Final Project

Overall Intro:

```
  #a.Data Description: Titanic data set includes the following variables-:
  #survival, pclass(Ticket class), sex, age, sibsp(# of siblings/spouses),
  #parch( # of parents/children), ticket, fare, cabin, and embarked.
  #The response variable is survival with inference and prediction as priority.
  #b. Goal is to find out misclassification rate to determine if passengers
   #  survived or not
    # Goal is to find out what are the important predictors in determining the
     #survival
```

Formulating Questions:

```
  #a. Did sex, age, embark location, or any of the other predictors effect
  #individual's survival on the titanic? What is the misclassification rate when
  #determining whether passenger survived? The logistic regression and classification
  #decision tree models are used to answer the question stated. The logistic regression
  #and classification decision tress are used since the response variable- survival,
  #is categorical. More specifically, since there are more than two predictors in
  #this data set with two categories-survived or not survived for the response variable,
  #the Multiple Logistic regression is used. The advantage of using the Multiple
  #Logistic regression allows for the ability to determine the significance of the
  #influential predictors while highlighting the outliers. The advantage of using
  #the classification decision trees is that the visualization is made easier therefore
  #its easy to categorize individuals that survived or not.
```

```
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

Loading and Data Cleaning

```
library(MASS)
library(tree)
library(readxl)
dataset <- read_excel("C:/Users/itzal/OneDrive/Desktop/UH LIFE/YEAR 3 UH/MATH 4322 Into to Data Science

dataset$Sex = as.factor(dataset$Sex)
dataset$Emarked = as.factor(dataset$Emarked)
dataset$Pclass = as.factor(dataset$Pclass)
NA_Age=sum(is.na(dataset$Age))
NA_Ticket=sum(is.na(dataset$Ticket))
NA_Survied = sum(is.na(dataset$Survived))
NA_Cabin=sum(is.na(dataset$Cabin))
percent_cabin_na = (NA_Cabin/1309) * 100  #77.2%
```

```r
#remove the Name, Ticket, and Cabin columns
dataset_One = subset(dataset, select = -c(Name,Ticket, Cabin))
sum=0;
totalObs=0
for(i in 1:length(dataset_One$Age)){
  if(! any(is.na(dataset_One$Age[i]))) #if there is no NA val
    {
      sum = sum+dataset_One$Age[i]
      totalObs=totalObs+1
    }
}
avgAge = sum/ totalObs
for(i in 1:length(dataset_One$Age)){
  if(any(is.na(dataset_One$Age[i]))) #if there is  NA val
  {
    dataset_One$Age[i]=avgAge # set NA to the ageAge value
  }
}
```

Creating Model on the Data Set

```r
#2b. Creating a Multiple Logistic Model
model_avgAge.glm = glm(Survived ~ .,
                data = dataset_One,
                family = "binomial")
summary(model_avgAge.glm)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = "binomial", data = dataset_One)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7678  -0.4890  -0.3480   0.4842   2.5210
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.2989099  0.4530988   9.488  < 2e-16 ***
## PassengerId -0.0002422  0.0002197  -1.102 0.270277
## Pclass2     -0.9462872  0.2752179  -3.438 0.000585 ***
## Pclass3     -1.9286716  0.2726924  -7.073 1.52e-12 ***
## Sexmale     -3.7334290  0.1899960 -19.650  < 2e-16 ***
## Age         -0.0326683  0.0071534  -4.567 4.95e-06 ***
## SibSp       -0.2951968  0.0940088  -3.140 0.001689 **
## Parch       -0.0813450  0.0987599  -0.824 0.410130
## Fare         0.0022635  0.0020526   1.103 0.270146
## EmarkedQ     0.1394848  0.3486300   0.400 0.689087
## EmarkedS    -0.2381533  0.2187464  -1.089 0.276278
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 1730.29  on 1305  degrees of freedom
## Residual deviance:  962.01  on 1295  degrees of freedom
##   (3 observations deleted due to missingness)
## AIC: 984.01
##
## Number of Fisher Scoring iterations: 5
```

Analysis on the Data Set

```
# According to this data set, the most important predictors were Pclass(Ticket class),
#   Sexmale, and Age.
# Logit = 4.3 -0.94(PClass2) - 1.92(PClass3) - 3.73(SexMale) - 0.032(Age) - 0.29(SibSp)
# Where Sex = 0 if female; 1 if male
# Where PClass2 = 0 if Class is not 2;1 if Class=2, PClass3 = 0 if Class is not 3;
   #1 if Class=3

#As the ticket class number, age, and the number of siblings/spouses increase,
#the probability of survival decreases.
#Observation: Being male decreases the chances of survival.
```

Logistic Regression Prediction on the Data Set:

```
# 2c)
  set.seed (2)
  error_rate = 0
  for (i in 1:10) {
    sample <- sample.int(n=nrow(dataset_One),
                         size= floor((.80)*nrow(dataset_One)),
                         replace = F)
   train_set_one <- dataset_One[sample,]
   test_set_one <- dataset_One[-sample,]

   model_avgAge.glm.train <- glm(Survived ~ .,
                                 data = train_set_one,
                                 family = "binomial",
                                 subset = sample)
   model_avgAge.glm.train.pred <- predict(model_avgAge.glm.train, test_set_one,
                                          type = "response")
   con.matrix = table(model_avgAge.glm.train.pred>0.5,test_set_one$Survived)
   con.matrix
   error_rate = error_rate + (con.matrix[1,2]+con.matrix[2,1])/sum(con.matrix)
}
final = error_rate/10
final
```

```
## [1] 0.1480247
```

Prediction Analysis of the model: Since the data is split into training and test set, we are trying to predict if whether or not the observation will survive. According to the prediction, there is about 15% of misclassification.

We also did Inference Logistic Regression model on the full data set as we wanted to know what are the significant predictors that effected the survival.

Logistic Regression Inference on the full Data Set

```r
#2d. Model fit on full data
#Multiple Logistics Regression with full Data Set
model_avgAge.glm.fullDatasetOne <- glm(Survived ~ .,
                                data = dataset_One,
                                family = "binomial")
summary(model_avgAge.glm.fullDatasetOne)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = "binomial", data = dataset_One)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7678  -0.4890  -0.3480   0.4842   2.5210
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.2989099  0.4530988   9.488  < 2e-16 ***
## PassengerId -0.0002422  0.0002197  -1.102 0.270277
## Pclass2     -0.9462872  0.2752179  -3.438 0.000585 ***
## Pclass3     -1.9286716  0.2726924  -7.073 1.52e-12 ***
## Sexmale     -3.7334290  0.1899960 -19.650  < 2e-16 ***
## Age         -0.0326683  0.0071534  -4.567 4.95e-06 ***
## SibSp       -0.2951968  0.0940088  -3.140 0.001689 **
## Parch       -0.0813450  0.0987599  -0.824 0.410130
## Fare         0.0022635  0.0020526   1.103 0.270146
## EmarkedQ     0.1394848  0.3486300   0.400 0.689087
## EmarkedS    -0.2381533  0.2187464  -1.089 0.276278
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1730.29  on 1305  degrees of freedom
## Residual deviance:  962.01  on 1295  degrees of freedom
##   (3 observations deleted due to missingness)
## AIC: 984.01
##
## Number of Fisher Scoring iterations: 5
```

```r
model_avgAge.glm.fullDatasetOnePred <- predict(model_avgAge.glm.fullDatasetOne,
                                               dataset_One, type = "response")
con.matrix.fullDatasetOnePred = table(model_avgAge.glm.fullDatasetOnePred>0.5,
                                      dataset_One$Survived)
con.matrix.fullDatasetOnePred
```

```
##
##           0   1
##   FALSE 735 109
##   TRUE   79 383
```

```
error_rate.fullDatasetOnePred =(con.matrix.fullDatasetOnePred[1,2]+
                                  con.matrix.fullDatasetOnePred[2,1])/
sum(con.matrix.fullDatasetOnePred)
error_rate.fullDatasetOnePred
```

```
## [1] 0.143951
```

Analysis on the full Data Set

```
#According to the inference, the most significant predictors are Pclass(Ticket class),
#   Sexmale, and Age.
#Observation: Prediction or inference did not make any significant changes
# to the results as they both share the same significant predictors and have similar
# misclassification  rate.
```

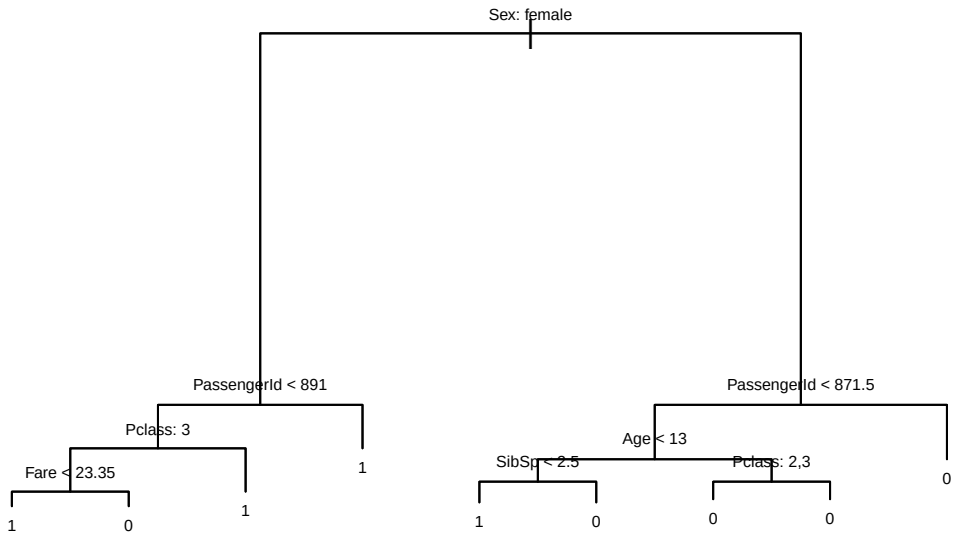
Classification Trees on the Data Set

```
library(tree)
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
library(boot)
set.seed(1)
dataset_One = na.omit(dataset_One)
dataset_One$Survived <- as.factor(dataset_One$Survived)
dataset_One$Emarked<- as.factor(dataset_One$Emarked)
train <- sample.int(n=nrow(dataset_One),
          size= floor((.80)*nrow(dataset_One)),
          replace = F)
survived.train <- dataset_One[train,]
survived.test <- dataset_One[-train,]
tree.survived = tree(Survived ~. , data=survived.train)
plot(tree.survived)
text(tree.survived, pretty=0, cex=0.5)
```



Calculating Test Error Rate for classification Tree on the Data Set

```
tree.survived.predict = predict(tree.survived, survived.test, type='class')
table(tree.survived.predict, survived.test$Survived)
```

```
##
## tree.survived.predict   0   1
##                     0 139  21
##                     1  13  89
```

```
error_rate = (con.matrix[1,2]+con.matrix[2,1])/sum(con.matrix)
error_rate
```

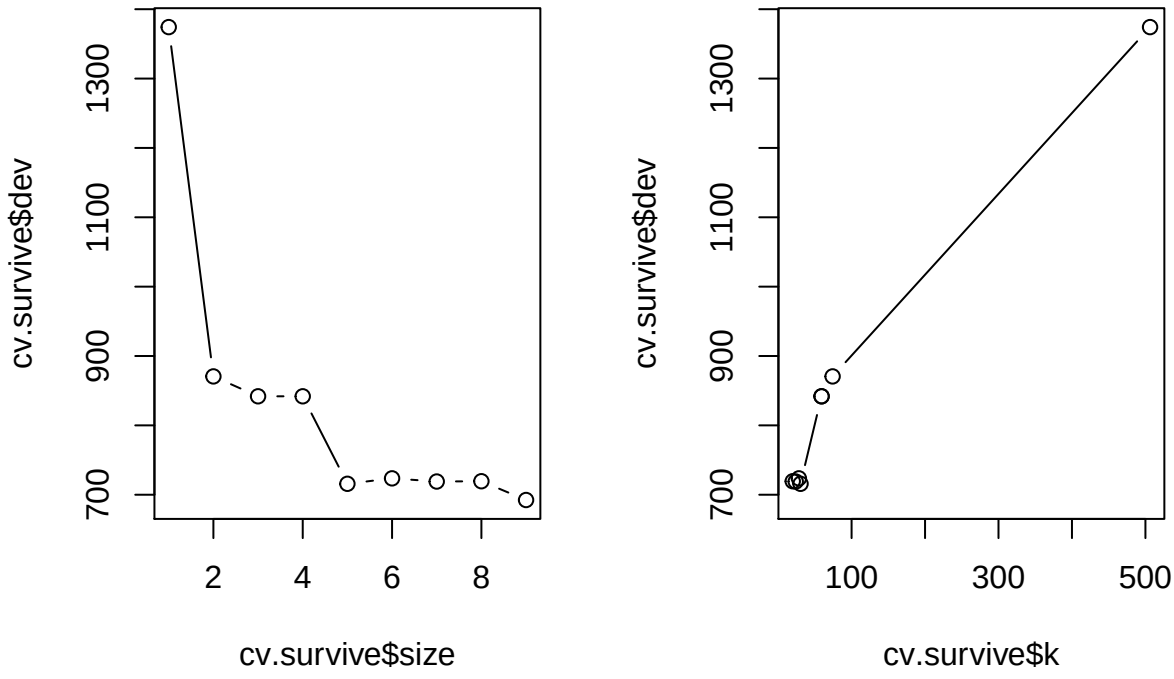
```
## [1] 0.1264368
```

Analysis for the Classification Tree:

```
#Survival(Y/N) = Sex+PassengerId+Age+Pclass+SibSp+Fare
# The mislassifcation rate is 12.6%.
```

Pruning - Data Set

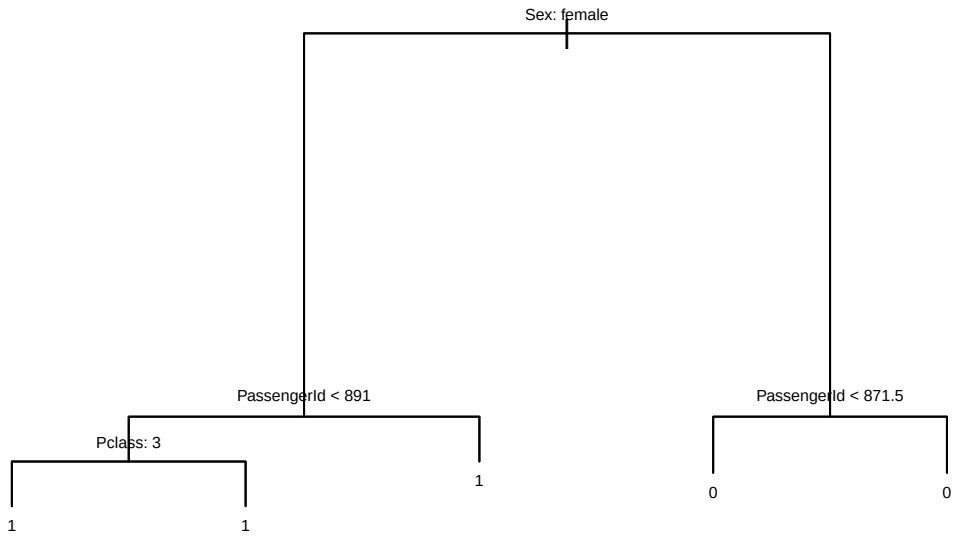
```r
#Prune - dataset_One
cv.survive <- cv.tree(tree.survived, FUN=prune.tree)
par(mfrow=c(1, 2))
plot(cv.survive$size, cv.survive$dev, type="b")
plot(cv.survive$k, cv.survive$dev, type="b")
```




```r
pruned.survived = prune.tree(tree.survived, best=5)
par(mfrow=c(1, 1))
summary(pruned.survived)
```

```
##
## Classification tree:
## snip.tree(tree = tree.survived, nodes = c(8L, 6L))
## Variables actually used in tree construction:
## [1] "Sex"         "PassengerId" "Pclass"
## Number of terminal nodes:  5
## Residual mean deviance:  0.6472 = 672.5 / 1039
## Misclassification error rate: 0.1466 = 153 / 1044
```

```r
plot(pruned.survived)
text(pruned.survived, pretty=0, cex=0.5)
```



Analysis on Pruning the Data Set

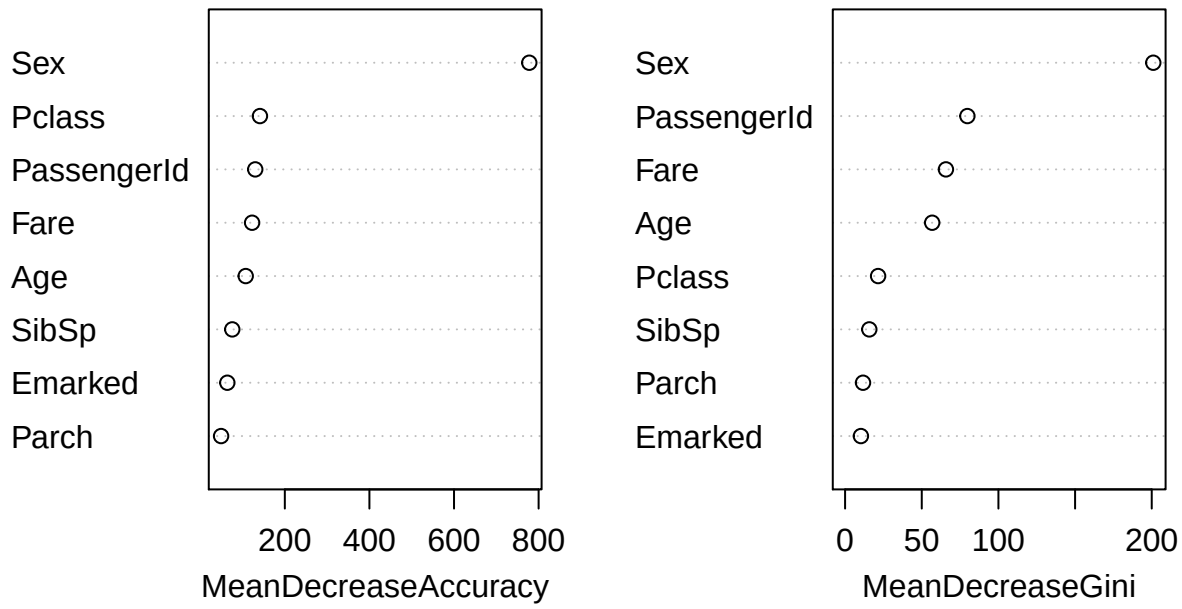
```
#According the plot above, the best size to prune the tree is 5.
# In this case, pruning the tree did not help as the error rate went from 12.6%
# to 14.6%.

#Survival(Y|N) = Sex+PassengerId+Pclass
```

Now, we applied Random Forest to get a better error rate for the tree.

Classification Tree RF- Data Set One

```
#Random Forest - Dataset_One
p = ncol(dataset_One) - 1 #No. of predictors
B = 10000 #No. of bootstrap trees
bag.model = randomForest(Survived ~ .,
                         data = survived.train,
                         ntree = B,
                         mtry = sqrt(p),
                         na.action = na.omit,
                         importance = TRUE)
varImpPlot(bag.model)
```




```
predict.rf = predict(bag.model, survived.test, type='class')
matrix.con = table(predict.rf, survived.test$Survived)

test_error_rate = (matrix.con[1,2]+matrix.con[2,1])/sum(matrix.con)
test_error_rate
```

```
## [1] 0.1068702
```

Analysis RF Tree on the Data Set

```
#Applying random forest to the tree resulted in a better misclassifcation rate
# of 10.7%.
#According to the random forest tree, the most significant predictor is Sex.
```

Classification RF for 10 iterations - Data Set One

```
error_rate = 0
set.seed(22)
for (i in 1:10) {
    train3 <- sample.int(n=nrow(dataset_One),
              size= floor((.80)*nrow(dataset_One)),
              replace = F)
    survived.train3 <- dataset_One[train3,]
    survived.test3 <- dataset_One[-train3,]
```

```
    B = 10000 #No. of bootstrap trees
    p = ncol(dataset_One) - 1 #No. of predictors
    bag.model.3 = randomForest(Survived ~ .,
                               data = survived.train3,
                               ntree = B,
                               mtry = sqrt(p),
                               na.action = na.omit,
                               importance = TRUE)

    predict.rf3 = predict(bag.model.3, survived.test3, type='class')
    mat=table(predict.rf3, survived.test3$Survived)
    error_rate = error_rate + (mat[1,2]+mat[2,1])/sum(mat)
    error_rate
}
final=error_rate / 10
final
```

```
## [1] 0.1156489
```

Analysis for 10 iterations on Random Forest:

```
# The misclassifcation rate increased slightly from 10.7% to 11.6%
```

Classification RF Interpretation - Full Data Set

```
B = 10000 #No. of bootstrap trees
bag.model.k = randomForest(Survived ~ .,
                           data = dataset_One,
                           ntree = B,
                           mtry = 5,
                           na.action = na.omit,
                           importance = TRUE)
bag.model.k #To get an outline of bagging results.
```

```
##
## Call:
##  randomForest(formula = Survived ~ ., data = dataset_One, ntree = B,      mtry = 5, importance = TRU
##                Type of random forest: classification
##                      Number of trees: 10000
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 12.17%
## Confusion matrix:
##     0   1 class.error
## 0 755  59  0.07248157
## 1 100 392  0.20325203
```

```
varImpPlot(bag.model.k)
```

bag.model.k


```
#Using k=5 on the whole data set, we observe a estimated
#error rate of 12.17%
```

Conclusion

```
#Top predictive performance, statistically, comes from the single randomForest
#as it has the lesser misclassifcation error rate compared to logistic model.
#Of all the predictors, the most important predictor is Sex.
```