions
    p = predict(tree, Xtest, type="class")
    #calculate misclassification rate
    errorList <- append(errorList, 1-sum(diag(table(Ytest,p)))/sum(table(Ytest,p)))
}
meanError <- c(meanError, mean(errorList))
}
```

```
## Warning in rep(1:n_folds, length.out = dim(X)): first element used of
## 'length.out' argument
```

```
## Warning in rep(1:n_folds, length.out = dim(X)): first element used of
## 'length.out' argument
```

```
## Warning in rep(1:n_folds, length.out = dim(X)): first element used of
## 'length.out' argument
```

```
## Warning in rep(1:n_folds, length.out = dim(X)): first element used of
## 'length.out' argument
```

```
## Warning in rep(1:n_folds, length.out = dim(X)): first element used of
## 'length.out' argument
```

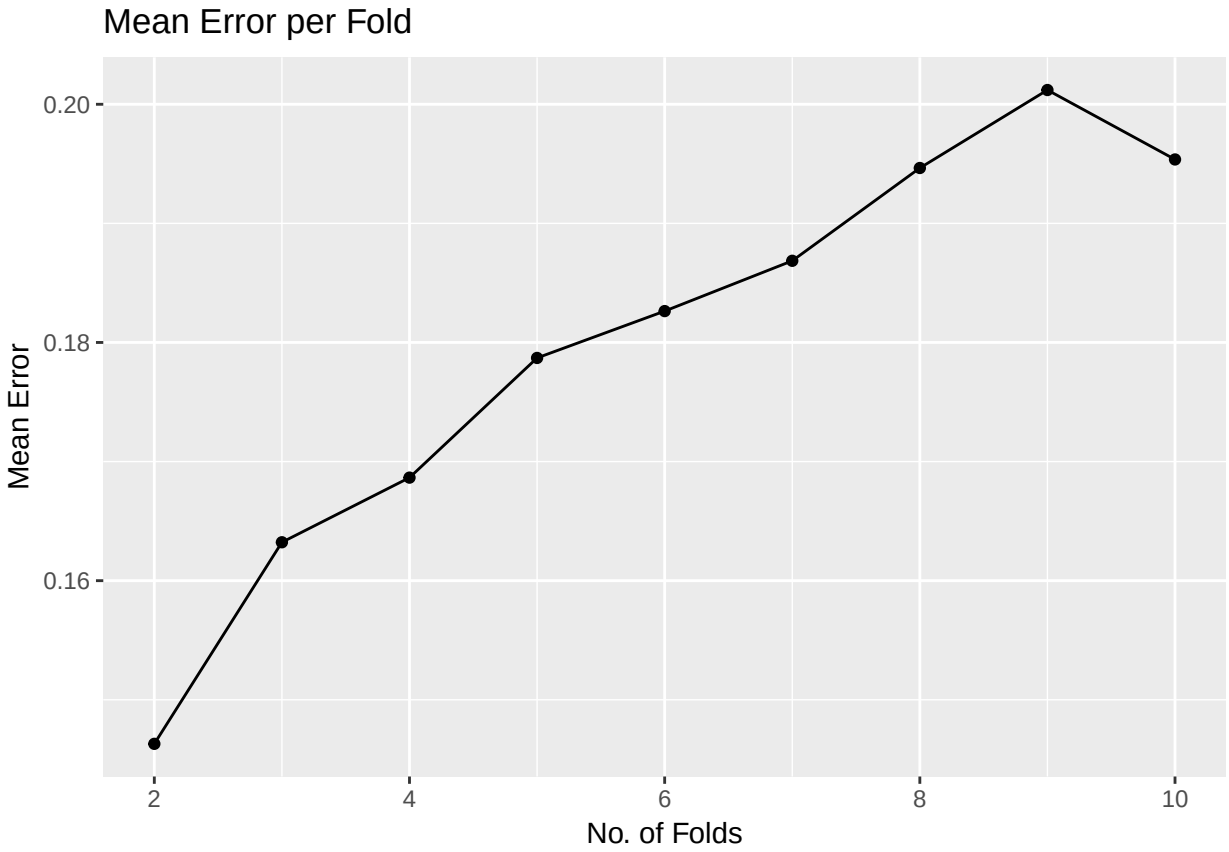
```
## Warning in rep(1:n_folds, length.out = dim(X)): first element used of
## 'length.out' argument
```

```
## Warning in rep(1:n_folds, length.out = dim(X)): first element used of
## 'length.out' argument
```

```
## Warning in rep(1:n_folds, length.out = dim(X)): first element used of
## 'length.out' argument
```

```
## Warning in rep(1:n_folds, length.out = dim(X)): first element used of
## 'length.out' argument
```

```r
meanError <- meanError[2:10]
errors <- data.frame(Mean = meanError, Folds = 2:10)
ggplot(errors, aes(Folds, Mean)) +
    geom_line() +
    geom_point() +
    labs(title = "Mean Error per Fold") +
    labs(x = "No. of Folds") +
    labs(y = "Mean Error")
```



It's counter intuitive to have the lowest mean error at the point of 2 folds. I personally suspect my analysis is wrong, but now that I don't have other data I will go with 5 folds. It's still yielding a small number of errors while in the same time give the opportunity to have multiple models and this leads to avoiding over-training.