Project - Titanic Survival Prediction

Edwin Leonardi Liong

January 8, 2019

Introduction

It has been over a century since the RMS Titanic sank on its maiden voyage in 1912. The ship was carrying estimated 2224 passengers, but only less than 800 passengers survived the tragic accident. For many decades, the tragic story has been gaining attention from researchers and media all over the world. A blockbuster movie in 1997 also helped catapulted people interest.

In this project, we will attempt to predict the probability of a passenger surviving from the disaster given a set of attributes of the passenger. This problem is categorised as a classification, that is when the outcome target is a category and in this instance, is either survived or not survived. This project is a great choice for learning experience and analysis due to the small size of dataset and relatable variables.

Later on, we are going to apply supervised machine learning algorithms that work best for answering the classification question. Throughout this journey, we may come across interesting results that allow us to make few insights regarding the data.

Dataset - Exploration and Preparation

Titanic data is publicly available for download. First we will go through the dataset to study the structure and its completeness. At the same time, we check which attribute(s) that can be used as the predictor and which may need to be removed from the dataset.

Let's start by importing the required libraries and downloading the data

```
#importing libraries
library(tidyverse)
library(scales)
library(caret)
library(dplyr)
library(purrr)
library(Hmisc)
library(rpart)
library(rpart.plot)
library(ggplot2)
library(ROCR)
library(knitr)
library(e1071)

set.seed(1)
```

```
#downloading titanic full data set from url and assign empty string or blank
values to NA
titanic_data <-
read.csv("http://biostat.mc.vanderbilt.edu/wiki/pub/Main/DataSets/titanic3.cs
v",na.strings=c("","NA"))
```

We then run this script and find that the dataset contains 1309 passenger details along with the 14 variables

```
#observe the structure of the data and its content
head(titanic_data)
```

```
##   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 29.00     0     0  24160 211.3375      B5        S    2   NA
## 2  0.92     1     2 113781 151.5500 C22 C26        S   11   NA
## 3  2.00     1     2 113781 151.5500 C22 C26        S <NA>   NA
## 4 30.00     1     2 113781 151.5500 C22 C26        S <NA>  135
## 5 25.00     1     2 113781 151.5500 C22 C26        S <NA>   NA
## 6 48.00     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
```

```
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 0.92 2 30 25 48 63 39 53 71 ...
##  $ 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/ 186 levels "A10","A11","A14",..: 44 80 80 80 80
150 146 16 62 NA ...
```

```
##  $ embarked : Factor w/ 3 levels "C","Q","S": 3 3 3 3 3 3 3 3 3 1 ...
##  $ boat     : Factor w/ 27 levels "1","10","11",..: 12 3 NA NA NA 13 2 NA
27 NA ...
##  $ body     : int  NA NA NA 135 NA NA NA NA NA 22 ...
##  $ home.dest: Factor w/ 369 levels "?Havana, Cuba",..: 309 231 231 231 231
237 162 24 22 229 ...
```

```
describe(titanic_data)
```

```
## titanic_data
##
##  14  Variables      1309  Observations
## ---------------------------------------------------------------------------
-
## pclass
##        n  missing distinct     Info     Mean      Gmd
##     1309        0        3    0.817    2.295   0.8689
##
## Value          1     2     3
## Frequency    323   277   709
## Proportion 0.247 0.212 0.542
## ---------------------------------------------------------------------------
-
## survived
##        n  missing distinct     Info      Sum     Mean      Gmd
##     1309        0        2    0.708      500    0.382   0.4725
##
## ---------------------------------------------------------------------------
-
## name
##        n  missing distinct
##     1309        0     1307
##
## lowest : Abbing, Mr. Anthony                  Abbott, Master. Eugene Joseph
Abbott, Mr. Rossmore Edward      Abbott, Mrs. Stanton (Rosa Hunt) Abelseth,
Miss. Karen Marie
## highest: Zabour, Miss. Hileni                 Zabour, Miss. Thamine
Zakarian, Mr. Mapriededer        Zakarian, Mr. Ortin                  Zimmerman,
Mr. Leo
## ---------------------------------------------------------------------------
-
## sex
##        n  missing distinct
##     1309        0        2
##
## Value      female   male
## Frequency     466    843
## Proportion  0.356  0.644
## ---------------------------------------------------------------------------
-
```

```
## age
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     1046      263       98    0.999    29.88    16.06        5       14
##      .25      .50      .75      .90      .95
##       21       28       39       50       57
##
## lowest :  0.17  0.33  0.42  0.67  0.75, highest: 70.50 71.00 74.00 76.00
80.00
## ------------------------------------------------------------------------
-
## sibsp
##        n  missing distinct     Info     Mean      Gmd
##     1309        0        7     0.67   0.4989    0.777
##
## Value          0     1     2     3     4     5     8
## Frequency    891   319    42    20    22     6     9
## Proportion 0.681 0.244 0.032 0.015 0.017 0.005 0.007
## ------------------------------------------------------------------------
-
## parch
##        n  missing distinct     Info     Mean      Gmd
##     1309        0        8    0.549    0.385   0.6375
##
## Value          0     1     2     3     4     5     6     9
## Frequency   1002   170   113     8     6     6     2     2
## Proportion 0.765 0.130 0.086 0.006 0.005 0.005 0.002 0.002
## ------------------------------------------------------------------------
-
## ticket
##        n  missing distinct
##     1309        0      929
##
## lowest : 110152      110413      110465      110469      110489
## highest: W./C. 6608  W./C. 6609  W.E.P. 5734 W/C 14208   WE/P 5735
## ------------------------------------------------------------------------
-
## fare
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     1308        1      281        1     33.3    38.61    7.225    7.568
##      .25      .50      .75      .90      .95
##    7.896   14.454   31.275   78.051  133.650
##
## lowest :   0.0000   3.1708   4.0125   5.0000   6.2375
## highest: 227.5250 247.5208 262.3750 263.0000 512.3292
## ------------------------------------------------------------------------
-
## cabin
##        n  missing distinct
##      295     1014      186
##
```

```
## lowest : A10 A11 A14 A16 A18, highest: F33 F38 F4  G6  T
## ---------------------------------------------------------------------------
-
## embarked
##        n  missing distinct
##     1307        2        3
##
## Value          C     Q     S
## Frequency    270   123   914
## Proportion 0.207 0.094 0.699
## ---------------------------------------------------------------------------
-
## boat
##        n  missing distinct
##      486      823       27
##
## lowest : 1   10  11  12  13 , highest: A   B   C   C D D
## ---------------------------------------------------------------------------
-
## body
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      121     1188      121        1    160.8      113       16       35
##      .25      .50      .75      .90      .95
##       72      155      256      297      307
##
## lowest :   1   4   7   9  14, highest: 312 314 322 327 328
## ---------------------------------------------------------------------------
-
## home.dest
##        n  missing distinct
##      745      564      369
##
## lowest : ?Havana, Cuba                      Aberdeen / Portland, OR
Albany, NY                         Altdorf, Switzerland
Amenia, ND
## highest: Worcester, England                 Worcester, MA
Yoevil, England / Cottage Grove, OR Youngstown, OH
Zurich, Switzerland
## ---------------------------------------------------------------------------
-
```

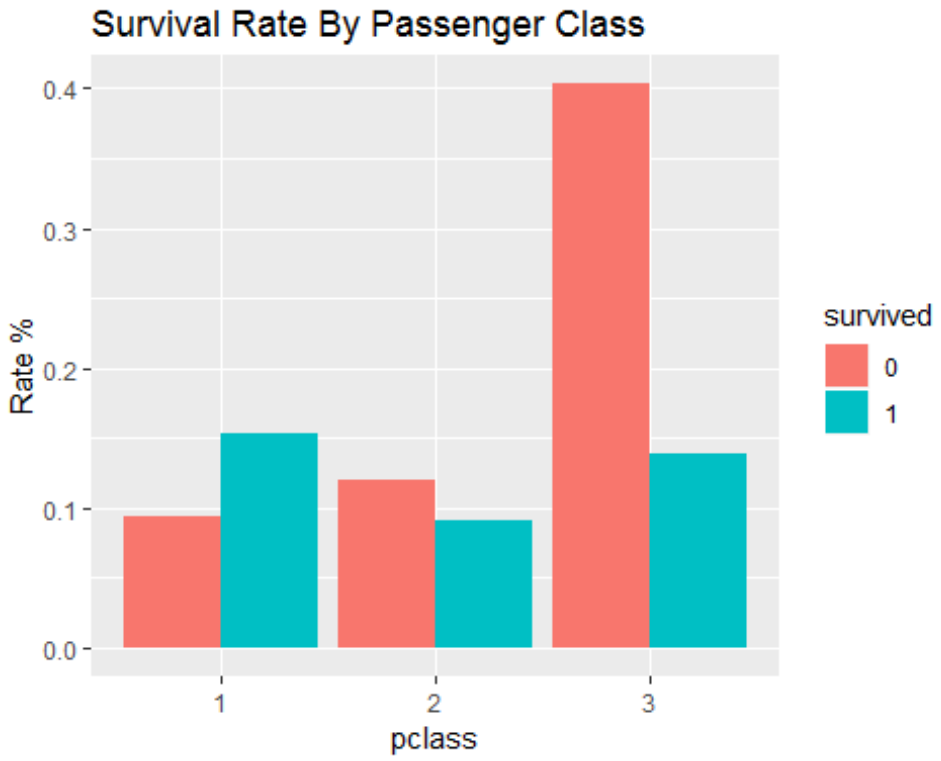
The variables are analyzed as below

survived is the survival outcome of the passenger and it is also defined as our dependent variable (prediction). The mean value shows that the survival rate was 38%. No missing values detected. With just two possible values, the variable is converted to a factor

```
titanic_data$survived<-as.factor(titanic_data$survived)
```

pclass has 3 distinct values: first, second and third class. No missing values detected. This is to be treated as categorical attribute, hence the conversion to data type factor. On the plot below, we can see that 3rd class passengers are in a higher proportion for not to survive compared to the other classes

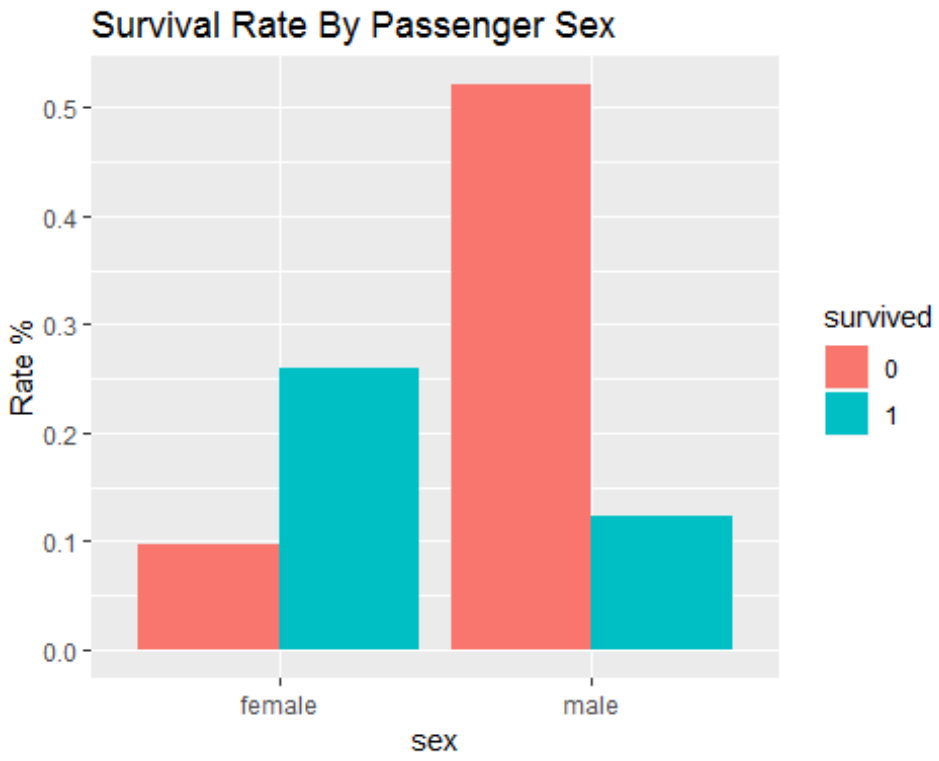
```
titanic_data$pclass<-as.factor(titanic_data$pclass)
titanic_data %>% ggplot(aes(pclass,fill=survived)) + geom_bar(aes(y =
(..count..)/sum(..count..)), position="dodge") + ylab("Rate %") +
ggtitle("Survival Rate By Passenger Class")
```



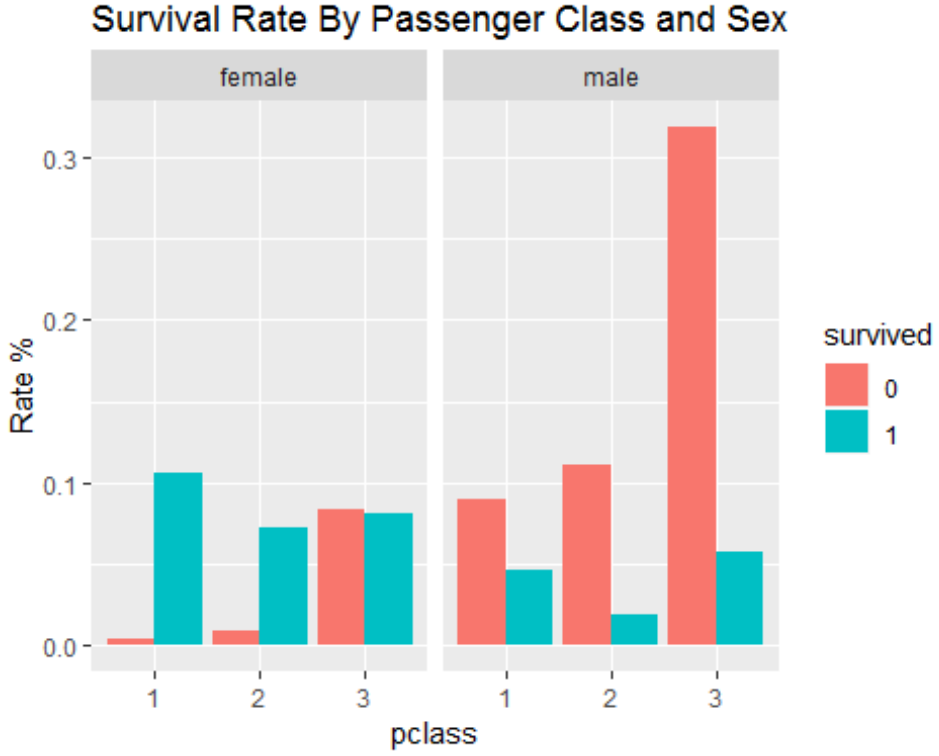
name attribute has 1307 distinct values. It does make sense that name should not have any influence towards one survival. We can drop the name attribute for now

sex attribute is either male or female. No missing value detected. Gender should be in a factor type.

```
titanic_data$sex<-as.factor(titanic_data$sex)
titanic_data %>% ggplot(aes(sex,fill=survived)) + geom_bar(aes(y =
(..count..)/sum(..count..)), position="dodge") + ylab("Rate %") +
ggtitle("Survival Rate By Passenger Sex")
```



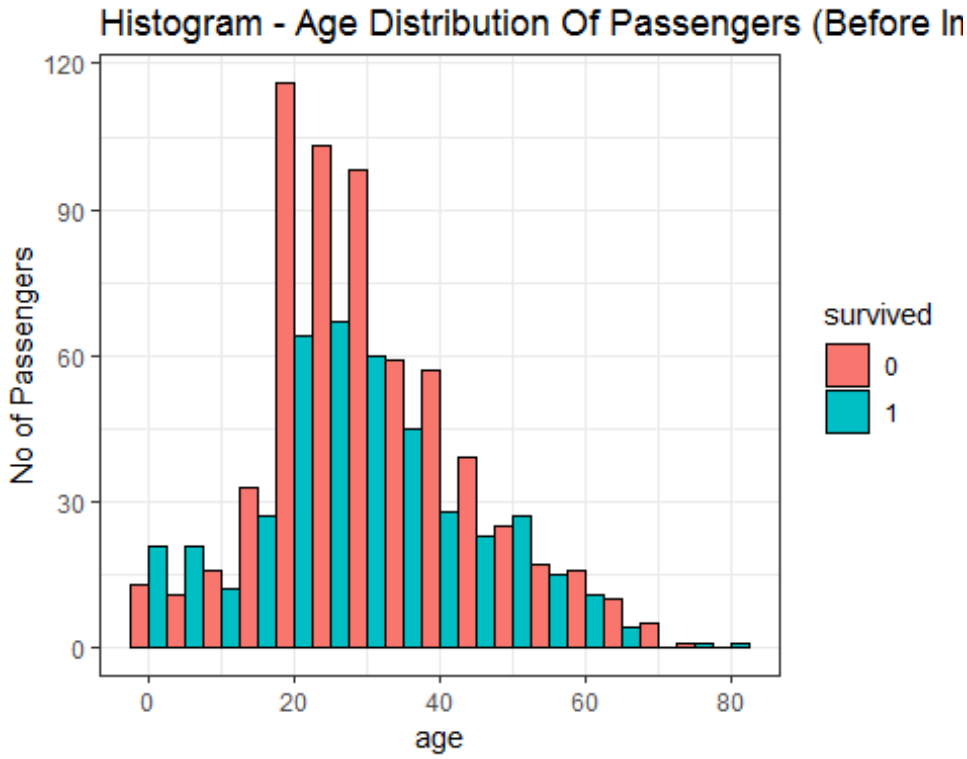

```
titanic_data %>% ggplot(aes(pclass,fill=survived)) + geom_bar(aes(y =
(..count..)/sum(..count..)),   position="dodge") + facet_wrap(~sex)
+ylab("Rate %") +  ggtitle("Survival Rate By Passenger Class and Sex")
```



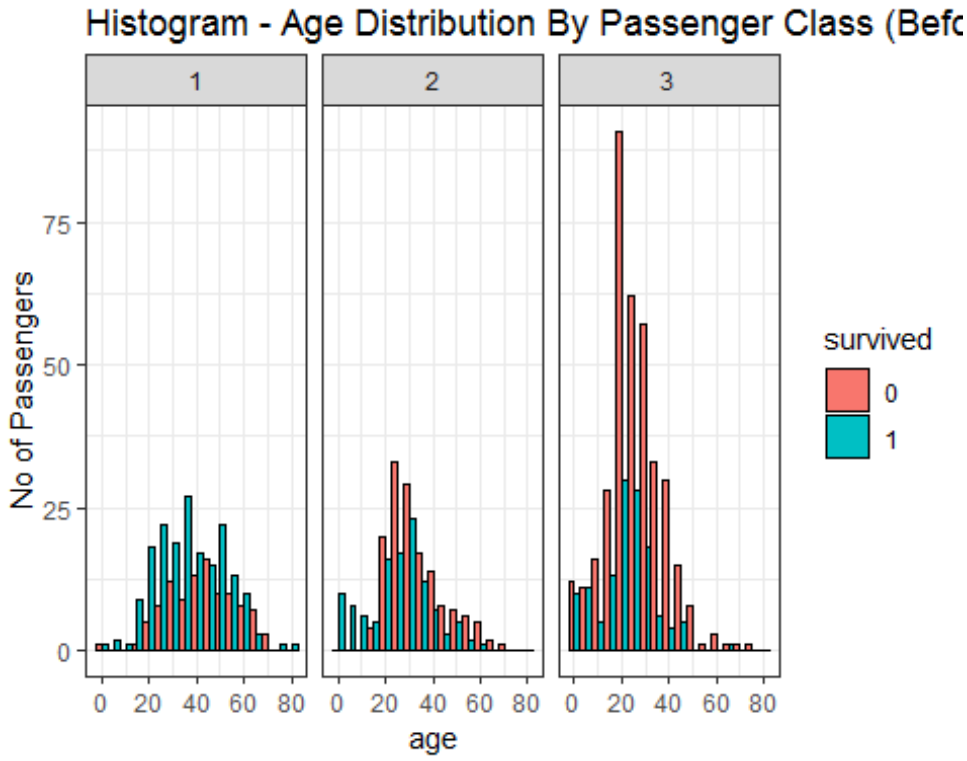
60% passengers are male. Despite this, male survival rate is not as good as the female counterpart. When sliced further by passenger class, it appears that 3rd class male passengers have the worst odds for survival. Also note that 3rd class female passengers are less likely to survive compared to being in 1st or 2nd class. This is in line with our quick analysis earlier on the pclass plot

age does have a quite number of missing values. Though so, it is still worth keeping for the model. The missing values will be populated using the mean of the passengers within the same pclass. Age statistics before imputation is shown in the distribution below

```
titanic_data %>% ggplot(aes(age, fill = survived))+
    geom_histogram(binwidth = 5, colour = "black", position =
"dodge",alpha=1)+
    theme_bw()+ggtitle("Histogram - Age Distribution Of Passengers (Before
Imputation)") +ylab("No of Passengers")
```

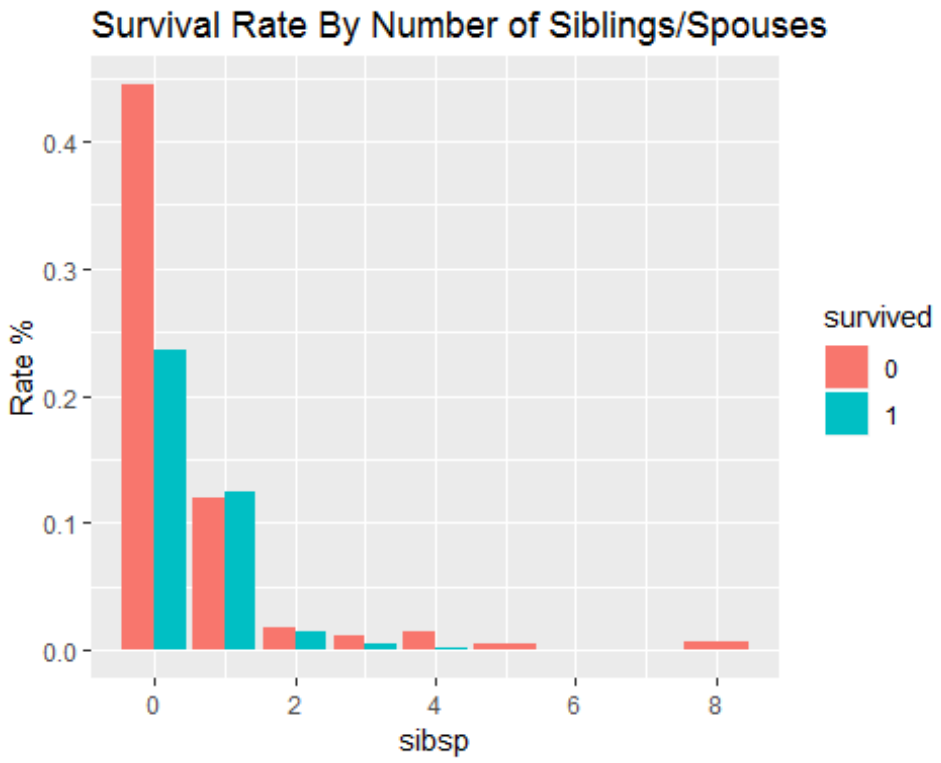



```
titanic_data %>% ggplot(aes(age, fill = survived))+
    geom_histogram(binwidth = 5, colour = "black", position =
"dodge",alpha=1)+
    theme_bw()+ ggtitle("Histogram - Age Distribution By Passenger Class
(Before Imputation)") +facet_wrap(~pclass) +ylab("No of Passengers")
```



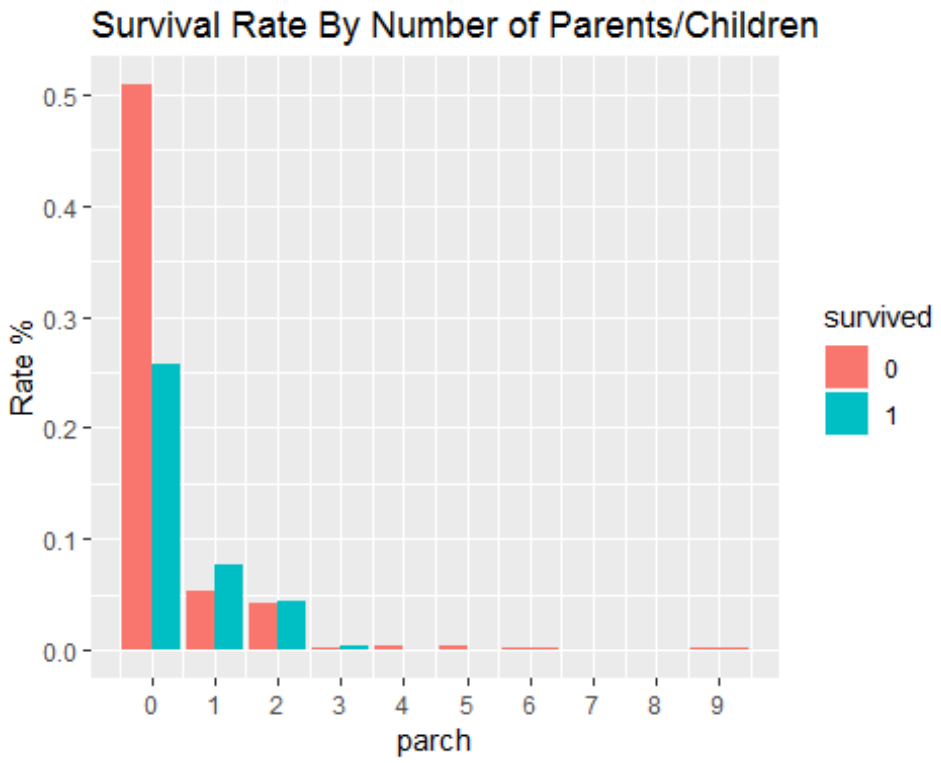
sibsp attribute indicates the number of siblings/spouses aboard the ship. No missing values detected so the feature is good to use.

```
titanic_data %>% ggplot(aes(sibsp,fill=survived)) + geom_bar(aes(y =
(..count..)/sum(..count..)), position="dodge") + ylab("Rate %") +
ggtitle("Survival Rate By Number of Siblings/Spouses")
```



parch refers to the number of parent/children aboard the ship. No missing values detected so the feature is good to use.

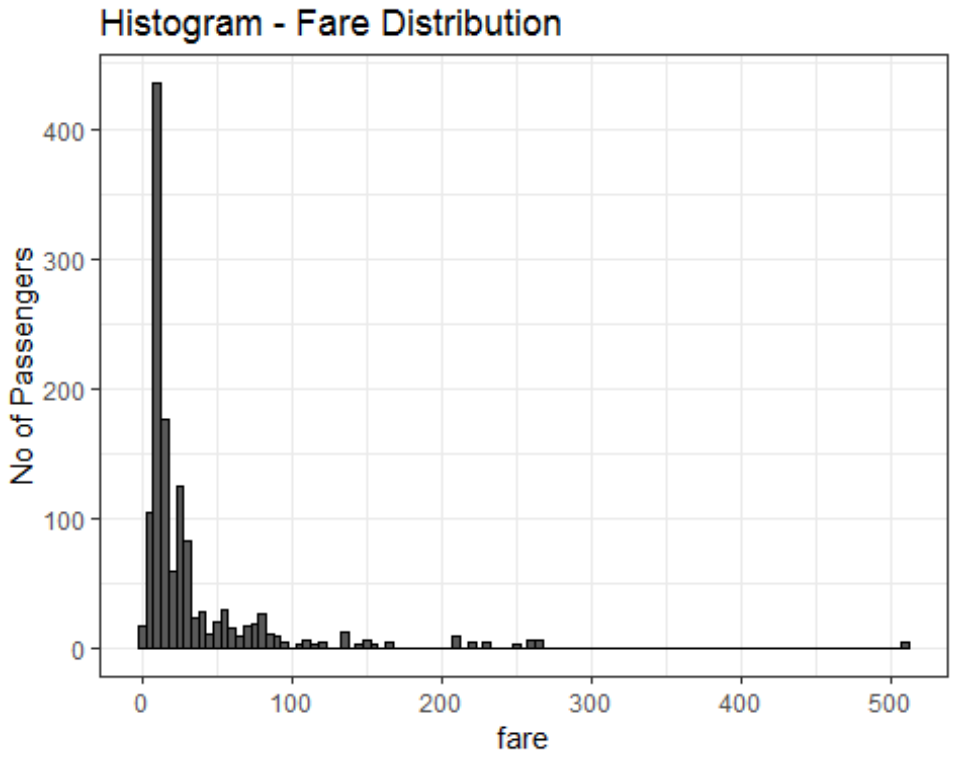
```
titanic_data %>% ggplot(aes(parch,fill=survived)) + geom_bar(aes(y =
(..count..)/sum(..count..)), position="dodge") + ylab("Rate %") +
ggtitle("Survival Rate By Number of Parents/Children") +
scale_x_continuous(breaks = scales::pretty_breaks(15))
```



ticket is one of the categorical attributes, however since it has too many unique values, this may not serve much purpose. We will leave this one off as a predictor

fare attribute can be a good feature to include. With 1 missing value, this can be tacked in similar fashion as the age imputation.

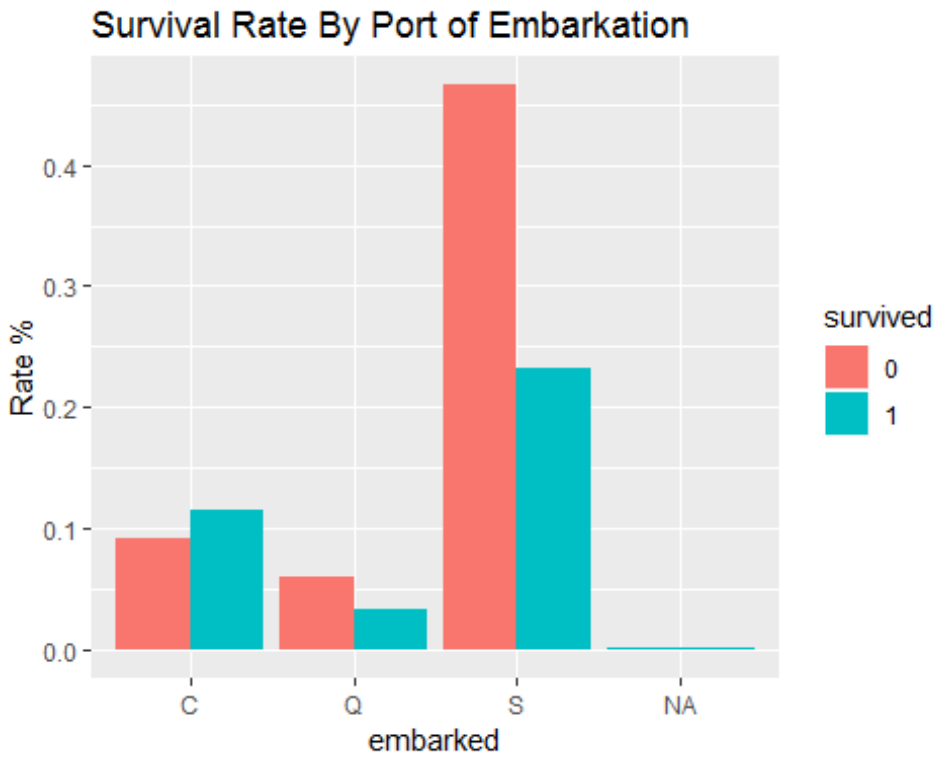
```
titanic_data %>% ggplot(aes(fare))+ geom_histogram(binwidth = 5, colour =
"black", position = "dodge",alpha=1)+
    theme_bw()+ ggtitle("Histogram - Fare Distribution") +ylab("No of
Passengers")
```



cabin variable has many missing values. It is reasonable not to omit the rows from the dataset and also imputation strategy will not be a good idea. Thus, cabin needs to be dropped from the dataset

We can potentially include **embarked** attribute which is the port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton). But before, the 2 missing values will be populated using the mode, which should be a fair and quick solution. In a glance, we can't deduce if the embarkation port is a significant factor to one's survival. **boat**, **body** and **home.dest** can be removed from the dataset mainly because of the high number of missing values

```
titanic_data$embarked<-as.factor(titanic_data$embarked)
titanic_data %>% ggplot(aes(embarked,fill=survived)) + geom_bar(aes(y =
(..count..)/sum(..count..)), position="dodge") +  ylab("Rate %") +
ggtitle("Survival Rate By Port of Embarkation")
```



The following is the script along with comments to highlight the data preparation and imputation exercise

```
#get the mean values of age and fare by passenger class and store them as
data frames
titanic_class_age_mean <- titanic_data %>% filter(!is.na(age)) %>%
group_by(pclass) %>% dplyr::summarise(class_age_mean=mean(age)) %>%
as.data.frame(.)
titanic_class_fare_mean <- titanic_data %>% filter(!is.na(fare)) %>%
group_by(pclass) %>% dplyr::summarise(class_fare_mean=mean(fare)) %>%
as.data.frame(.)

#function to calculate mode value
Mode <- function(x) {
u <- unique(x)
u[which.max(tabulate(match(x, u)))]
}

#get mode value of embarked
mode_embarked <- Mode(titanic_data$embarked)

#join data frames obtained with the main titanic_data set
titanic_data <- inner_join(titanic_data,titanic_class_age_mean,by="pclass")
titanic_data <- inner_join(titanic_data,titanic_class_fare_mean,by="pclass")

#replace the age, embarked and fare values with the logical condition for
```

```
imputation
titanic_data <- mutate(titanic_data,age = if_else(is.na(age), class_age_mean,
age), fare=if_else(is.na(fare), class_fare_mean, fare))
titanic_data <- mutate(titanic_data,embarked =
if_else(is.na(embarked),mode_embarked,embarked))

#drop attributes that will not be used in the modelling
titanic_data<-titanic_data%>%select(-name,-class_age_mean,-class_fare_mean,-
boat,-ticket,-body,-cabin,-home.dest)
```

Run the describe() again, and this time we are down with 8 variables and no missing values detected. All is ready to go. Note that the age mean before and after imputation is rouglhy similar 29.88 and 29.35 respectively.

```
describe(titanic_data)
```

```
## titanic_data
##
##  8  Variables      1309  Observations
## ---------------------------------------------------------------------------
-
## pclass
##        n  missing distinct
##     1309        0        3
##
## Value          1     2     3
## Frequency    323   277   709
## Proportion 0.247 0.212 0.542
## ---------------------------------------------------------------------------
-
## survived
##        n  missing distinct
##     1309        0        2
##
## Value          0     1
## Frequency    809   500
## Proportion 0.618 0.382
## ---------------------------------------------------------------------------
-
## sex
##        n  missing distinct
##     1309        0        2
##
## Value      female   male
## Frequency     466    843
## Proportion  0.356  0.644
## ---------------------------------------------------------------------------
-
## age
##        n  missing distinct     Info     Mean      Gmd      .05      .10
```

```
##     1309        0      101    0.996    29.35    14.24      7.4     16.0
##      .25      .50      .75      .90      .95
##     22.0     26.0     37.0     48.0     55.0
##
## lowest :  0.17  0.33  0.42  0.67  0.75, highest: 70.50 71.00 74.00 76.00
80.00
## ------------------------------------------------------------------------
-
## sibsp
##        n  missing distinct     Info     Mean      Gmd
##     1309        0        7     0.67   0.4989    0.777
##
## Value          0     1     2     3     4     5     8
## Frequency    891   319    42    20    22     6     9
## Proportion 0.681 0.244 0.032 0.015 0.017 0.005 0.007
## ------------------------------------------------------------------------
-
## parch
##        n  missing distinct     Info     Mean      Gmd
##     1309        0        8    0.549    0.385   0.6375
##
## Value          0     1     2     3     4     5     6     9
## Frequency   1002   170   113     8     6     6     2     2
## Proportion 0.765 0.130 0.086 0.006 0.005 0.005 0.002 0.002
## ------------------------------------------------------------------------
-
## fare
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     1309        0      282        1    33.28    38.59    7.225    7.570
##      .25      .50      .75      .90      .95
##    7.896   14.454   31.275   78.020  133.650
##
## lowest :   0.0000   3.1708   4.0125   5.0000   6.2375
## highest: 227.5250 247.5208 262.3750 263.0000 512.3292
## ------------------------------------------------------------------------
-
## embarked
##        n  missing distinct
##     1309        0        3
##
## Value          C     Q     S
## Frequency    270   123   916
## Proportion 0.206 0.094 0.700
## ------------------------------------------------------------------------
-
```

Our next step is to split the full dataset into training and test set. 20% ratio is chosen for the test

```r
#split data into train and test dataset
test_index<-
createDataPartition(y=titanic_data$survived,times=1,p=0.2,list=FALSE)
titanic_train<-titanic_data[-test_index,]
titanic_test<-titanic_data[test_index,]

#check the result of the split
dim(titanic_train)

## [1] 1047    8

dim(titanic_test)

## [1] 262   8
```

Modeling Approach and Result

In this section, we are going to create and discuss about the prediction modelling based on supervised learning technique. First, we will implement the logistic regression and then construct decision tree algorithm for comparison

1. Binomial Logistic Regression

Logistic Regression is often used for prediction analysis where the dependent variable is of a categorical type. The outcome of Titanic prediction is also a category makes this a fit candidate for logistic regression. Logistic regression can be binomial or multinomial.

In logistic regression, prediction is made with the use of probability. In the Titanic scenario, we agree that straightforward outcomes are:

- 0 = 0% the passenger would not survive
- 1 = 100% the passenger would survive

However, any value between 0-1 denotes a certain degree of confidence on the survival. Therefore, we can formulate in such a way that if model output is above than 0.5, we will categorize it as 'survived'. Anything below 0.5, we treat as 'not survived'. This then reinforces our regression to be a binomial.

```r
#1st Model - Binomial Logistic Regression
glm_fit <- titanic_train %>% glm(survived ~ ., data=., family = "binomial")
summary(glm_fit)

##
## Call:
## glm(formula = survived ~ ., family = "binomial", data = .)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6498  -0.6491  -0.4413   0.6625   2.5629
##
## Coefficients:
```

```
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.054155   0.442251   9.167  < 2e-16 ***
## pclass2     -1.114169   0.276568  -4.029 5.61e-05 ***
## pclass3     -2.167968   0.281389  -7.705 1.31e-14 ***
## sexmale     -2.537924   0.177505 -14.298  < 2e-16 ***
## age         -0.042700   0.007437  -5.742 9.37e-09 ***
## sibsp       -0.332629   0.101882  -3.265   0.0011 **
## parch       -0.003613   0.097092  -0.037   0.9703
## fare         0.001589   0.001994   0.797   0.4255
## embarkedQ   -0.274096   0.338884  -0.809   0.4186
## embarkedS   -0.389389   0.211053  -1.845   0.0650 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1392.63  on 1046  degrees of freedom
## Residual deviance:  963.25  on 1037  degrees of freedom
## AIC: 983.25
##
## Number of Fisher Scoring iterations: 5
```

We can see that variables pclass, sex, age, sibsp are statistically significant. Sex is shown to have strong correlation with survival. If the passenger is male, then the odds reduced by 2.53. A unit increase in age also reduces the odds by 0.043. Not surprisingly being a 2nd or 3rd class passenger lowers the odds significantly. Interestingly, we find a little bit of a factor of port of embarkation (Southampton) in the model.

We also identify a gap between the null deviance and residual deviance attributed to each of the statistically significant variables, especially with sex (reducing the residual deviance by 278). The table below lists the variables with the respective reduction of deviance

```
anova(glm_fit,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: survived
##
## Terms added sequentially (first to last)
##
##
##          Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                      1046    1392.63
## pclass    2  101.484      1044    1291.14 < 2.2e-16 ***
## sex       1  278.488      1043    1012.66 < 2.2e-16 ***
## age       1   30.611      1042     982.04 3.153e-08 ***
## sibsp     1   14.112      1041     967.93 0.0001722 ***
## parch     1    0.025      1040     967.91 0.8744118
```

```
## fare      1    1.265      1039     966.64 0.2606981
## embarked  2    3.394      1037     963.25 0.1831882
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now, we are going to evaluate how well the logistic regression model against the test data

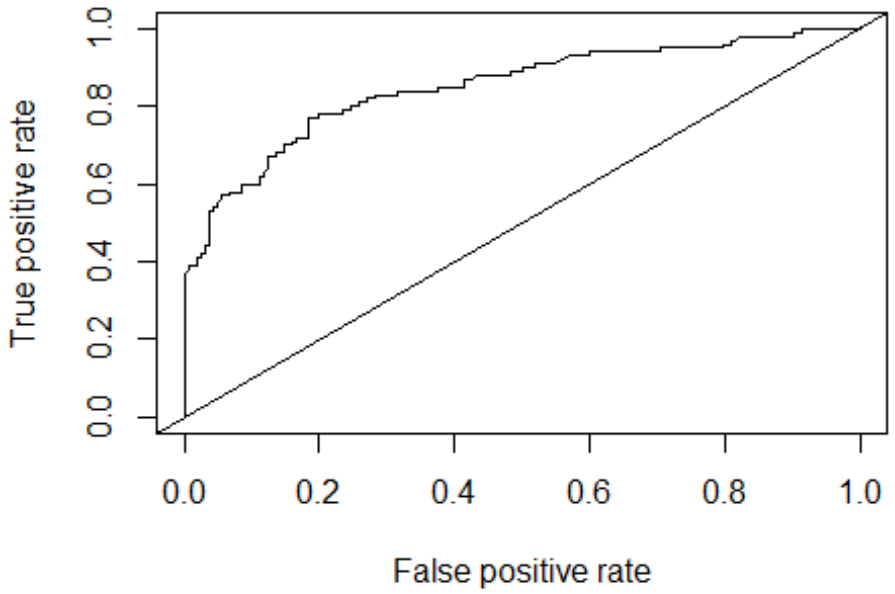
```
#1st Model - Prediction and Accuracy
p_hat <- predict(glm_fit, newdata = titanic_test, type = "response")
y_hat <- ifelse(p_hat > 0.5, 1, 0) %>% factor
cm <- confusionMatrix(y_hat, titanic_test$survived)
cm

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 140  32
##          1  22  68
##
##                Accuracy : 0.7939
##                  95% CI : (0.7398, 0.8412)
##     No Information Rate : 0.6183
##     P-Value [Acc > NIR] : 7.716e-10
##
##                   Kappa : 0.5548
##  Mcnemar's Test P-Value : 0.2207
##
##             Sensitivity : 0.8642
##             Specificity : 0.6800
##          Pos Pred Value : 0.8140
##          Neg Pred Value : 0.7556
##              Prevalence : 0.6183
##          Detection Rate : 0.5344
##    Detection Prevalence : 0.6565
##       Balanced Accuracy : 0.7721
##
##        'Positive' Class : 0
##

accuracy<-cm$overall['Accuracy']
accuracy

##  Accuracy
## 0.7938931
```

The accuracy hits 0.7939, which can be perceived as good. Sensitivity and specificity are relatively high.

To measure the model performance for the binary classification problem, we will analyze the True Positive Rate against the False Positive Rate Receiver Operating Characteristic

Curve (ROC) / Area Under Curve (AUC). The higher AUC (closer to 1), the better the prediction ability of the model

```
#Plot ROC curve
predict_perf <- prediction(p_hat, titanic_test$survived)
metric <- performance(predict_perf, measure = "tpr", x.measure = "fpr")
plot(metric)
abline(a=0, b=1)
```




```
#Calculate AUC area
auc <- performance(predict_perf, measure = "auc")
auc
```

```
## An object of class "performance"
## Slot "x.name":
## [1] "None"
##
## Slot "y.name":
## [1] "Area under the ROC curve"
##
## Slot "alpha.name":
## [1] "none"
##
## Slot "x.values":
## list()
##
## Slot "y.values":
```

```
## [[1]]
## [1] 0.8497222
##
##
## Slot "alpha.values":
## list()
```

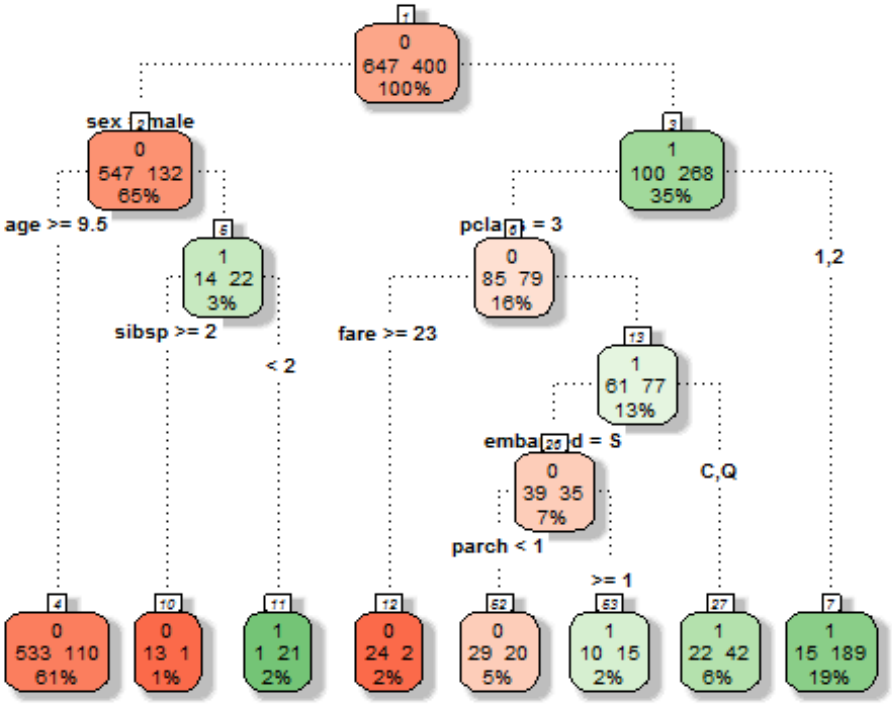
We can be satisfied with the auc 0.85, a good classifier. Let's save the result to data frame and next we'll see if we can get a better result with Decision Tree

```
results<- data_frame(method = "Binomial Logistic Regression", Accuracy =
accuracy, AUC=as.numeric(auc@y.values))
```

2. Decision Tree

Another popular predictive modelling tool for classification is Decision Tree which produces a model that predicts outcome by using any attributes to split data repeatedly until it reaches the purity of the subsets. By plotting the decision tree, we can easily interpret which information gain is larger from the attributes.

```
#2nd Model - Decision Tree
tree <- rpart(survived ~ ., data=titanic_train, method="class")
rpart.plot(tree,type=4, cex=0.6, extra=101, box.palette="RdGn",branch.lty=3,
shadow.col="gray", nn=TRUE)
```



Looking at the above tree plot, we can deduce as follows:

- The gender of the passenger is the most important factor determining the survival, with being male above 9.5 years old has least possibility to survive. For male under 9.5 years old, having less than 2 siblings helps increase the odds to survive.
- On the other leaf node (female), non-3rd class passengers have almost 20% survived chance. This possibly supports the theory that higher class female passengers might have be given the priority to be on the lifeboat. Age does not seem to have contributed in the female survival rate. Attributes like fare, port of embarkation and parents children relationship also play as survival factors for 1st class female passengers.

Now, let's calculate the accuracy of the prediction model against the same test dataset

```
#2nd Model - Prediction and Accuracy
pred <- predict(tree, titanic_test, type="class")
confusionMatrix(pred, titanic_test$survived)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 146  41
##          1  16  59
##
##                Accuracy : 0.7824
##                  95% CI : (0.7275, 0.8309)
##     No Information Rate : 0.6183
##     P-Value [Acc > NIR] : 9.64e-09
##
##                   Kappa : 0.5159
##  Mcnemar's Test P-Value : 0.001478
##
##             Sensitivity : 0.9012
##             Specificity : 0.5900
##          Pos Pred Value : 0.7807
##          Neg Pred Value : 0.7867
##              Prevalence : 0.6183
##          Detection Rate : 0.5573
##    Detection Prevalence : 0.7137
##       Balanced Accuracy : 0.7456
##
##        'Positive' Class : 0
##
```

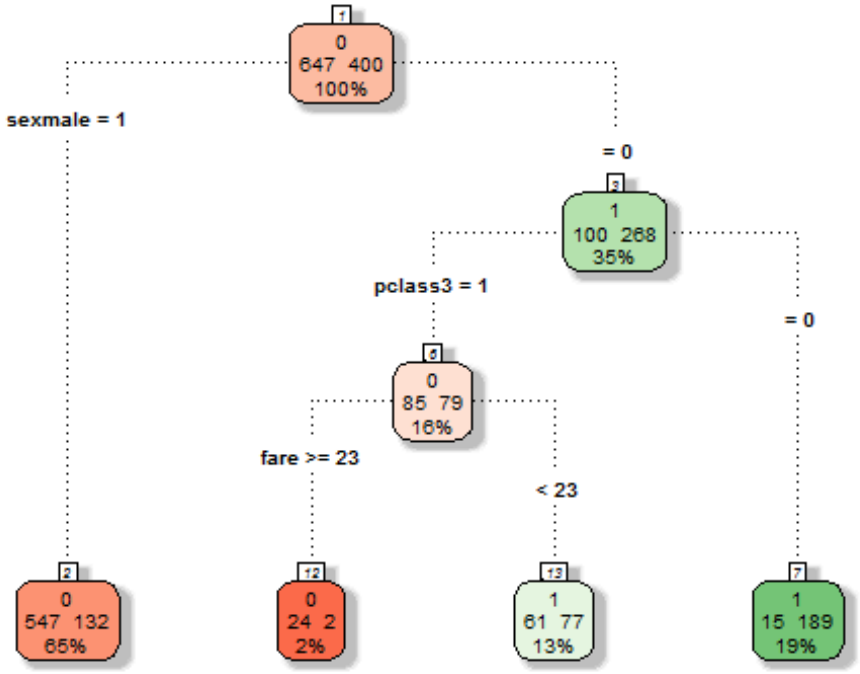
The accuracy is slightly less than the one predicted with the Logistic Regression We do also need to consider that there may be a situation of overfitting in our decision tree. Let's check by running the model against the train data

```
#2nd Model - Check Overfitting
pred_train <- predict(tree, titanic_train, type="class")
confusionMatrix(pred_train, titanic_train$survived)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 599 133
##          1  48 267
##
##                Accuracy : 0.8271
##                  95% CI : (0.8028, 0.8496)
##     No Information Rate : 0.618
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6184
##  Mcnemar's Test P-Value : 4.274e-10
##
##             Sensitivity : 0.9258
##             Specificity : 0.6675
##          Pos Pred Value : 0.8183
##          Neg Pred Value : 0.8476
##              Prevalence : 0.6180
##          Detection Rate : 0.5721
##    Detection Prevalence : 0.6991
##       Balanced Accuracy : 0.7967
##
##        'Positive' Class : 0
##
```

We get higher accuracy for training data compared to test data, which is not ideal. To address the issue, we can further work on the algorithm by repeated cross validation method with trainControl. 3 separate 10-fold validations are used

```
set.seed(1)
folds = createMultiFolds(titanic_train$survived, k = 10, times = 3)
control <- trainControl(method = "repeatedcv", index = folds)
tree_cv <- train(survived ~ ., data = titanic_train, method = "rpart",
trControl = control)
rpart.plot(tree_cv$finalModel,type=4, cex=0.6, extra=101,
box.palette="RdGn",branch.lty=3, shadow.col="gray", nn=TRUE)
```




```
pred_cv <- predict(tree_cv, titanic_test)
cm<-confusionMatrix(pred_cv, titanic_test$survived)
cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 141  32
##          1  21  68
##
##                Accuracy : 0.7977
##                  95% CI : (0.7439, 0.8446)
##     No Information Rate : 0.6183
##     P-Value [Acc > NIR] : 3.179e-10
##
##                   Kappa : 0.5622
##  Mcnemar's Test P-Value : 0.1696
##
##             Sensitivity : 0.8704
##             Specificity : 0.6800
##          Pos Pred Value : 0.8150
##          Neg Pred Value : 0.7640
##              Prevalence : 0.6183
##          Detection Rate : 0.5382
##    Detection Prevalence : 0.6603
##       Balanced Accuracy : 0.7752
```

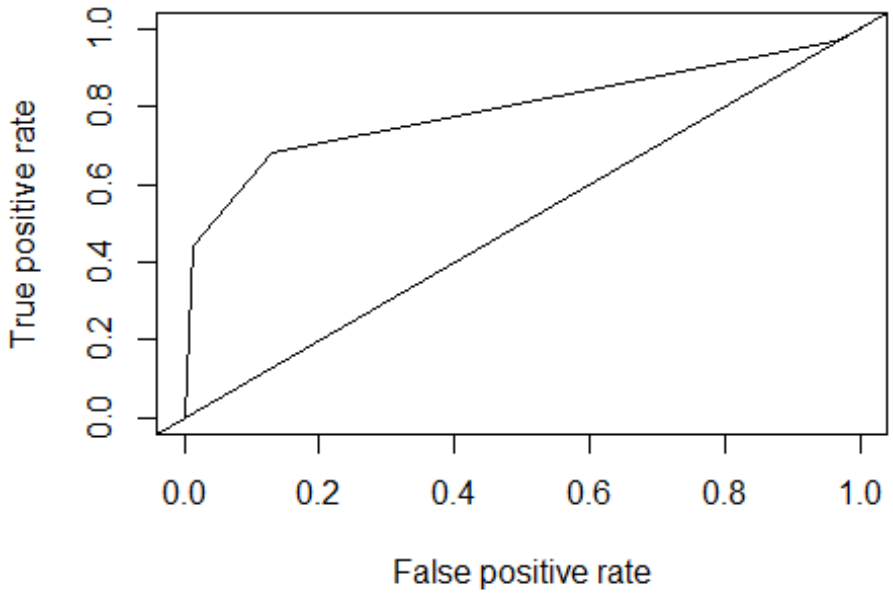
```
##
##        'Positive' Class : 0
##

accuracy<-cm$overall['Accuracy']
accuracy

##  Accuracy
## 0.7977099
```

We have pruned the earlier tree and come up with a better accuracy result after cross-validation. Only sex, pclass and fare are considered in the new model. Now, if we run this model against the same train data, it yield more or less the same accuracy compared with running against test set.

```
pred_cv_train <- predict(tree_cv, titanic_train)
confusionMatrix(pred_cv_train, titanic_train$survived)$overall['Accuracy']

##  Accuracy
## 0.7994269
```

At this stage, we can accept the newer tree model as the better tree. Next, we are assessing it towards the performance metric (ROC curve and AUC)

```
#Plot ROC curve
pred_cv <- predict(tree_cv, titanic_test, type="prob")
predict_perf_cv <- prediction(pred_cv[,2], titanic_test$survived)
metric <- performance(predict_perf_cv, measure = "tpr", x.measure = "fpr")
plot(metric)
abline(a=0, b=1)
```




```
#Calculate AUC area
auc <- performance(predict_perf_cv, measure = "auc")
auc
```

```
## An object of class "performance"
## Slot "x.name":
## [1] "None"
##
## Slot "y.name":
## [1] "Area under the ROC curve"
##
## Slot "alpha.name":
## [1] "none"
##
## Slot "x.values":
## list()
##
## Slot "y.values":
## [[1]]
## [1] 0.7923765
##
##
## Slot "alpha.values":
## list()
```

Performance wise, the decision tree has smaller AUC value compared to the Logistic Regression method

```
results <- bind_rows(results, data_frame(method="Decision Tree", Accuracy =
accuracy, AUC=as.numeric(auc@y.values)))
```

Finally, let's display the comparison result of both algorithms

```
results
```

```
## # A tibble: 2 x 3
##   method                         Accuracy   AUC
##   <chr>                             <dbl> <dbl>
## 1 Binomial Logistic Regression    0.794 0.850
## 2 Decision Tree                   0.798 0.792
```

Conclusion

We began with the exploration on Titanic data set where we looked at the content and structure of the passengers data. We went through the attributes and provide some visualization for better grasp on how each feature related to survival rate. Furthermore, we made necessary type conversions on the variables and decided if the variable ws for keep or to be removed from the data set. Any missing values were identified and computed. Finally, the dataset was split into training and test set.

We were able to produce predictive models for survival outcome using two different supervised machine learning algorithms, namely, binomial logistic regression and decision tree. Both models resulted in on-par accuracy (79%), however, binomial logistic regression performed much better in AUC, with as high as 85%, making it a satisfactory classifier.

Despite this, we have not yet achieved greater accuracy, leaving the room for improvement. For a later stage, we can conduct thorough feature engineering on the data set and/or experiment few other algorithms. Overfitting may also be another area to research further.