Survival prediction for the Titanic data set

Introduction

The Titanc data set was obtained on Kaggle (https://www.kaggle.com/c/titanic). It comes with 2 files (train.csv and test.csv). The objective of this study is to predict the survival/death of the passengers in the test.csv file by using the passenger data in the train.csv file as training set. Logistic regression is used as the modeling approach in this study.

Firstly, the two csv files are loaded in R.

```
train <- read.csv("train.csv")
test <- read.csv("test.csv")
```

The next step involves inspecting the structures of the two data sets.

```
#Inspect the structure of train.csv
str(train)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 1
91 358 277 16 559 520 629 417 581 ...
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2
1 1 ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : Factor w/ 681 levels "110152","110413",..: 524 597 6
70 50 473 276 86 396 345 133 ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : Factor w/ 148 levels "","A10","A14",..: 1 83 1 57 1
1 131 1 1 1 ...
##  $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4
 2 ...
```

In the training data set, there are 891 passengers with 12 descriptors. The types of descriptors can be grouped as below.

1. Integer: PassengerID, Survived (0 for death, 1 for survived), Pclass, SibSp (number of siblings), and Parch (number of parents and children)

2. Factor: Name, Sex, Ticket, Cabin and Embarked

3. Number: Age and Fare

The descriptors, Survived, Pclass, and EmBarked, are essentially categorical variables.

Regarding the structure of the testing data set.

```
#Inspect the structure of test.csv
str(test)
```

```
## 'data.frame':    418 obs. of  11 variables:
##  $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
##  $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
##  $ Name       : Factor w/ 418 levels "Abbott, Master. Eugene Joseph
",..: 210 409 273 414 182 370 85 58 5 104 ...
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2
1 2 ...
##  $ Age        : num  34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp      : int  0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch      : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Ticket     : Factor w/ 363 levels "110469","110489",..: 153 222 7
4 148 139 262 159 85 101 270 ...
##  $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
##  $ Cabin      : Factor w/ 77 levels "","A11","A18",..: 1 1 1 1 1 1 1
 1 1 1 ...
##  $ Embarked   : Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3
 ...
```

The output shows that the structure of the testing data set is essentially the same as that of the training data set with the exception of the missing Survived descriptor. The objective of this study is to predict the surival data of the 418 passengers in the testing data set.

From now on, I will focus on the training data set first in order to do some exploratory data analyses and model building.

Exploratory Data Analysis

The next step is to inspect the first few rows of data in the training data set.

```
head(train)
```

```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
## 4           4        1      1
## 5           5        0      3
## 6           6        0      3
##                                                  Name    Sex Age Sib
Sp
```

```
## 1                              Braund, Mr. Owen Harris   male  22
 1
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38
 1
## 3                               Heikkinen, Miss. Laina female  26
 0
## 4        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35
 1
## 5                            Allen, Mr. William Henry   male  35
 0
## 6                                    Moran, Mr. James   male  NA
 0
##   Parch           Ticket    Fare Cabin Embarked
## 1     0        A/5 21171  7.2500              S
## 2     0         PC 17599 71.2833   C85        C
## 3     0 STON/O2. 3101282  7.9250              S
## 4     0           113803 53.1000  C123        S
## 5     0           373450  8.0500              S
## 6     0           330877  8.4583              Q
```

After having a rough idea of what the training data set looks like, it would be great to get an overall summary of the data.

```
summary(train)

##   PassengerId       Survived          Pclass
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000
##  Median :446.0   Median :0.0000   Median :3.000
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##
##                                     Name         Sex           Age

##  Abbing, Mr. Anthony                   :  1   female:314   Min.   : 0.
42
##  Abbott, Mr. Rossmore Edward           :  1   male  :577   1st Qu.:20.
12
##  Abbott, Mrs. Stanton (Rosa Hunt)      :  1                Median :28.
00
##  Abelson, Mr. Samuel                   :  1                Mean   :29.
70
##  Abelson, Mrs. Samuel (Hannah Wizosky):  1                3rd Qu.:38.
00
##  Adahl, Mr. Mauritz Nils Martin        :  1                Max.   :80.
00
##  (Other)                               :885                NA's   :17
7
##      SibSp           Parch             Ticket         Fare
```

```
##  Min.   :0.000   Min.   :0.0000   1601    :  7   Min.   :  0.00
##  1st Qu.:0.000   1st Qu.:0.0000   347082  :  7   1st Qu.:  7.91
##  Median :0.000   Median :0.0000   CA. 2343:  7   Median : 14.45
##  Mean   :0.523   Mean   :0.3816   3101295 :  6   Mean   : 32.20
##  3rd Qu.:1.000   3rd Qu.:0.0000   347088  :  6   3rd Qu.: 31.00
##  Max.   :8.000   Max.   :6.0000   CA 2144 :  6   Max.   :512.33
##                                   (Other) :852
##          Cabin     Embarked
##             :687    :  2
##  B96 B98    :  4   C:168
##  C23 C25 C27:  4   Q: 77
##  G6         :  4   S:644
##  C22 C26    :  3
##  D          :  3
##  (Other)    :186
```

There are several points to note in the output.

1. The median of Survived is 0 –> 50% of the passengers in the training data set are dead.
2. Most people were in class 3 in Titanic.
3. Most passengers on Titanic were male.
4. Passengers were across different age groups, ranging from babies to old people. Moreover, the age values of 177 people are missing.
5. 50% of the passengers did not have siblings. However, some people had up to 8 siblings on board.
6. Some passengers might have up to 6 children and parents with them. Big families!
7. 687 values in the Ticket variable are missing. (687/891)* 100 % ~ 77%. Thus, basically 77% of the Ticket values are missing.
8. 2 values in the Embarked variable are missing.

Let's visualize some variables next. However, not all variables will be visualized because not all of them are useful for survival prediction.

PassengerId, Name, and Ticket are just nominal variables so they are not related to the survival rate prediction and they are not going to be visualized.

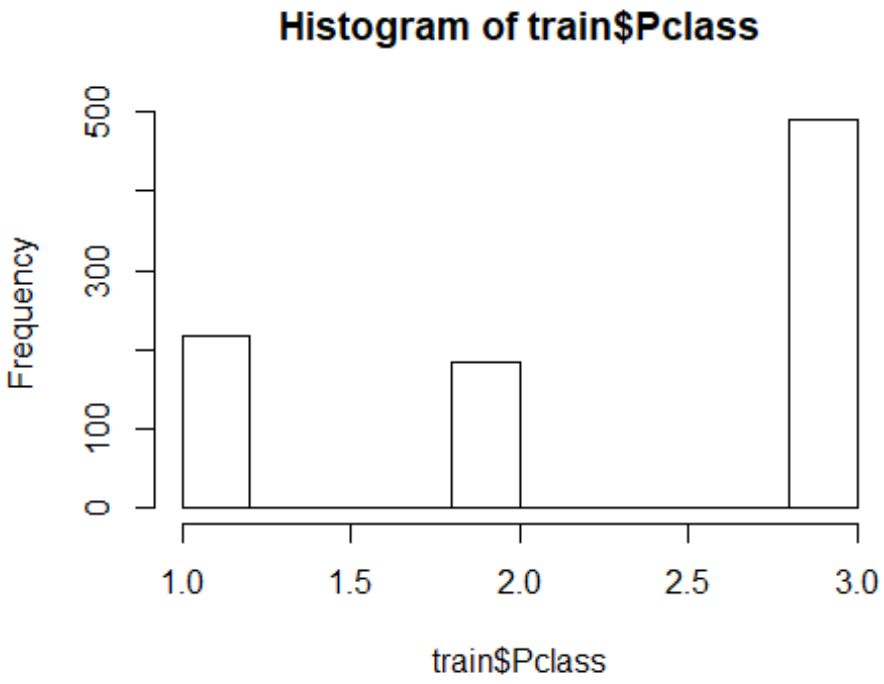
I am going to visualize the Survived variable with a histogram.

```
hist(train$Survived)
```

From the plot, most passengers are dead in the training set.

Let's take a look at the Pclass (ticket class) variable by using a histogram.

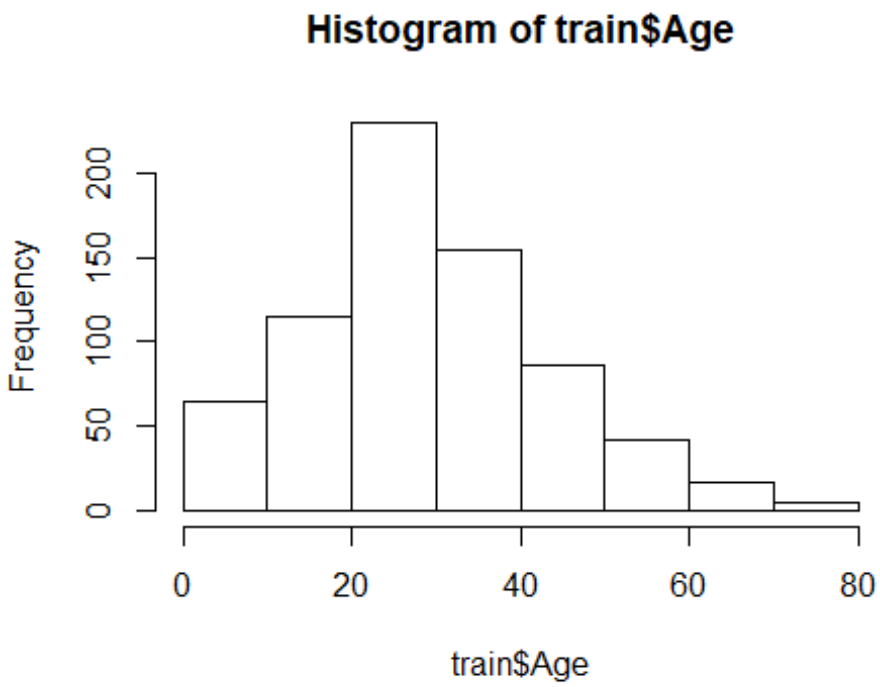
```
hist(train$Pclass)
```



Most passengers picked the third class for their tickets.

The Age variable is then visualized with a density plot in order to show the distribution of ages with the missing values omitted.

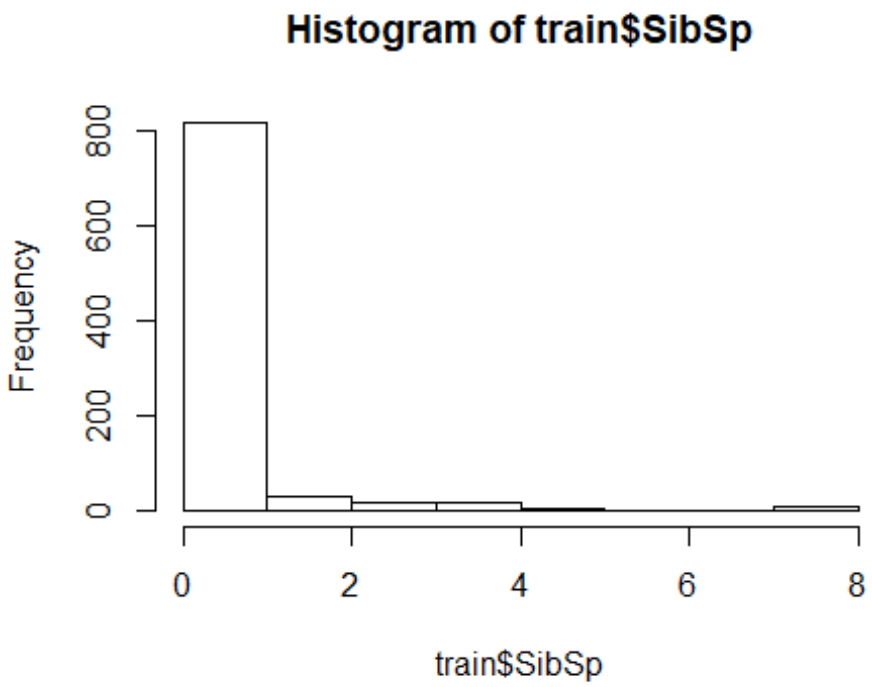
```
hist(train$Age)
```



Most passengers fall in 20 to 40 years old. There are 177 values (~ 20%) missing in this variable and this will need to be dealt with separately.

Let's take a look at the number of siblings.

```
hist(train$SibSp)
```



It turns out that most people did not have any siblings with them. Passengers with more than 4 siblings on board were very rare.

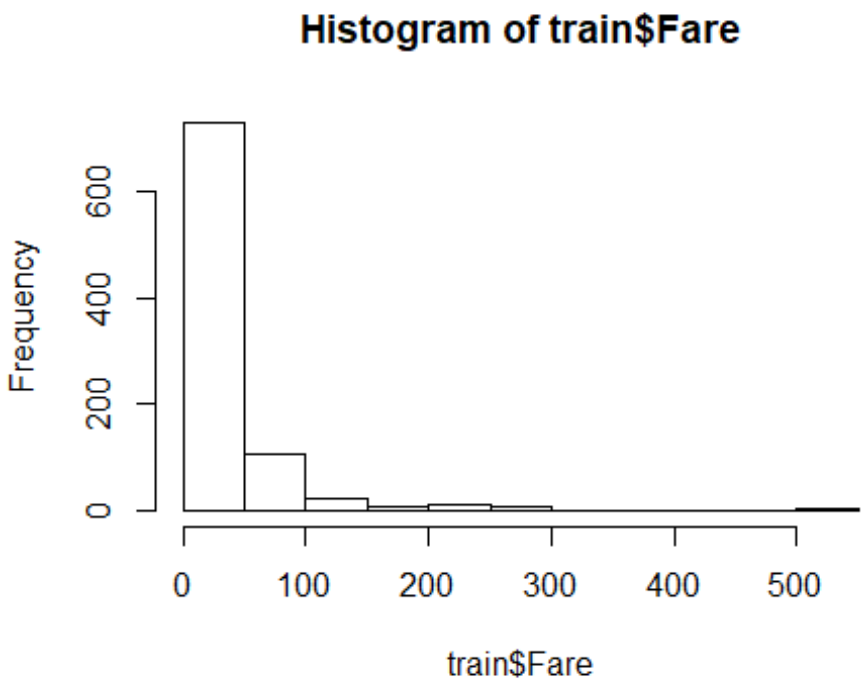
Another variable related to family is the one describing the number of children and parents.

```
hist(train$Parch)
```

Most people did not have any parents nor children with them. Passengers with 2 or more children and parents on board were very rare.

Regarding Fare, let's see.

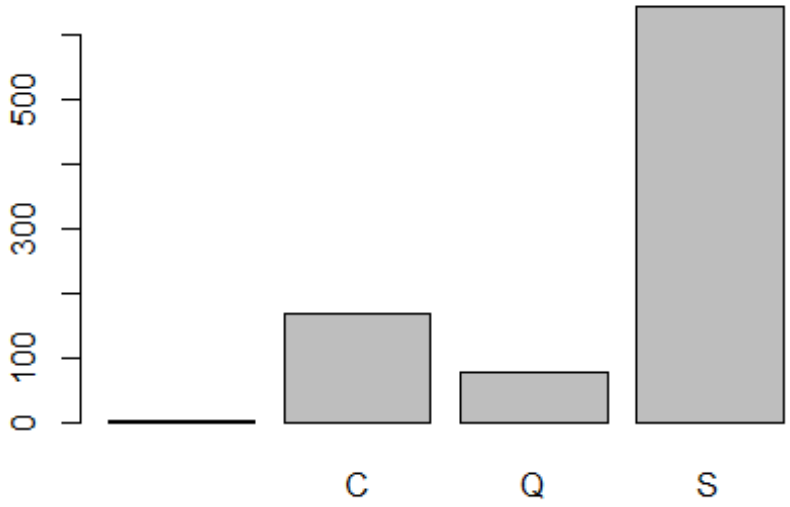
```
hist(train$Fare)
```



Most people picked cheap tickets.

Since most values in Cabin are missing, this variable is not visualized. Regarding Embarked, since it is a qualitative variable, it will be visualized by using a barplot.

```
barplot(table(train$Embarked))
```



'S' (S for Southampton) is the most frequent value in the Embarked variable. Since only 2 values are missing in Embarked, they are going to be neglected from now on.

```
train <- train[!is.na(train$Embarked),]
rownames(train) <- NULL
#Check for missing values in the Embarked variable
table(is.na(train$Embarked))

##
## FALSE
##   891
```

There are no missing values in Embarked now.

Now I need to deal with the missing values in the Age variable. Several points to note:

1. Simple missing value imputation with mean/median/mode is not a good way to solve this problem because this can change the variance of Age. Moreover, this oversimplified approach does not produce any realistic estimates.

2. Linear regression is a more sophisticated approach in order to impute missing values. However, it assumes the values are normally distributed. I am going to perform a statistical test for normality (Shapiro-Wilk Normality Test) for Age.

```
shapiro.test(train$Age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  train$Age
## W = 0.98146, p-value = 7.337e-08
```

For Shapiro-Wilk normality test, H_0: The population is normally distributed. H_a: The population is not normally distributed.

Since the p-value is < 0.05 (at 95% confidence interval), the null hypothesis is rejected. Thus, the variables in Age are not normally distributed.

Thus, imputing the missing values in Age is not a good way out. Imputing the missing values in Age using linear regression may yield unrealistic estimates.

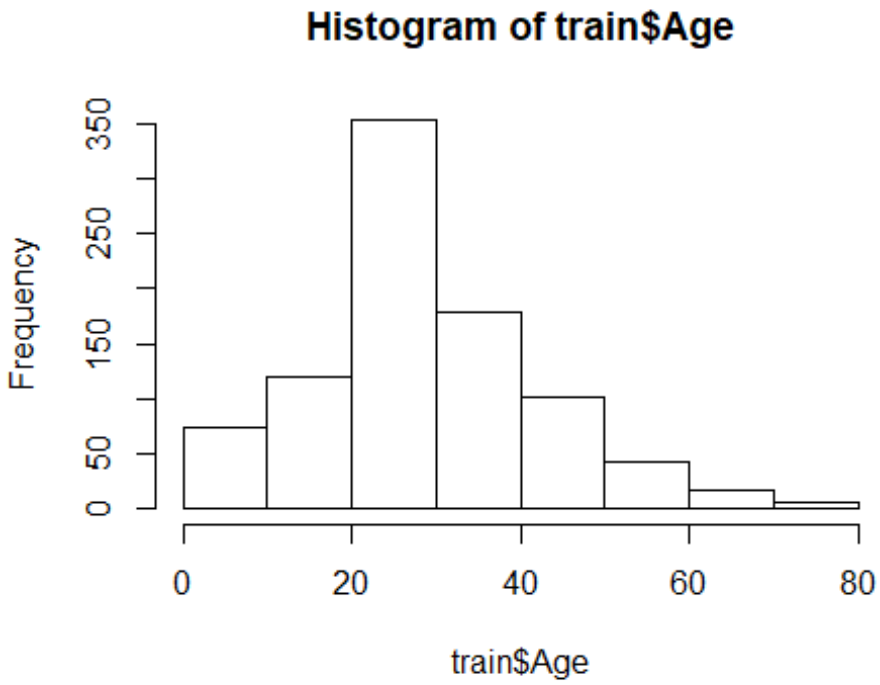
3. kNN (k Nearest Neighbour) is a good way out in this case because it does not assume any distribution of data. It imputes a missing value based on distance between a missing value and its k nearest neighbour. In order to perform kNN imputation, the VIM R library is loaded.

```
#Load the VIM library
library("VIM")

## Warning: package 'VIM' was built under R version 3.4.4

## Loading required package: colorspace

## Loading required package: grid

## Loading required package: data.table

## Warning: package 'data.table' was built under R version 3.4.4

## VIM is ready to use.
##  Since version 4.0.0 the GUI is in its own package VIMGUI.
##
##           Please use the package to use the new (and old) GUI.

## Suggestions and bug-reports can be submitted at: https://github.com/
alexkowa/VIM/issues

##
## Attaching package: 'VIM'

## The following object is masked from 'package:datasets':
##
##     sleep

# An empirical rule to choose the number k is to take the square root o
f the number of #training samples.
# The number of existing data entries in Age = 891 - 177 = 714. Square
root (714) = 26
#
```

```
# kNN imputation
train <- kNN(train, variable ="Age", k = 26)
table(is.na(train$Age))

##
## FALSE
##   891
```

There are no missing values in Age now. A good imputation would make no change to the population distribution. Let's visualize the distribution of Age after missing values imputation.

```
hist(train$Age)
```



The distribution after imputation is essentially the same as that before imputation. I will perform a statistical test to make sure that the population distribution remains as a non-normal distribution.

```
shapiro.test(train$Age)

##
##  Shapiro-Wilk normality test
##
## data:  train$Age
## W = 0.97377, p-value = 1.4e-11
```

The p-value is <0.05, so the null hypothesis is rejected. The Age population is not normally distributed.

Since not all variables are going to be included in model building, I build another data frame to store the required variables for modeling in 'train.red'.

```
train.red <- subset(train, select = c(Survived,Pclass, Sex, Age, SibSp,
 Parch, Fare, Embarked))
colnames(train.red)

## [1] "Survived" "Pclass"   "Sex"      "Age"      "SibSp"    "Parch"

## [7] "Fare"     "Embarked"
```

These are the variables which are required in my model.

Model Building

Logistic Regression

I will perform the survival prediction using logistic regression because it is very good at performing binary classification. In order to do so, I will need to train my model. Thus, I will split the data in train.csv into training set (80%) and testing set (20%) in order to validate the model.

```
train.model <- train.red[1:712,]
test.model <- train.red[713:891,]
```

Applying the logistic regression model to the train.model

```
model <- glm(Survived ~ ., family=binomial(link='logit'), data=train.mo
del)
summary(model)

##
## Call:
## glm(formula = Survived ~ ., family = binomial(link = "logit"),
##     data = train.model)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6484  -0.6110  -0.4122   0.6453   2.4668
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.652e+01  5.354e+02   0.031  0.97539
## Pclass      -1.244e+00  1.716e-01  -7.251 4.12e-13 ***
## Sexmale     -2.686e+00  2.238e-01 -12.001  < 2e-16 ***
## Age         -4.513e-02  9.209e-03  -4.901 9.56e-07 ***
## SibSp       -3.324e-01  1.234e-01  -2.693  0.00708 **
## Parch       -1.647e-01  1.407e-01  -1.170  0.24196
## Fare         1.155e-04  2.680e-03   0.043  0.96563
## EmbarkedC   -1.075e+01  5.354e+02  -0.020  0.98398
## EmbarkedQ   -1.071e+01  5.354e+02  -0.020  0.98404
```

```
## EmbarkedS   -1.106e+01  5.354e+02  -0.021  0.98352
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 952.58  on 711  degrees of freedom
## Residual deviance: 634.80  on 702  degrees of freedom
## AIC: 654.8
##
## Number of Fisher Scoring iterations: 12
```

At 95% confidence interval, where alpha = 0.05, only 4 variables are statistically significant, namely, Pclass, sexmale, Age, and SibSp in which Sexmale is the most important because it has the smallest p-value. Thus, Sexmale is highly associated with Survived. The interpretation of the 4 statistically significant important variables are as follows. A unit increase in age decreases the log odds of suvival by 0.045. Being a male decreases the log odds of survival by 2.68. A unit increase in the ticket class decreases the log odds of survival by 1.2. An increase in the number of siblings on board decreases the log odds of survival by 3.3.

Test for significance of the overall regression In this section, the significance of the overall regression was tested by using the difference between null deviance and residual deviance to obtain the p-value.

```
1-pchisq(317.78,9)
```

```
## [1] 0
```

Since the p-value is 0, the overall regression is significance.

Model Fit Assessment

Goodness of Fit Hypothesis Testing Using deviance residual

```
c(deviance(model), 1-pchisq(deviance(model),702))
```

```
## [1] 634.7977051   0.9668369
```

Since the p-value > 0.5, the null hypothesis must be accepted. This indicates that the fitting is a good fit.

Assessing the goodness of fit using Pearson residuals

```
## Using Pearson residuals
pearson_residuals <- residuals(model, type="pearson")
pearson_residuals.tvalue <- sum(pearson_residuals^2)
c(pearson_residuals.tvalue, 1-pchisq(pearson_residuals.tvalue,702))
```

```
## [1] 737.5550051   0.1707896
```

Since the p-value > 0.5, the null hypothesis must be accepted. This indicates that the fitting is a good fit.

Cross Validation Using Data in test.model

```
prediction <- predict(model, newdata = test.model, type='response')
prediction <- ifelse(prediction > 0.5, 1,0) #0.5 as the threshold value
```

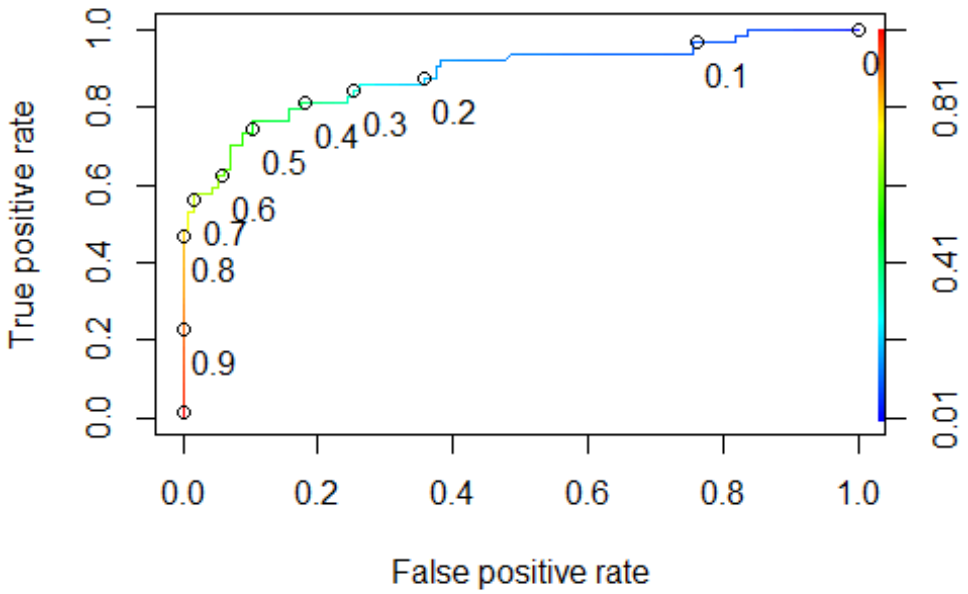
Performance assessment using a confusion matrix as provided in the caret R package

```
library("caret")

## Loading required package: lattice

## Loading required package: ggplot2

library("e1071")
confusionMatrix(data=prediction, reference=test.model$Survived)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 103  17
##          1  12  47
##
##                Accuracy : 0.838
##                  95% CI : (0.7757, 0.8887)
##     No Information Rate : 0.6425
##     P-Value [Acc > NIR] : 5.581e-09
##
##                   Kappa : 0.6411
##  Mcnemar's Test P-Value : 0.4576
##
##             Sensitivity : 0.8957
##             Specificity : 0.7344
##          Pos Pred Value : 0.8583
##          Neg Pred Value : 0.7966
##              Prevalence : 0.6425
##          Detection Rate : 0.5754
##    Detection Prevalence : 0.6704
##       Balanced Accuracy : 0.8150
##
##        'Positive' Class : 0
##
```

The accuracy of the logistic regression model is about 84%, which is quite good.

Another way to assess the prediction performance is to employ the ROC curve using the ROCR library.

```
library(ROCR)

## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess

predictions <- predict(model, newdata=test.model, type="response")
pred <- prediction(predictions, test.model$Survived)
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
plot(perf, colorize = TRUE, text.adj = c(-0.2,1.7), print.cutoffs.at =
seq(0,1,0.1))
```



A ROC plot is obtained by plotting the true positive rate and the false positive rate. By computing the area under the ROC curve (AUC), an indicator of the reliability of the prediction can be obtained.

```
auc <- performance(pred, measure = "auc")
auc <- auc@y.values[[1]]
auc

## [1] 0.8855299
```

An AUC value of 1 is ideal. In this case, the AUC value is 0.89, which is very close to 1. Thus, this model comes with good predictability.

Prediction with the new set data

Selecting the required variables for survival prediction

```
test.data <- subset(test, select = c(2,4,5,6,7,9,11))
colnames(test.data)
```

```
## [1] "Pclass"   "Sex"      "Age"      "SibSp"    "Parch"    "Fare"

## [7] "Embarked"
```

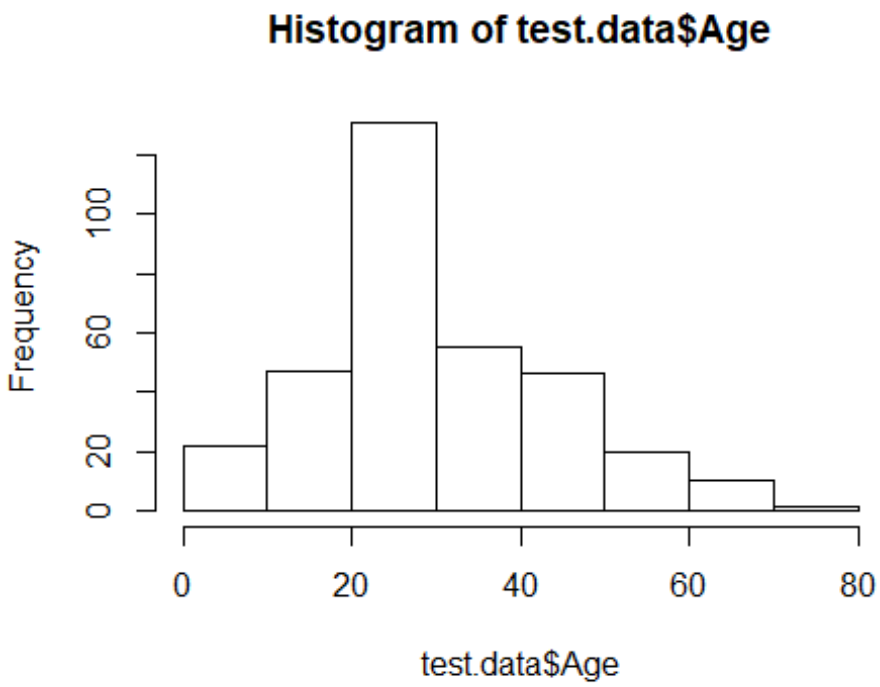
```
summary(test.data)
```

```
##      Pclass          Sex           Age            SibSp
##  Min.   :1.000   female:152   Min.   : 0.17   Min.   :0.0000
##  1st Qu.:1.000   male  :266   1st Qu.:21.00   1st Qu.:0.0000
##  Median :3.000                Median :27.00   Median :0.0000
##  Mean   :2.266                Mean   :30.27   Mean   :0.4474
##  3rd Qu.:3.000                3rd Qu.:39.00   3rd Qu.:1.0000
##  Max.   :3.000                Max.   :76.00   Max.   :8.0000
##                               NA's   :86
##      Parch             Fare         Embarked
##  Min.   :0.0000   Min.   :  0.000   C:102
##  1st Qu.:0.0000   1st Qu.:  7.896   Q: 46
##  Median :0.0000   Median : 14.454   S:270
##  Mean   :0.3923   Mean   : 35.627
##  3rd Qu.:0.0000   3rd Qu.: 31.500
##  Max.   :9.0000   Max.   :512.329
##                   NA's   :1
```

There are 86 missing values in Age and 1 missing value in Fare. Again, kNN imputation will be used for imputing missing values in Age.

Visualizing the distribution of Age in test.data before missing values imputation

```
hist(test.data$Age)
```




```
test.data <- kNN(test.data, variable ="Age", k = 20)
table(is.na(test.data$Age))

##
## FALSE
##   418
```

There are no more missing values in Age. Let's visualize the population distribution of Age after missing values imputation.

```
hist(test.data$Age)
```

The distribution of Age is roughly the same before and after missing values imputation.

Predicting survival data for passengers in test.data

```
#Ignore the 1 missing variable in Fare
test.data <- test.data[!is.na(test.data$Fare),]
rownames(test.data) <- NULL

#Predict
testdatapre <-predict(model, newdata=test.data, type="response")
testprefinal <- ifelse(testdatapre > 0.5, 1,0)
testprefinal
```

```
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
 18
##   0   0   0   0   1   0   1   0   1   0   0   0   1   0   1   1   0
  0
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35
 36
##   1   0   0   0   1   1   1   0   1   0   0   0   0   0   0   1   1
  0
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53
 54
##   1   1   0   0   0   0   0   1   1   0   0   0   1   0   1   0   1
  1
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71
 72
##   0   0   0   0   0   1   0   0   0   1   1   1   1   0   1   1   1
```

```
  0
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89
 90
##   1   1   1   1   0   1   0   1   0   0   0   0   0   0   1   1   1
  0
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107
108
##   1   0   1   0   1   0   1   0   1   0   1   0   0   0   1   0   0
  0
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
126
##   0   0   0   1   1   1   1   0   0   1   1   1   1   0   1   0   0
  1
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
144
##   0   1   0   0   0   0   0   0   0   0   0   0   1   0   0   1   0
  0
## 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161
162
##   0   0   0   0   0   0   1   0   0   0   0   1   1   0   1   1   0
  1
## 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179
180
##   0   0   1   0   0   1   1   0   0   0   0   0   1   1   0   1   1
  0
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197
198
##   0   1   0   1   0   1   0   0   0   0   0   0   0   0   0   1   1
  0
## 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215
216
##   1   1   0   0   1   0   0   1   0   1   0   0   0   0   0   0   0
  1
## 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233
234
##   0   1   0   1   0   1   0   1   1   0   1   0   0   0   1   0   0
  0
## 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251
252
##   0   0   0   1   1   1   1   0   0   0   0   1   0   1   1   1   0
  1
## 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269
270
##   0   0   0   0   0   1   0   0   0   1   1   0   0   0   0   1   0
  0
## 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287
288
##   0   1   1   0   1   0   0   0   0   1   0   1   1   1   0   0   1
  0
## 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305
```

```
306
##   0   0   1   0   0   0   0   1   0   1   0   0   0   0   0   1   1
  0
## 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323
324
##   0   0   0   0   0   0   1   1   1   0   0   0   0   0   0   0   1
  1
## 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341
342
##   0   1   0   0   0   1   1   0   1   0   1   0   0   0   0   0   0
  0
## 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359
360
##   1   0   1   0   1   0   1   1   0   0   0   1   0   1   0   0   1
  0
## 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377
378
##   1   1   0   1   0   0   1   1   0   0   1   0   0   1   1   1   0
  0
## 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395
396
##   0   0   0   1   1   0   1   0   0   0   0   1   1   0   0   0   1
  0
## 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413
414
##   1   0   0   1   0   1   1   0   0   0   0   1   1   1   1   1   0
  1
## 415 416 417
##   0   0   0
```

```
table(testprefinal)
```

```
## testprefinal
##   0   1
## 259 158
```

Using the logistic regression, 255 passengers are predicted to be dead while 162 passengers are predicted to be alive.

Finally, I would like to output the prediction results to titanic_prediction_results.csv file.

```
#Remove the row where Fare has a missing value
test <- test[!is.na(test$Fare),]
rownames(test) <- NULL

export <- data.frame(PassengerID = test$PassengerId, Survived = testpre
final)
write.csv(export, file = 'titanic_prediction_results.csv', row.names =
F)
```