Predict survival of the Titanic

Maria George

December 4, 2015

```
#Loading the necessary packages

# install.packages("pROC")
library(pROC) # Useful for computing and plotting classifer metrics

# install.packages("arm")
library(arm) # For small datasets, more stable learning methods

# install.packages("randomForest")
library(randomForest) # R package to fit a random forest model

library(ggplot2) # R package to plot the data
```

```
# Loading the titanic data and storing it into a local variable
titanic.Data <- read.csv(file = "titanic.csv")

# Displaying the first few rows of the dataset
head(titanic.Data)
```

Splitting your data into a training and test set based on an 80-20 split, in other words, 80% of the observations will be in the training set.

```
##   pclass survived                                             name    sex
## 1      1        1                    Allen, Miss. Elisabeth Walton female
## 2      1        1                   Allison, Master. Hudson Trevor   male
## 3      1        0                     Allison, Miss. Helen Loraine female
## 4      1        0             Allison, Mr. Hudson Joshua Creighton   male
## 5      1        0 Allison, Mrs. Hudson J C (Bessie Waldo Daniels) female
## 6      1        1                               Anderson, Mr. Harry   male
##       age sibsp parch ticket     fare   cabin embarked boat body
## 1 29.0000     0     0  24160 211.3375      B5        S    2   NA
## 2  0.9167     1     2 113781 151.5500 C22 C26        S   11   NA
## 3  2.0000     1     2 113781 151.5500 C22 C26        S        NA
## 4 30.0000     1     2 113781 151.5500 C22 C26        S       135
## 5 25.0000     1     2 113781 151.5500 C22 C26        S        NA
## 6 48.0000     0     0  19952  26.5500     E12        S    3   NA
##                         home.dest
## 1                     St Louis, MO
## 2 Montreal, PQ / Chesterville, ON
## 3 Montreal, PQ / Chesterville, ON
## 4 Montreal, PQ / Chesterville, ON
## 5 Montreal, PQ / Chesterville, ON
## 6                     New York, NY
```

```r
#View(titanic.Data)

# Displaying the summary of the dataset
summary(titanic.Data)
```

```
##      pclass          survived                                      name
##  Min.   :1.000   Min.   :0.000   Connolly, Miss. Kate            :   2
##  1st Qu.:2.000   1st Qu.:0.000   Kelly, Mr. James                :   2
##  Median :3.000   Median :0.000   Abbing, Mr. Anthony             :   1
##  Mean   :2.295   Mean   :0.382   Abbott, Master. Eugene Joseph   :   1
##  3rd Qu.:3.000   3rd Qu.:1.000   Abbott, Mr. Rossmore Edward     :   1
##  Max.   :3.000   Max.   :1.000   Abbott, Mrs. Stanton (Rosa Hunt):   1
##                                  (Other)                         :1301
##      sex           age              sibsp            parch
##  female:466   Min.   : 0.1667   Min.   :0.0000   Min.   :0.000
##  male  :843   1st Qu.:21.0000   1st Qu.:0.0000   1st Qu.:0.000
##               Median :28.0000   Median :0.0000   Median :0.000
##               Mean   :29.8811   Mean   :0.4989   Mean   :0.385
##               3rd Qu.:39.0000   3rd Qu.:1.0000   3rd Qu.:0.000
##               Max.   :80.0000   Max.   :8.0000   Max.   :9.000
##               NA's   :263
##       ticket          fare                      cabin      embarked
##  CA. 2343:  11   Min.   :  0.000                  :1014    :  2
##  1601    :   8   1st Qu.:  7.896   C23 C25 C27    :   6   C:270
##  CA 2144 :   8   Median : 14.454   B57 B59 B63 B66:   5   Q:123
##  3101295 :   7   Mean   : 33.295   G6             :   5   S:914
##  347077  :   7   3rd Qu.: 31.275   B96 B98        :   4
##  347082  :   7   Max.   :512.329   C22 C26        :   4
##  (Other) :1261   NA's   :1         (Other)        : 271
##       boat           body                      home.dest
##         :823   Min.   :  1.0                        :564
##  13     : 39   1st Qu.: 72.0   New York, NY         : 64
##  C      : 38   Median :155.0   London               : 14
##  15     : 37   Mean   :160.8   Montreal, PQ         : 10
##  14     : 33   3rd Qu.:256.0   Cornwall / Akron, OH:  9
##  4      : 31   Max.   :328.0   Paris, France        :  9
##  (Other):308   NA's   :1188    (Other)              :639
```

```r
#Displaying he structure if the dataset
str(titanic.Data)
```

```
## 'data.frame':    1309 obs. of  14 variables:
##  $ pclass   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ survived : int  1 1 0 0 0 1 1 0 1 0 ...
##  $ name     : Factor w/ 1307 levels "Abbing, Mr. Anthony",..: 22 24 25 26 27 31 46 47 51 55 ...
##  $ sex      : Factor w/ 2 levels "female","male": 1 2 1 2 1 2 1 2 1 2 ...
##  $ age      : num  29 0.917 2 30 25 ...
##  $ sibsp    : int  0 1 1 1 1 0 1 0 2 0 ...
##  $ parch    : int  0 2 2 2 2 0 0 0 0 0 ...
##  $ ticket   : Factor w/ 929 levels "110152","110413",..: 188 50 50 50 50 125 93 16 77 826 ...
##  $ fare     : num  211 152 152 152 152 ...
##  $ cabin    : Factor w/ 187 levels "","A10","A11",..: 45 81 81 81 81 151 147 17 63 1 ...
##  $ embarked : Factor w/ 4 levels "","C","Q","S": 4 4 4 4 4 4 4 4 4 2 ...
```

```
##  $ boat     : Factor w/ 28 levels "","1","10","11",..: 13 4 1 1 1 14 3 1 28 1 ...
##  $ body     : int  NA NA NA 135 NA NA NA NA NA 22 ...
##  $ home.dest: Factor w/ 370 levels "","?Havana, Cuba",..: 310 232 232 232 232 238 163 25 23 230 ...
```

```
# Counting the number of rows in the dataset
rowcount <- nrow(titanic.Data)
# Setting a random seed so that results can be reproduced
set.seed(1)

# Defining the indices of the training dataset
# 80% of the observations are in training set
train <- sample(rowcount, as.integer(0.8*rowcount))
# Calculating the length of training dataset
length(train)
```

```
## [1] 1047
```

```
# Defining the training datasset based on the indices obatined
training.Data <- titanic.Data[train, ]
# Displaying the number of observations in training dataset
nrow(training.Data)
```

```
## [1] 1047
```

```
# Defining the testing dataset
test.Data <- titanic.Data[-train, ]
# Displaying the number of observations in training dataset
nrow(test.Data)
```

```
## [1] 262
```

The dataset titanic is loaded and saved into a local variable titanic.Data. The sample() function is used to randomly calculate the indices for training data. 80% of the observations are used as training data. The origininal data records 1309 observations of 14 variables. Training data contains 1047 observations. Testing data contains 262 observations.

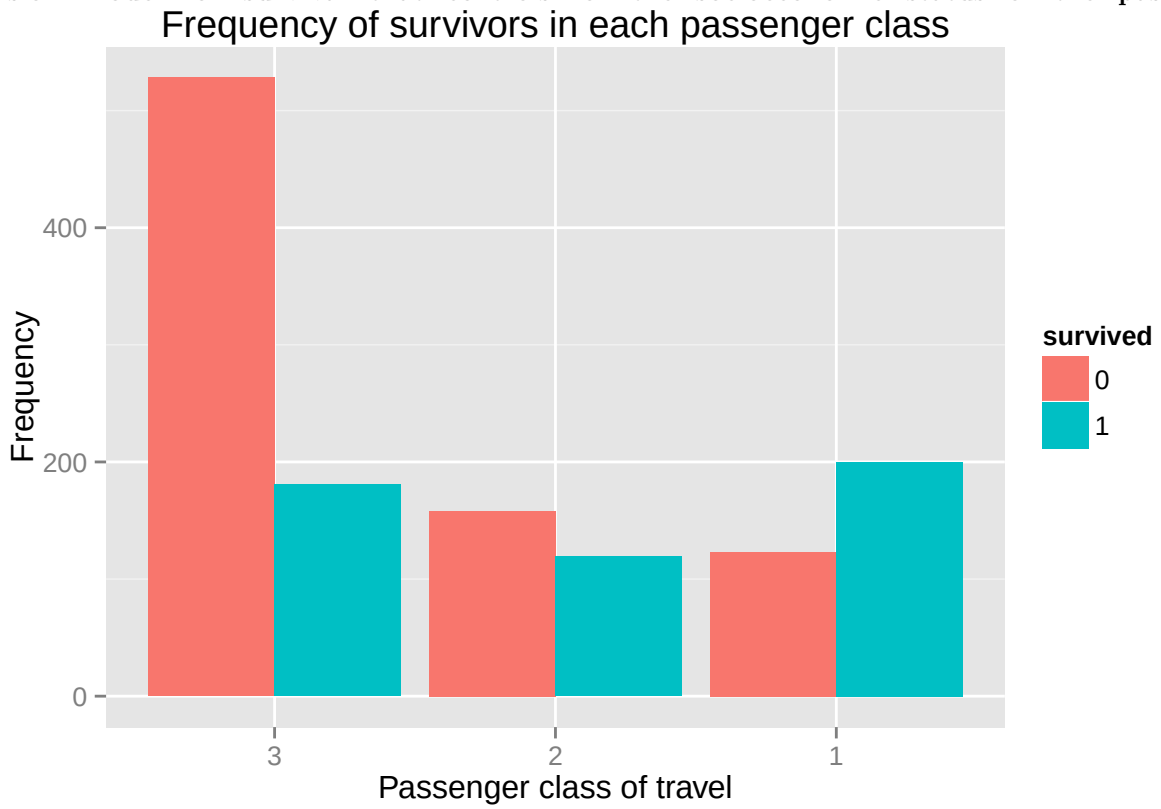
```
# Converting the column pclass into a factor variable
# passenger class 3 being the lowest and class 1 being the highest
titanic.Data$pclass <- factor(titanic.Data$pclass,levels = c(3, 2, 1),
                              ordered = TRUE)

# Converting the response variable survived into a factor variable
# 0 representing survived and 1 representing not survived
titanic.Data$survived <- factor(titanic.Data$survived, levels = c(0,1),
                                ordered = TRUE)

# Plotting the frequency of survivors in each passenger class
ggplot(titanic.Data, aes(pclass, fill = survived)) +
        geom_bar(position="dodge") +
```

```
  xlab ("Passenger class of travel") +
  ylab("Frequency") +
  ggtitle("Frequency of survivors in each passenger class")
```

The goal is to predict the survival of passengers. Firstly, training a logistic regression model for survival that controls for the socioeconomic status of the passenger.



From the graph, it can be observed that the number of people who did not survive is highest for passenger class 3 and lowest for passenger class 1. Whereas, the number of survivors is highest for passenger class 1 and lowest for passenger class 3.

```
# Reloading the titanic data again as pclass and survived were converted to
# ordered factors
titanic.Data <- read.csv(file = "titanic.csv")

# Converting pclass and survived into factor variable
titanic.Data$pclass <- as.factor(titanic.Data$pclass)
titanic.Data$survived <- as.factor(titanic.Data$survived)

# Redefining the training and test data as the datatype was changed
training.Data <- titanic.Data[train, ]
test.Data <- titanic.Data[-train, ]

# Fitting a logistic regression model to the training dataset
# response variable: survived and predictor variable: pclass(class of the passenger)
glm.passengerClass <- glm(survived ~ pclass, training.Data,
                          family = binomial)
# Displaying the summary statistics of the fitted model
summary(glm.passengerClass)
```

```
##
## Call:
## glm(formula = survived ~ pclass, family = binomial, data = training.Data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.363  -0.771  -0.771   1.003   1.648
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.4253     0.1286   3.307 0.000943 ***
## pclass2      -0.6876     0.1850  -3.717 0.000202 ***
## pclass3      -1.4864     0.1607  -9.251  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1390.7  on 1046  degrees of freedom
## Residual deviance: 1297.4  on 1044  degrees of freedom
## AIC: 1303.4
##
## Number of Fisher Scoring iterations: 4
```

From the summary statistics of the fitted model, we can observe that the intercept estimate is 0.4253. The p-value is less than 0.5, indicating that the there is a statistically significant association between class of the passenger(pclass) and survival. The slope parameter of pclass2 is -0.6876 and that of pclass 3 is -1.4864. All the parameters have a p-value less than 0.05 indicating that they are statistically significant.

In the dataset, passenger class 1 (pclass = 1) represents passengers with higher socioeconomic status and pclass = 3 represents passengers with lowest socioeconomic status. It can be seen that the as the passenger class increases, the log odds of survival rate decreases by 0.6876 for class 2 passengers and by 1.4864 for class 3 passengers. This means that as the socioeconomic status of the passenger decreases, the number of survived passenger also decreases.

Based on this model, calculating the probability of survival for lower class passengers Logistic reression model is represented mathematically as, log(p/1+p) = b0 + b1 * x

In this case, let p denote probability of survival, then, log(p/1+p) = b0 + b1 * pclass

By substituting the value of b0 = 0.4253, b1 = -1.4864 (obtained from summary statistics of fiited model-glm.passengerClass) and pclass = 3 (lower class passengers),

==> p = 0.2570

Thus the probability of survival of the lower class of passengers is 0.259129

```r
# Finding the probability of surviaval of only the lower class passengers in the
# raining dataset
training.prediction <- predict(glm.passengerClass, subset(training.Data, pclass==3), type = "response")

# Displaying the first few probability values
head(training.prediction)
```

```
##       749      1187      1172      1231       861       819
## 0.2570922 0.2570922 0.2570922 0.2570922 0.2570922 0.2570922
```

From the above code, it can be seen that the probability of survial of the lower class passengers as predicted by the fitted model is 0.2570922. This is same as the mathematically derived value of 0.2570.

Next, Evaluating the performance of this model.

```
# Predicting the survival of passengers from the test data using the fitted
# model
yhat <- predict(glm.passengerClass, test.Data,
                    type = "response")
```

Predicting the survival of passengers for each observation in the test set using the model fitted above and saving these predictions as yhat. predict() function is used to predict the survival rate of passengers from the test data using the fitted model glm.passengerClass.

yhat now contains the probability of survival of each observation in the test dataset.

```
# Creating a vector of 0 with length same as the number of rows of test data
glm.pred <- rep(0, nrow(test.Data))

# Transformimg to 1(survived) for observations for which predicted probability
# exceeds 0.5
glm.pred[yhat >.5] <- 1
```

Using a threshold of 0.5 to classify predictions. The above commands create a vector of class predictions based on whether the predicted probability of a survival is greater than or less than 0.5. The first command creates a vector of 0 with length same as the number of rows of test data. The second line transforms to 1 all of the elements for which the predicted probability of a suvival exceeds 0.5.

Given these predictions, a confusion matrix is created in order to determine how many observations were correctly or incorrectly classified.

```
# Creating a confusion matrix to determine how many observations were
# correctly or incorrectly classified
confmatrix.pclass <-table(glm.pred,test.Data$survived)

# Displaing the confusion matrix
confmatrix.pclass
```

```
##
## glm.pred   0   1
##        0 137  55
##        1  23  47
```

```r
# Calculating the accuracy
sum(diag(confmatrix.pclass))/sum(confmatrix.pclass)
```

```
## [1] 0.7022901
```

False positives are defined as observations where the model predicted a positive value when in fact that the actual value is negative. In the titanic data, a false positive is defined as observations where the model predicted the passenger as survived, when according to the actual observation, the passenger did not survive.

From the confusion matrix we can see that the number of false postives are 23. This indicates that 23 passengers which the model predicted will survive, actually did not survive.

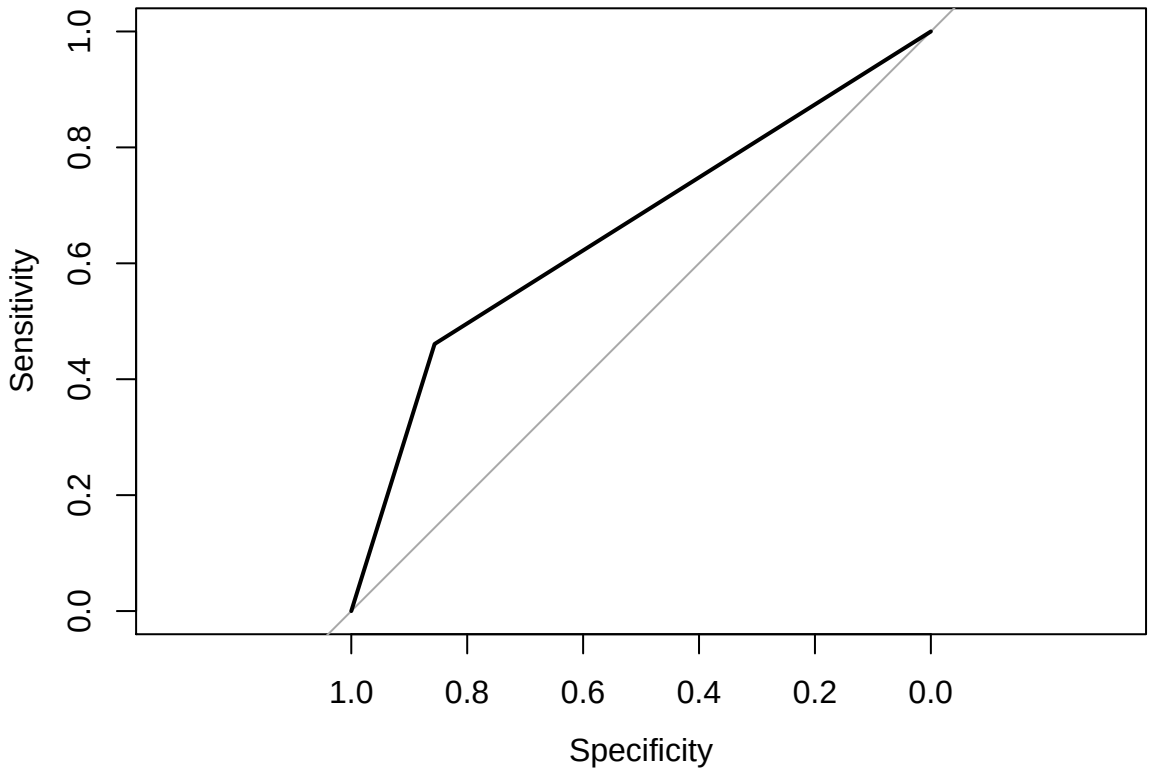
Also, the accuracy of the model is 70.22901%. Thus the misclassification rate is about 29.77%. This includes false positive and false negative.

```r
# Building a roc curve using predictor varaiable pclass
roc.titanic <- roc(test.Data$survived, glm.pred)
# Displaying the value of the roc object
roc.titanic
```

Plotting the ROC curve for this model.

```
##
## Call:
## roc.default(response = test.Data$survived, predictor = glm.pred)
##
## Data: glm.pred in 160 controls (test.Data$survived 0) < 102 cases (test.Data$survived 1).
## Area under the curve: 0.6585
```

```r
# Plotting the roc curve
plot(roc.titanic)
```




```
##
## Call:
## roc.default(response = test.Data$survived, predictor = glm.pred)
##
## Data: glm.pred in 160 controls (test.Data$survived 0) < 102 cases (test.Data$survived 1).
## Area under the curve: 0.6585
```

ROC or the Receiver operating characteristic curve is a measure of classifier performance. This curve is plotted by taking into account the proportions of positive data points that are correctly classified as positive (true positives or sensitivity) and proportion of negative data points that are mistakenly considered as positive (false positives or 1-specificity).

The curve represents the variation of true positives and false positives as we vary the threshold value. Thus, the curve shows the trade off between the two values. The ideal ROC curve hugs the top left corner, indicating a high true positive rate and a low false positive rate. The dotted line represents the "no information" classifier, that is the curve we would have obtaines if passenger class and survival rate was not associated.

Thus, from the graph it is clear that there is a relation between class of the passenger and the survival rate. The overall performance of a classifier, summarized over all possible thresholds, is given by the area under the (ROC). Fromthe summary statistic of roc.titanic, the Area under the curve: 0.6911. A classifier that performs no better than chance have an AUC of 0.5. With an ROC of 0.6911, which is above 0.5, the passenger class can be used to predict survival rate, and not just beacuse of chance. Models with AUC more than 0.8 are said to have a good performance in classifying the resposnse varaiable. With AUC of 0.6911, this model performs moderately well in classifying the response variable.

```
# Calculating the number of NAs for the age column
length(which(is.na(titanic.Data$age)))
```

Using the data to construct a new predictor variable based on a passenger's listed title (i.e. Mr., Mrs., Miss., Master).

```
## [1] 263
```

From the above code, it can be seen that 263 observations have their age as NA. This accounts to approximately, $263/1309 = 20\%$ of the observations.

Thus, Title of the passenger is an interesting variable to predict passenger survival. Title can be obatined from name of the passenger, which is present in the datatset for all the obervations. Title can be used as a representative for the age varaible. Moreover, Title encapsulates age, gender, marital status. Thus, title can play an important role in the prediction of survival rate.

```
# Using the custom function to extract title from the column name
f <- function(name) {
for (title in c("Master", "Miss", "Mrs.", "Mr.")) {
  if (grepl(title, name)) {
     return(title)
} }
return("Nothing")
}

# Extracting title from the column name for each observation in the titanic.Data
# dataset.
titanic.Data$title <- lapply(titanic.Data$name, f)

# Converting title to factor variable
titanic.Data$title <- as.character(titanic.Data$title)
titanic.Data$title <- as.factor(titanic.Data$title)

# Checking the structure of the titanic.Data with the addition of new column
# title
str(titanic.Data)
```

Custom function to add this predictor to the dataset.

```
## 'data.frame':    1309 obs. of  15 variables:
##  $ pclass   : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ survived : Factor w/ 2 levels "0","1": 2 2 1 1 1 2 2 1 2 1 ...
##  $ name     : Factor w/ 1307 levels "Abbing, Mr. Anthony",..: 22 24 25 26 27 31 46 47 51 55 ...
##  $ sex      : Factor w/ 2 levels "female","male": 1 2 1 2 1 2 1 2 1 2 ...
##  $ age      : num  29 0.917 2 30 25 ...
##  $ sibsp    : int  0 1 1 1 1 0 1 0 2 0 ...
##  $ parch    : int  0 2 2 2 2 0 0 0 0 0 ...
##  $ ticket   : Factor w/ 929 levels "110152","110413",..: 188 50 50 50 50 125 93 16 77 826 ...
##  $ fare     : num  211 152 152 152 152 ...
##  $ cabin    : Factor w/ 187 levels "","A10","A11",..: 45 81 81 81 81 151 147 17 63 1 ...
##  $ embarked : Factor w/ 4 levels "","C","Q","S": 4 4 4 4 4 4 4 4 4 2 ...
##  $ boat     : Factor w/ 28 levels "","1","10","11",..: 13 4 1 1 1 14 3 1 28 1 ...
##  $ body     : int  NA NA NA 135 NA NA NA NA NA 22 ...
##  $ home.dest: Factor w/ 370 levels "","?Havana, Cuba",..: 310 232 232 232 232 238 163 25 23 230 ...
##  $ title    : Factor w/ 5 levels "Master","Miss",..: 2 1 2 3 4 3 2 3 4 3 ...
```

Title of the passenger is extracted from the name column of titanic.Data dataset using the custom function f and stored as a new column title. This is done for all the observations in the datatset using the lapply() function. Since the lappply() function returns a list as response, the title column is then converted to factor variable.

```
# Redefining the training and testing dataset to include the new column title

# pclass was converted to factor variables for plotting
# purpose. Reconverting them back to integer
# titanic.Data$pclass <- as.factor(titanic.Data$pclass)

# Checking the structure titanic.Data dataset
str(titanic.Data)
```

```
## 'data.frame':    1309 obs. of  15 variables:
##  $ pclass   : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ survived : Factor w/ 2 levels "0","1": 2 2 1 1 1 2 2 1 2 1 ...
##  $ name     : Factor w/ 1307 levels "Abbing, Mr. Anthony",..: 22 24 25 26 27 31 46 47 51 55 ...
##  $ sex      : Factor w/ 2 levels "female","male": 1 2 1 2 1 2 1 2 1 2 ...
##  $ age      : num  29 0.917 2 30 25 ...
##  $ sibsp    : int  0 1 1 1 1 0 1 0 2 0 ...
##  $ parch    : int  0 2 2 2 2 0 0 0 0 0 ...
##  $ ticket   : Factor w/ 929 levels "110152","110413",..: 188 50 50 50 50 125 93 16 77 826 ...
##  $ fare     : num  211 152 152 152 152 ...
##  $ cabin    : Factor w/ 187 levels "","A10","A11",..: 45 81 81 81 81 151 147 17 63 1 ...
##  $ embarked : Factor w/ 4 levels "","C","Q","S": 4 4 4 4 4 4 4 4 4 2 ...
##  $ boat     : Factor w/ 28 levels "","1","10","11",..: 13 4 1 1 1 14 3 1 28 1 ...
##  $ body     : int  NA NA NA 135 NA NA NA NA NA 22 ...
##  $ home.dest: Factor w/ 370 levels "","?Havana, Cuba",..: 310 232 232 232 232 238 163 25 23 230 ...
##  $ title    : Factor w/ 5 levels "Master","Miss",..: 2 1 2 3 4 3 2 3 4 3 ...
```

```
# Redefining the training datasset
training.Data <- titanic.Data[train, ]

# Redefining the testing dataset
test.Data <- titanic.Data[-train, ]
```

As the titanic.Data now contains a new column title, the training and testing data are redefined to continue the analysis. Also, earlier in the analysis, the column pclass was converted to factors for plotting purpose. Bothe the coluns are converted back to the original int datatype. The training and test dataset are then redefined.

```
# Fitting a logistic regression model with pclass and title as predictor variables
glm.title <- glm(survived ~ pclass + title, training.Data,
                 family = binomial)

# Displaying the summary statistic of the fitted model
summary(glm.title)
```

Fitting a second logistic regression model including this new feature and analyzing if the new feature improved the model performance

```
##
## Call:
## glm(formula = survived ~ pclass + title, family = binomial, data = training.Data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1896  -0.6410  -0.4155   0.6704   2.2327
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.6909     0.3544   4.771 1.84e-06 ***
## pclass2       -0.9235     0.2269  -4.070 4.71e-05 ***
## pclass3       -1.8515     0.2057  -9.003  < 2e-16 ***
## titleMiss      0.3990     0.3382   1.180  0.23811
## titleMr.      -2.2455     0.3321  -6.762 1.36e-11 ***
## titleMrs.      0.6109     0.3669   1.665  0.09589 .
## titleNothing  -1.8975     0.5851  -3.243  0.00118 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1390.69  on 1046  degrees of freedom
## Residual deviance:  978.77  on 1040  degrees of freedom
## AIC: 992.77
##
## Number of Fisher Scoring iterations: 4
```

The null deviance shows how well the response is predicted by the model with nothing but an intercept. The residual deviance shows how well the response is predicted by the model when the predictors are included.

From the summary statistics, it can be seen that the Null deviance is 1390.69 and Residual deviance is 978.77. Thus, it can be seen that there is a significant reduction in deviance when the predictor variables title and pclass are included in the model.

Comparing the model glm.title with glm.passengerClass (model fitted using pclass alone as the predictor variable), it can be seen that the the residual deviance has decreased from 1297.6 (using glm.passengerClass) to 978.77(using glm.title).

Also, there is a significant reduction in AIC or Akaike's Information criterion. The model with the lowest AIC is the preferred one. AIC of glm.passengerClass is 1297.6 and that of glm.title is 990.77. Thus glm.title, the model fitted using title and pclass is preferred to glm.passengerClass, model fitted using pclass alone as the predictor variable.

From the summary statistic table it can be observed that titleMr has a very low p-value and hence is hih statistical significane.

```r
# Performing likelihood ratio test
anova(glm.passengerClass, glm.title, test ="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: survived ~ pclass
## Model 2: survived ~ pclass + title
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      1044    1297.42
```

```
## 2      1040     978.77  4   318.65 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

A logistic regression is said to provide a better fit to the data if it demonstrates an improvement over a model with fewer predictors. This is achieved using likelihood ratio test, which compares the likelihood of the data under the full model against the likelihood of the data under a model with fewer predictors. Removing predictors, will result in lower log likelihood, and if this difference is statistically significant, then the null hypothesis can be rejected.

Likelihood ratio test can be performed using anova() from base R. From the summary statistics table, it can be observed that difference in residual deviance between the models glm.passengerClass and gl.title is 318.78. This difference is statistically significant with p-value < 0.05.

Thus, we can conclude that glm.title which uses 2 predictor variables pclass and title performs better than glm.passengerClass which includes only pclass as the predictor variable.

```
# Predicting the survival of passengers from the test data using the fitted
# model
yhat2 <- predict(glm.title, test.Data, type = "response")

# Creating a vector of 0 with length same as the number of rows of test data
glm.pred.title <- rep(0, nrow(test.Data))

# Transformimg to 1(survived) for observations for which predicted probability
# exceeds 0.5
glm.pred.title[yhat2 >.5] <- 1
```

Evaluating the overall fit of this model. The above commands create a vector of class predictions based on whether the predicted probability of a survival is greater than or less than 0.5. The first command creates a vector of 0 with length same as the number of rows of test data. The second line transforms to 1 all of the elements for which the predicted probability of a suvival exceeds 0.5.

Given these predictions, a confusion matrix is created in order to determine how many observations were correctly or incorrectly classified.

```
# Creating a confusion matrix to determine how many observations were
# correctly or incorrectly classified
confmatrix.title <- table(glm.pred.title, test.Data$survived)

# Displaying the confusion matrix
confmatrix.title
```

```
##
## glm.pred.title   0   1
##              0 138  32
##              1  22  70
```

```
# Displaying the prediction accuracy
sum(diag(confmatrix.title))/sum(confmatrix.title)
```

```
## [1] 0.7938931
```

Using the model, glm.title, fitted using pclass and title as predictor variables, it can be observed that 138 passengers who did not survive and 70 passengers who did survive, were correctly predicted by the model. 22 passengers who did not survive were misclassified as survived and 32 passengers who did survive were misclassified as not survived.

On the whole, the fitted model glm.title, as a prediction accuracy of 0.7938931 or 79.389 % . Thus, misclassification rate is 0.2061069 or 20.61 % Also, the title included here are Mr, Mrs, Miss and Master. The rest of the titles are classified as Nothing. This decreases the accuracy of the model because other titles like Rev, Sir etc are also relevant. Grouping all these titles together can affect the prediction accuracy.

Thus, we can see that the mis classification rate of this model, glm.title(20.61 %) is less than the model, glm.passengerClass() which had a mis classification rate of 29.77%

Fitting a random forest model

```
# Preparing the dataset to fit random forest
# Checking the structure of the tittanic.Data dataset to ensure that the predictor
# variables are factor
str(titanic.Data)
```

Using the randomForest function to fit a random forest model with passenger class and title as predictors. Predictions are made for the test set using the random forest model and these predictions are saved as yhat3.

```
## 'data.frame':    1309 obs. of  15 variables:
##  $ pclass   : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ survived : Factor w/ 2 levels "0","1": 2 2 1 1 1 2 2 1 2 1 ...
##  $ name     : Factor w/ 1307 levels "Abbing, Mr. Anthony",..: 22 24 25 26 27 31 46 47 51 55 ...
##  $ sex      : Factor w/ 2 levels "female","male": 1 2 1 2 1 2 1 2 1 2 ...
##  $ age      : num  29 0.917 2 30 25 ...
##  $ sibsp    : int  0 1 1 1 1 0 1 0 2 0 ...
##  $ parch    : int  0 2 2 2 2 0 0 0 0 0 ...
##  $ ticket   : Factor w/ 929 levels "110152","110413",..: 188 50 50 50 50 125 93 16 77 826 ...
##  $ fare     : num  211 152 152 152 152 ...
##  $ cabin    : Factor w/ 187 levels "","A10","A11",..: 45 81 81 81 81 151 147 17 63 1 ...
##  $ embarked : Factor w/ 4 levels "","C","Q","S": 4 4 4 4 4 4 4 4 4 2 ...
##  $ boat     : Factor w/ 28 levels "","1","10","11",..: 13 4 1 1 1 14 3 1 28 1 ...
##  $ body     : int  NA NA NA 135 NA NA NA NA NA 22 ...
##  $ home.dest: Factor w/ 370 levels "","?Havana, Cuba",..: 310 232 232 232 232 238 163 25 23 230 ...
##  $ title    : Factor w/ 5 levels "Master","Miss",..: 2 1 2 3 4 3 2 3 4 3 ...
```

```
titanic.Data$title <- as.ordered(titanic.Data$title)
titanic.Data$sex <- as.ordered(titanic.Data$sex)

# Redefining the training datasset
training.Data <- titanic.Data[train, ]

# Redefining the testing dataset
test.Data <- titanic.Data[-train, ]
```

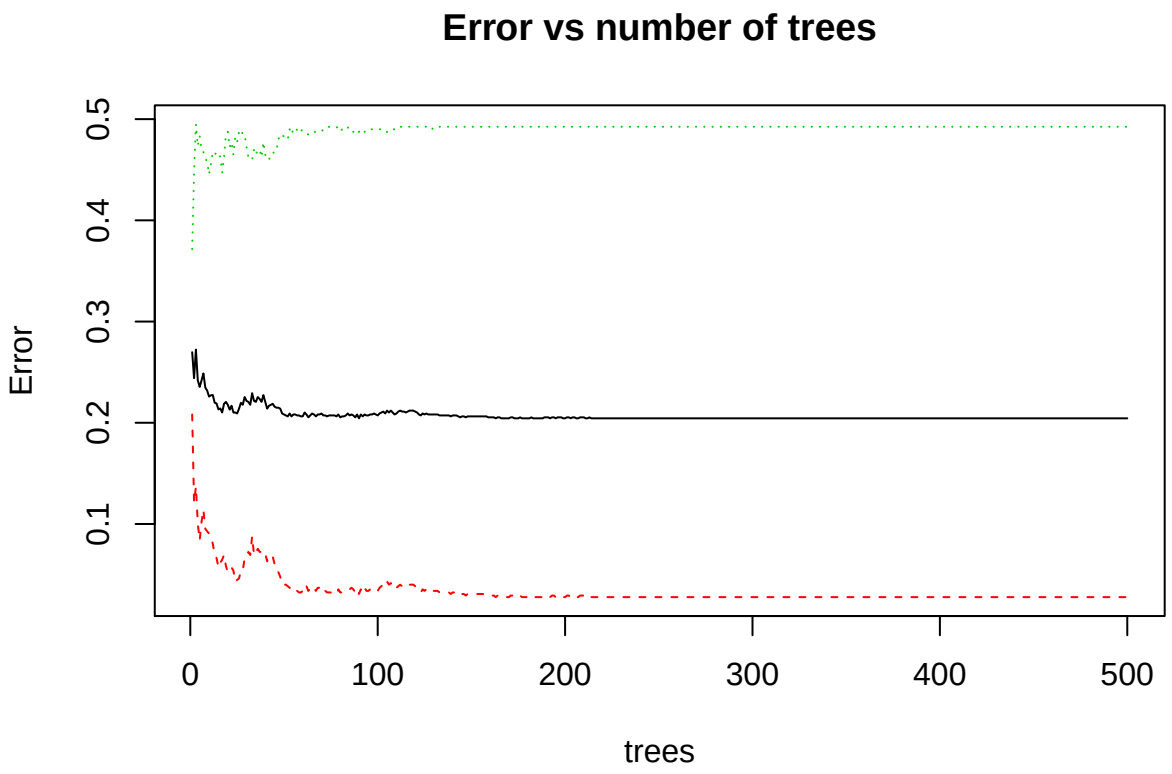
The above step is necessary beacuse in random forest, we are partitioning the dataset feature by feature. Thus its necessary that the predictor variables are in factor mode.

```r
# Seting the seed to reproduce the results
set.seed(1)
# Fitting a random forest using pclass and title as predictor variables
rf.titanic <- randomForest(survived ~ pclass + title,
                           data=training.Data,
                           ntree=500,
                           importance =TRUE)

# Displaying the statistic of the fitted model
rf.titanic
```

```
##
## Call:
##  randomForest(formula = survived ~ pclass + title, data = training.Data,      ntree = 500, importanc
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 20.44%
## Confusion matrix:
##     0   1 class.error
## 0 631  18  0.02773498
## 1 196 202  0.49246231
```

```r
# Plotting the random forest fitted model
plot(rf.titanic, main = "Error vs number of trees")
```



Random forest model is fitted onto the training data. The argument mtry, the number of variables randomly sampled as candidates at each split, is kept as the default value itself(sqrt(p) where p is number of variables in x).

The argument ntree is set to 500. The number was chosen in such a way that it is large enough to stabilize the OOB error. This can be observed in the Error vs number of trees plot.

```
# Predicting the survival rate on the test data using the fitted model
yhat3 = predict(rf.titanic, test.Data)

# Creating a confusion matrix to determine how many observations were
# correctly or incorrectly classified
confmatrix.rf <- table(yhat3, test.Data$survived)

# Displaying the confusion matrix
confmatrix.rf
```

```
##
## yhat3   0   1
##     0 153  50
##     1   7  52
```

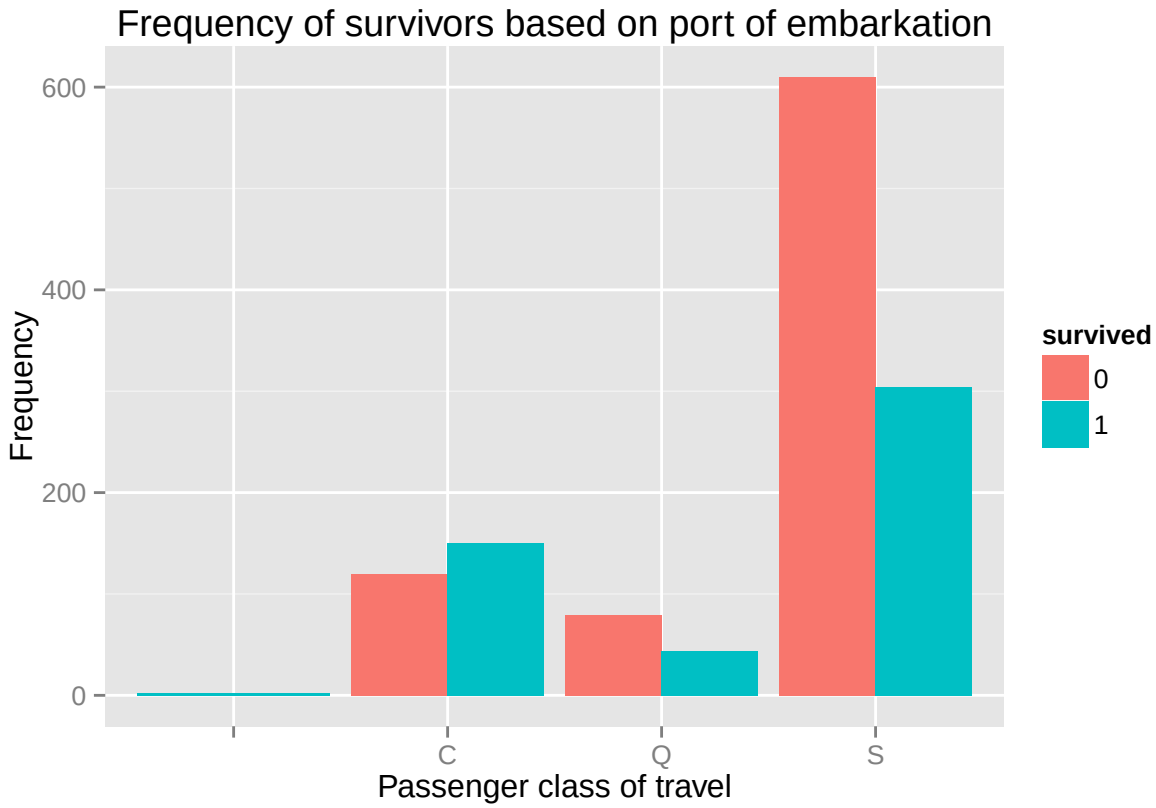
```
# Displaying the prediction accuracy
sum(diag(confmatrix.rf))/sum(confmatrix.rf)
```

```
## [1] 0.7824427
```

Survival rate of the passengers are predicted using the fitted model, rf.titanic. It can be seen that the prediction accuracy of the model is 0.7824427.

```
# Performing exploratory data analysis
# Plotting the frequency of survivors based on embarked port
ggplot(titanic.Data, aes(embarked, fill = survived)) +
        geom_bar(position="dodge") +
  xlab ("Passenger class of travel") +
  ylab("Frequency") +
  ggtitle("Frequency of survivors based on port of embarkation")
```

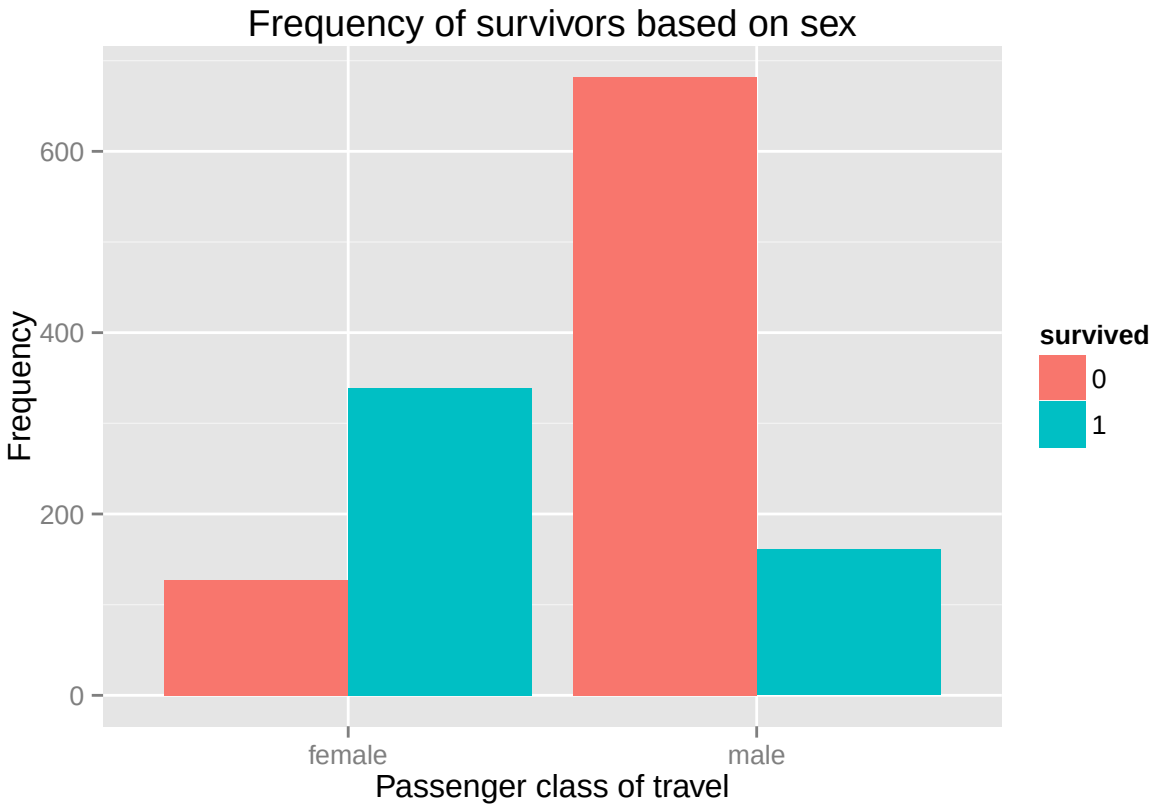
Developing a new random forest model to improve the model performance. Predictions are made for the test set using the new random forest model and these predictions are saved as



yhat4.

From the graph, it can be seen that the frequency of passengers who have survived and not survived is highest at the embarkation port S and lowest at embarkation port Q. Thus the variable embarked can be used to predict if a passenger has survived or not.

```
# Plotting the frequency of survivors based on sex
ggplot(titanic.Data, aes(sex, fill = survived)) +
        geom_bar(position="dodge") +
  xlab ("Passenger class of travel") +
  ylab("Frequency") +
  ggtitle("Frequency of survivors based on sex")
```

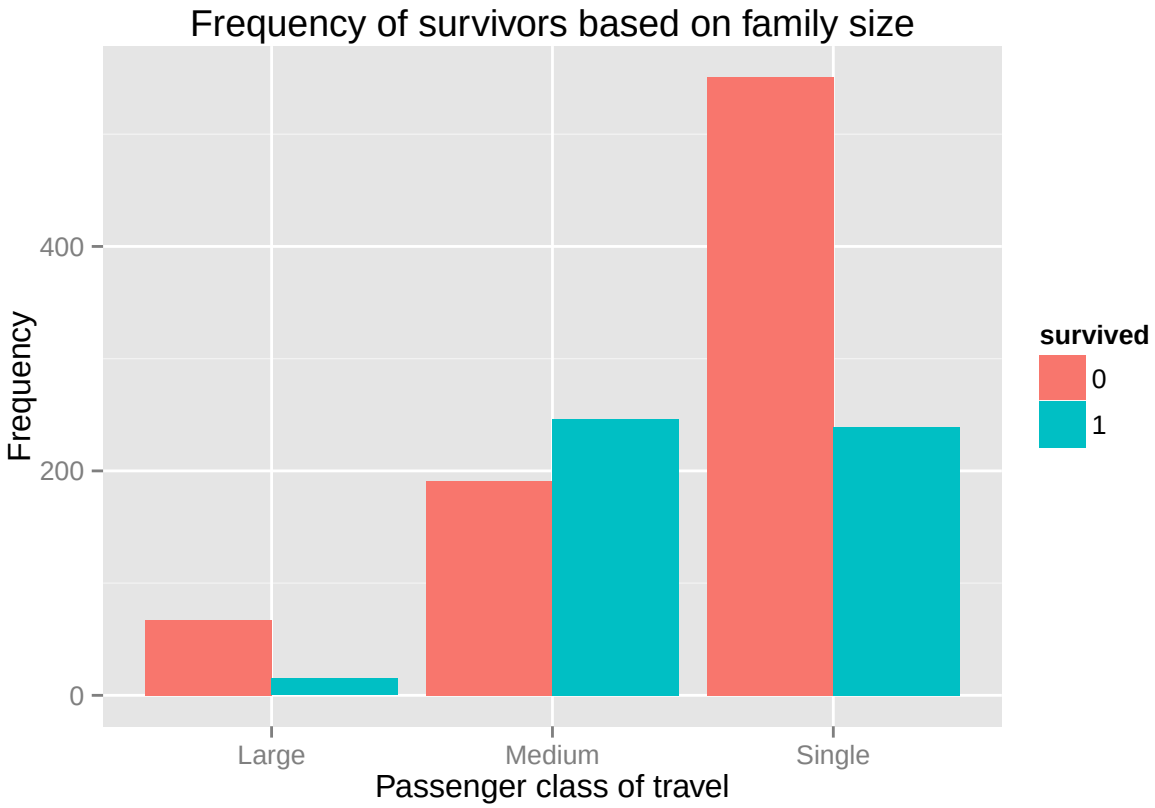


From the graph, it can be seen that the female passengers have a higher survival rate than male passengers. Thus the variable sex can be used to predict if a passenger has survived or not.

```
# Create a new variable Familysize=SibSp+Parch+1(passenger)
titanic.Data$FamilySize <- titanic.Data$sibsp + titanic.Data$parch + 1

# Fixing family category
titanic.Data$FamilyID<-'Large'
titanic.Data$FamilyID[titanic.Data$FamilySize == 1] <- 'Single'
titanic.Data$FamilyID[titanic.Data$FamilySize >= 2 &
                        titanic.Data$FamilySize <= 4] <- 'Medium'
titanic.Data$FamilyID <- as.factor(titanic.Data$FamilyID)

ggplot(titanic.Data, aes(FamilyID, fill = survived)) +
         geom_bar(position="dodge") +
  xlab ("Passenger class of travel") +
  ylab("Frequency") +
  ggtitle("Frequency of survivors based on family size")
```



From the graph, it can be seen that the single passengers have the highest non survival rate compared to medium and large families. Thus the variable FamilyID can be used to predict if a passenger has survived or not.

Based on the above analysis, a random forest model is fitted by using the predcitor variables embarked, sex and FamilyID in addition to title.

```
# Redefining the training datasset to include new coulmns
training.Data <- titanic.Data[train, ]

# Redefining the testing dataset to include new coulmns
test.Data <- titanic.Data[-train, ]

# Fitting a random forest using title,sex and embarked as predictor variables
rf.titanic.new <- randomForest(survived ~ title + sex + embarked +
                                 FamilyID,
                               data=training.Data,
                               importance =TRUE)

# Displaying the statistic of the fitted model
rf.titanic.new
```

```
##
## Call:
##  randomForest(formula = survived ~ title + sex + embarked + FamilyID,      data = training.Data, imp
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
```

```
##         OOB estimate of  error rate: 19.01%
## Confusion matrix:
##     0   1 class.error
## 0 563  86   0.1325116
## 1 113 285   0.2839196
```

```r
# Predicting the survival rate on the test data using the fitted model
yhat4 = predict(rf.titanic.new ,test.Data)

# Creating a confusion matrix to determine how many observations were
# correctly or incorrectly classified
confmatrix.rf.new <- table(yhat4, test.Data$survived)

# Displaying the confusion matrix
confmatrix.rf.new
```

```
##
## yhat4   0   1
##     0 139  29
##     1  21  73
```

```r
# Displaying the prediction accuracy
sum(diag(confmatrix.rf.new))/sum(confmatrix.rf.new)
```

```
## [1] 0.8091603
```

Predictions with the new random forest model are stored into the variable yhat4. From the confusion matrix, it can be observed that the model has a prediction accuracy of 0.8091603 or 80.91 %

Comparing the accuracy of each of the models from this problem set using

```r
# Dividing the canvas for plotting
par(mfrow = c(2,2))

# Building a roc curve for the logistic regression model using pclass as
# predictors
roc.class <- roc(test.Data$survived, glm.pred)
# Plotting the roc curve
plot(roc.class, main = "Log classification using pclass")
```

ROC curves.

```
##
## Call:
## roc.default(response = test.Data$survived, predictor = glm.pred)
##
## Data: glm.pred in 160 controls (test.Data$survived 0) < 102 cases (test.Data$survived 1).
## Area under the curve: 0.6585
```

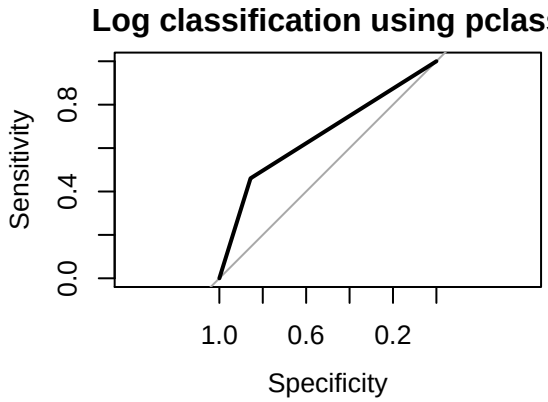
```r
# Building a roc curve for the logistic regression model using pclass and
# title as predictors
roc.title <- roc(test.Data$survived, glm.pred.title)
# Plotting the roc curve
plot(roc.title, main = "Log classification using pclass, title")
```

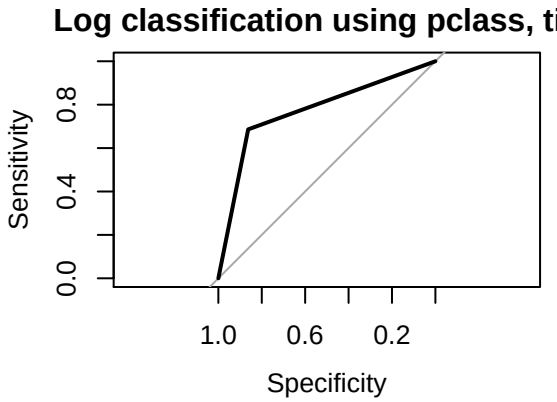
```
##
## Call:
## roc.default(response = test.Data$survived, predictor = glm.pred.title)
##
## Data: glm.pred.title in 160 controls (test.Data$survived 0) < 102 cases (test.Data$survived 1).
## Area under the curve: 0.7744
```

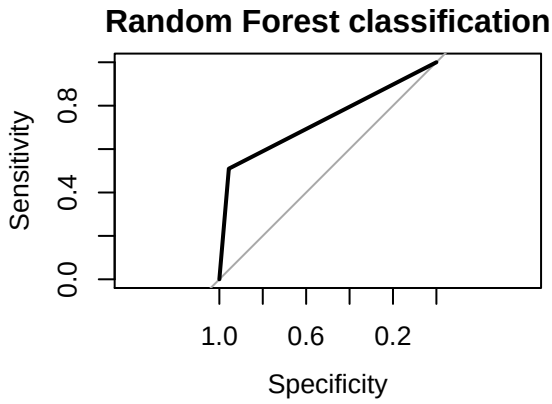
```r
# Ordering the predicted values by the model rf.titanic
yhat3 <- as.ordered(yhat3)
# Building a roc curve for the random forest model using pclass and title
# as predictors
roc.rf <- roc(test.Data$survived, yhat3)
# Plotting the roc curve
plot(roc.rf, main = "Random Forest classification")
```

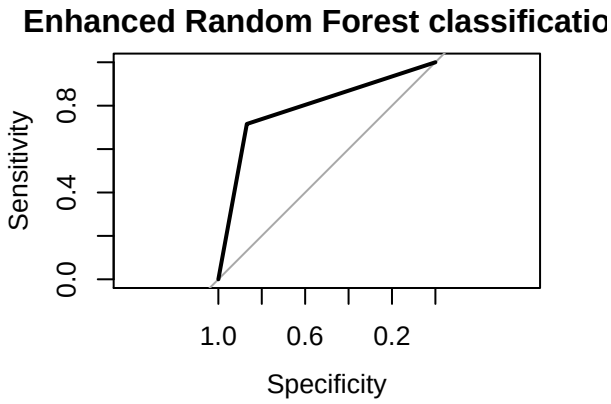
```
##
## Call:
## roc.default(response = test.Data$survived, predictor = yhat3)
##
## Data: yhat3 in 160 controls (test.Data$survived 0) < 102 cases (test.Data$survived 1).
## Area under the curve: 0.733
```

```r
# Ordering the predicted values by the model rf.titanic.new
yhat4 <- as.ordered(yhat4)
# Building a roc curve for the random forest model using pclass,title,sex and
# embarked as predictors
roc.rf.new <- roc(test.Data$survived, yhat4)
# Plotting the roc curve
plot(roc.rf.new, main = "Enhanced Random Forest classification")
```










```
##
## Call:
## roc.default(response = test.Data$survived, predictor = yhat4)
##
## Data: yhat4 in 160 controls (test.Data$survived 0) < 102 cases (test.Data$survived 1).
## Area under the curve: 0.7922
```

```
# Displaying the roc objects to observe area under the curve
roc.class
```

```
##
## Call:
## roc.default(response = test.Data$survived, predictor = glm.pred)
##
## Data: glm.pred in 160 controls (test.Data$survived 0) < 102 cases (test.Data$survived 1).
## Area under the curve: 0.6585
```

```
roc.title
```

```
##
## Call:
## roc.default(response = test.Data$survived, predictor = glm.pred.title)
##
## Data: glm.pred.title in 160 controls (test.Data$survived 0) < 102 cases (test.Data$survived 1).
## Area under the curve: 0.7744
```

```
roc.rf
```

```
##
## Call:
## roc.default(response = test.Data$survived, predictor = yhat3)
##
## Data: yhat3 in 160 controls (test.Data$survived 0) < 102 cases (test.Data$survived 1).
## Area under the curve: 0.733
```

```
roc.rf.new
```

```
##
## Call:
## roc.default(response = test.Data$survived, predictor = yhat4)
##
## Data: yhat4 in 160 controls (test.Data$survived 0) < 102 cases (test.Data$survived 1).
## Area under the curve: 0.7922
```

```
# Displaying the predictio accuracy of all the models
sum(diag(confmatrix.pclass))/sum(confmatrix.pclass)
```

```
## [1] 0.7022901
```

```
sum(diag(confmatrix.title))/sum(confmatrix.title)
```

```
## [1] 0.7938931
```

```
sum(diag(confmatrix.rf))/sum(confmatrix.rf)
```

```
## [1] 0.7824427
```

```
sum(diag(confmatrix.rf.new))/sum(confmatrix.rf.new)
```

```
## [1] 0.8091603
```

The overall performance of a classifier, is given by Area under the curve. It takes value between 0.5 and 1. The model with bigger AUC has better perfomance.

AUC for logistic regression model using predictor variable pclass: 0.6585

AUC for logistic regression model using predictor variable pclass and title : 0.7744

AUC for the random forest model using predictor variable pclass and title: 0.733

AUC for the random forest model using predictor variable title,sex, embarked and FamilyID: 0.7922

Thus, the model with biggest AUC is the random forest model using predictor variables title, sex, embarked and FamilyID

Comparing the Prediction accuracy of the different models

Prediction accuracy for logistic regression model using predictor variable pclass: 0.7022901

Prediction accuracy for logistic regression model using predictor variable pclass and title : 0.7938931

Prediction accuracy for the random forest model using predictor variable pclass and title: 0.7824427

Prediction accuracy for the random forest model using predictor variable title,sex, embarked and FamilyID: 0.8091603

The highest prediction accuracy is also for the random forest model using
predictor variable title,sex, embarked and FamilyID

Thus, it can be concluded that based on the area under the ROC curve and the prediction accuracy, the model that works best for predicting survival of the Titanic passengers is the random forest model using predictor variable title,sex, embarked and FamilyID