Survival Prediction , Titanic

Aditya Kumar Singh

July 29, 2016

Welcome to my script The sinking of the RMS Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. This sensational tragedy shocked the international community and led to better safety regulations for ships.

One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew. Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper-class.

In this script, I'm trying to analyze what sorts of people were likely to survive. I'm using logistic regression technique for modelling.

## Libraries used

library(Amelia)

## Warning: package 'Amelia' was built under R version 3.3.1

## Loading required package: Rcpp

## ##   
## ## Amelia II: Multiple Imputation  
## ## (Version 1.7.4, built: 2015-12-05)  
## ## Copyright (C) 2005-2016 James Honaker, Gary King and Matthew Blackwell  
## ## Refer to http://gking.harvard.edu/amelia/ for more information  
## ##

library(ROCR)

## Warning: package 'ROCR' was built under R version 3.3.1

## Loading required package: gplots

## Warning: package 'gplots' was built under R version 3.3.1

##   
## Attaching package: 'gplots'

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

## Importing dataset

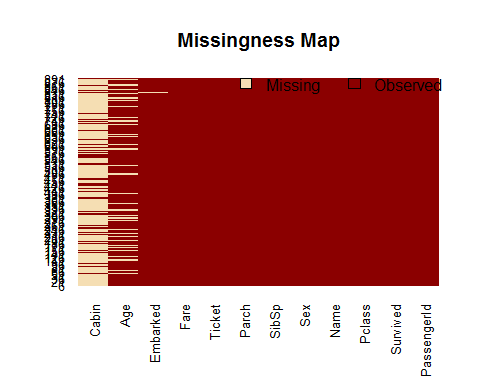
train <- read.csv("train.csv" , na.strings = "")  
test <- read.csv("test.csv" , na.strings = "")  
op <- read.csv("genderclassmodel.csv")

## Cleaning Dataset

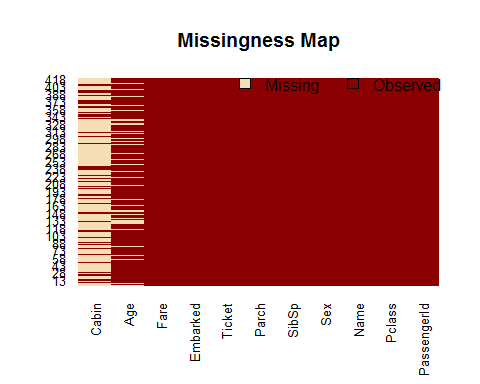
# Checking for missing values

**Training set**

missmap(train)

 **Test set** \*

missmap(test)

 We can see the variable cabin has too many missing values, we will not use it. We will also drop PassengerId since it is only an index and Ticket.

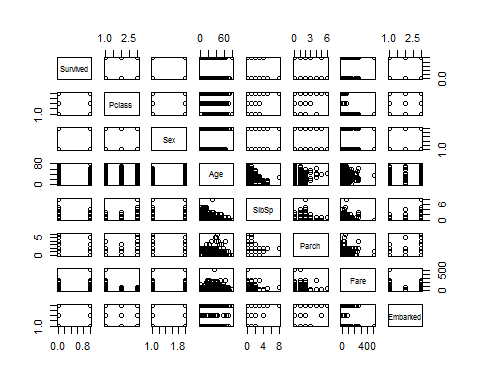
train <- train[,-c(1,4,9,11)]  
test<- test[,-c(1,3,8,10)]  
op <- read.csv("genderclassmodel.csv")  
test <- cbind(op$Survived , test)  
colnames(test)[[1]] <- "Survived"

In age column we will try to fill the empty columns using averaging technique.

train$Age[is.na(train$Age)] <- mean(train$Age , na.rm = T)  
test$Age[is.na(test$Age)] <- mean(test$Age , na.rm = T)  
train <- na.omit(train)

We can see the pattern of distribution between all variables and output

plot(train)



## Creating Model

We will be building few models and will select the best model on the basis of AIC score which will help us in getting an idea that how our model will perfrom on a value outside of training set

model1 <- glm(Survived~. , data = train , family = binomial )  
summary(model1)

##   
## Call:  
## glm(formula = Survived ~ ., family = binomial, data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.6446 -0.5907 -0.4230 0.6220 2.4431   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 5.285188 0.564778 9.358 < 2e-16 \*\*\*  
## Pclass -1.100058 0.143529 -7.664 1.80e-14 \*\*\*  
## Sexmale -2.718695 0.200783 -13.540 < 2e-16 \*\*\*  
## Age -0.039901 0.007854 -5.080 3.77e-07 \*\*\*  
## SibSp -0.325777 0.109384 -2.978 0.0029 \*\*   
## Parch -0.092602 0.118708 -0.780 0.4353   
## Fare 0.001918 0.002376 0.807 0.4194   
## EmbarkedQ -0.034076 0.381936 -0.089 0.9289   
## EmbarkedS -0.418817 0.236794 -1.769 0.0769 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1182.82 on 888 degrees of freedom  
## Residual deviance: 784.19 on 880 degrees of freedom  
## AIC: 802.19  
##   
## Number of Fisher Scoring iterations: 5

model2 <- glm(Survived~Pclass+Sex+Age+SibSp , data = train , family = binomial )  
summary(model2)

##   
## Call:  
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp, family = binomial,   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.6857 -0.6055 -0.4215 0.6125 2.4528   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 5.185020 0.478347 10.839 < 2e-16 \*\*\*  
## Pclass -1.169400 0.119747 -9.766 < 2e-16 \*\*\*  
## Sexmale -2.732586 0.194302 -14.064 < 2e-16 \*\*\*  
## Age -0.040013 0.007772 -5.148 2.63e-07 \*\*\*  
## SibSp -0.356735 0.103918 -3.433 0.000597 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1182.8 on 888 degrees of freedom  
## Residual deviance: 790.3 on 884 degrees of freedom  
## AIC: 800.3  
##   
## Number of Fisher Scoring iterations: 5

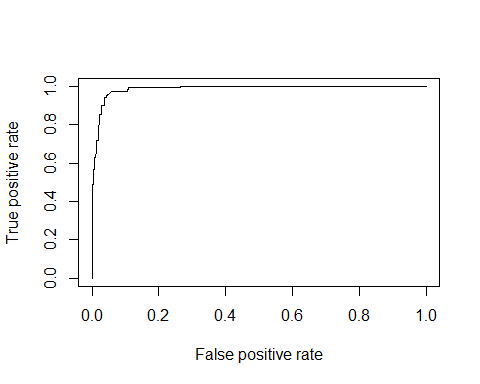
model3 <- glm(Survived~Pclass+Sex+Age+SibSp+Age:Sex , data = train , family = binomial )  
summary(model3)

##   
## Call:  
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Age:Sex,   
## family = binomial, data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.4294 -0.5948 -0.4158 0.5847 2.5393   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 4.45276 0.51787 8.598 < 2e-16 \*\*\*  
## Pclass -1.19396 0.12213 -9.776 < 2e-16 \*\*\*  
## Sexmale -1.38707 0.44832 -3.094 0.001975 \*\*   
## Age -0.01166 0.01177 -0.991 0.321617   
## SibSp -0.33826 0.09999 -3.383 0.000717 \*\*\*  
## Sexmale:Age -0.04761 0.01483 -3.211 0.001325 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1182.8 on 888 degrees of freedom  
## Residual deviance: 779.6 on 883 degrees of freedom  
## AIC: 791.6  
##   
## Number of Fisher Scoring iterations: 5

We can see model3 has lowest AIC so we will go forward with it

## Building ROC curve to access the model perfromance

p <- predict(model3 , newdata = test , type = "response" )  
pr <- prediction(p, test$Survived)  
prf <- performance(pr, measure = "tpr", x.measure = "fpr")  
plot(prf)



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

## [1] 0.986955

## Prediction and checking model accuracy

We will be making prediction using our model and will be trying out various thresholds to improve the model accuracy. We will make use of F score(recall,Precision) to compute the threshold

p <- predict(model3 , newdata = test , type = "response" )  
pred <- ifelse(p>= 0.5, 1 ,0)  
result <- cbind(pred , test$Survived)  
a1<- table(test$Survived, pred)  
a1

## pred  
## 0 1  
## 0 248 29  
## 1 3 138

p <- predict(model3 , newdata = test , type = "response" )  
pred <- ifelse(p>= 0.55, 1 ,0)  
result <- cbind(pred , test$Survived)  
a2<- table(test$Survived, pred)  
a2

## pred  
## 0 1  
## 0 264 13  
## 1 7 134

p <- predict(model3 , newdata = test , type = "response" )  
pred <- ifelse(p>= 0.6, 1 ,0)  
result <- cbind(pred , test$Survived)  
a3<- table(test$Survived, pred)  
a3

## pred  
## 0 1  
## 0 268 9  
## 1 14 127

#Computing Fscores for each  
  
Fscore\_a1 <- 0.895  
Fscore\_a2 <- 0.933  
Fscore\_a1 <- 0.916

Threshold= 0.55 has the highest F score so we will select 0.55 as the threshold value for our model. Lets see the accuracy at this threshold #Accuracy

#accuracy = (TP + TN)/(Total no.of observations)  
accuracy <- 0.9521