Titanic Dataset Survival Prediction

Rohit Amalnerkar

4/23/2020

Dataset Information:

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 tragedy shocked the international community and lead to better safety regulations for ships. One of the reasons that the shipwreck lead 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.

Objective:

The main objective of the dataset is to Predict who survived the sinking ship by applying various Machine Learning Algorithms. o for dead 1 for survived.

Variable Description:

VARIABLE DESCRIPTIONS:

survival - Survival (0 = No; 1 = Yes)

pclass - Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd)

name - Name

sex - Sex

age - Age

sibsp - Number of Siblings/Spouses Aboard

parch - Number of Parents/Children Aboard

ticket - Ticket Number

fare - Passenger Fare

cabin - Cabin

embarked - Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)

Loading the data:

```
library(readxl)
titanic<-read_excel("C:/Users/Rohit/Downloads/Titanic.xls")
head(titanic)
```

```
## # A tibble: 6 x 12
##   PassengerId Survived Pclass Name  Sex     Age SibSp Parch Ticket  Fare
##         <dbl>    <dbl>  <dbl> <chr> <chr> <dbl> <dbl> <dbl> <chr>  <dbl>
## 1           1        0      3 Brau~ male     22     1     0 A/5 2~  7.25
## 2           2        1      1 Cumi~ fema~    38     1     0 PC 17~ 71.3
## 3           3        1      3 Heik~ fema~    26     0     0 STON/~  7.92
## 4           4        1      1 Futr~ fema~    35     1     0 113803 53.1
## 5           5        0      3 Alle~ male     35     0     0 373450  8.05
## 6           6        0      3 Mora~ male     NA     0     0 330877  8.46
## # ... with 2 more variables: Cabin <chr>, Embarked <chr>
```

Checking structture and dimensions:

```
str(titanic)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: num  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : num  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : num  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" "Heikkinen, Miss. La
ina" "Futrelle, Mrs. Jacques Heath (Lily May Peel)" ...
##  $ Sex        : chr  "male" "female" "female" "female" ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : num  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : num  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  NA "C85" NA "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

```
dim(titanic)
```

```
## [1] 891  12
```

Removing unwanted variables which don't add any value to the data:

We find that the following variables hold no value and hence we remove them

```
titanic$PassengerId<-NULL
titanic$Name<-NULL
titanic$Cabin<-NULL
titanic$Ticket<-NULL
```

Converting numeric columns to factor:

```
names<-c("Sex","Survived","Pclass","Embarked")
titanic[,names]<-lapply(titanic[,names],as.factor)
str(titanic)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    891 obs. of  8 variables:
##  $ Survived: Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass  : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ 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   : num  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch   : num  0 0 0 0 0 0 0 1 2 0 ...
##  $ Fare    : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked: Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
```

```
head(titanic)
```

```
## # A tibble: 6 x 8
##   Survived Pclass Sex      Age SibSp Parch  Fare Embarked
##   <fct>    <fct>  <fct>  <dbl> <dbl> <dbl> <dbl> <fct>
## 1 0        3      male      22     1     0  7.25 S
## 2 1        1      female    38     1     0 71.3  C
## 3 1        3      female    26     0     0  7.92 S
## 4 1        1      female    35     1     0 53.1  S
## 5 0        3      male      35     0     0  8.05 S
## 6 0        3      male      NA     0     0  8.46 Q
```

Checking for NAs and treating them:

```
colSums(is.na(titanic))
```

```
## Survived   Pclass      Sex      Age    SibSp    Parch     Fare Embarked
##        0        0        0      177        0        0        0        2
```

```
# for Age
median(titanic$Age,na.rm = T)
```

```
## [1] 28
```

```
titanic$Age[is.na(titanic$Age)]<-28 # replace NAs with the median value of Age
```

```
# for Embarked
summary(titanic$Embarked)
```

```
##    C    Q    S NA's
##  168   77  644    2
```

```
titanic$Embarked[is.na(titanic$Embarked)]<-"S"
# we assign "S" to the missing values as Class "S" has maximum votes in the data
```

```
# rechecking the NAs
colSums(is.na(titanic))
```

```
## Survived   Pclass      Sex      Age    SibSp    Parch     Fare Embarked
##        0        0        0        0        0        0        0        0
```

Dividing the Age columns into categories:

```
titanic$Age<-cut(titanic$Age,breaks = c(0,20,28,40,Inf),labels = c("c1","c2","c3","c4"))
str(titanic)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    891 obs. of  8 variables:
##  $ Survived: Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass  : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age     : Factor w/ 4 levels "c1","c2","c3",..: 2 3 2 3 3 2 4 1 2 1 ...
##  $ SibSp   : num  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch   : num  0 0 0 0 0 0 0 1 2 0 ...
##  $ Fare    : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked: Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
```

Scaling the Numeric columns:

```
names1<-c("Parch", "SibSp", "Fare")
titanic[,names1]<-lapply(titanic[,names1], scale)

summary(titanic)
```

```
##  Survived Pclass      Sex      Age           SibSp.V1
##  0:549    1:216   female:314   c1:179   Min.   :-0.474279
##  1:342    2:184   male  :577   c2:360   1st Qu.:-0.474279
##           3:491                c3:202   Median :-0.474279
##                                c4:150   Mean   : 0.000000
##                                         3rd Qu.: 0.432550
##                                         Max.   : 6.780355
##       Parch.V1             Fare.V1       Embarked
##  Min.   :-0.473408   Min.   :-0.648058   C:168
##  1st Qu.:-0.473408   1st Qu.:-0.488874   Q: 77
##  Median :-0.473408   Median :-0.357190   S:646
##  Mean   : 0.000000   Mean   : 0.000000
##  3rd Qu.:-0.473408   3rd Qu.:-0.024233
##  Max.   : 6.970233   Max.   : 9.661740
```

Splitting the data into training and testing:

```
set.seed(100)
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
index<-createDataPartition(titanic$Survived,p=0.70,list = F)
training_titanic<-titanic[index,]
testing_titanic<-titanic[-index,]

dim(training_titanic)
```

```
## [1] 625   8
```

```
dim(testing_titanic)
```

```
## [1] 266   8
```

Applying Logistic Regression:

```
titanic_model<-glm(Survived~.,data = training_titanic,family = "binomial")
summary(titanic_model)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = "binomial", data = training_titanic)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1027  -0.6974  -0.4105   0.6058   2.4715
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.27716    0.46282   7.081 1.43e-12 ***
## Pclass2     -0.82501    0.35749  -2.308  0.02101 *
## Pclass3     -2.03914    0.36186  -5.635 1.75e-08 ***
## Sexmale     -2.57536    0.23293 -11.057  < 2e-16 ***
## Agec2       -0.99272    0.30350  -3.271  0.00107 **
## Agec3       -0.71172    0.33401  -2.131  0.03310 *
## Agec4       -1.56704    0.37676  -4.159 3.19e-05 ***
## SibSp       -0.36164    0.13925  -2.597  0.00940 **
## Parch       -0.06057    0.12712  -0.476  0.63374
## Fare         0.13488    0.16149   0.835  0.40360
## EmbarkedQ    0.46524    0.44861   1.037  0.29970
## EmbarkedS   -0.23584    0.28594  -0.825  0.40949
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 832.49  on 624  degrees of freedom
## Residual deviance: 569.16  on 613  degrees of freedom
## AIC: 593.16
##
## Number of Fisher Scoring iterations: 5
```

In the summary we find that columns Parch and Fare have no significance hence we remove them.

```
training_titanic$Parch<-NULL
training_titanic$Fare<-NULL
testing_titanic$Parch<-NULL
testing_titanic$Fare<-NULL
```

Running the model again:

```
titanic_model<-glm(Survived~.,data = training_titanic,family = "binomial")
summary(titanic_model)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = "binomial", data = training_titanic)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0683  -0.7102  -0.4074   0.6652   2.4943
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.4057     0.4304   7.914 2.50e-15 ***
## Pclass2      -0.9715     0.3136  -3.098  0.00195 **
## Pclass3      -2.2169     0.2971  -7.462 8.53e-14 ***
## Sexmale      -2.5675     0.2285 -11.237  < 2e-16 ***
## Agec2        -0.9744     0.3002  -3.246  0.00117 **
## Agec3        -0.6977     0.3333  -2.093  0.03632 *
## Agec4        -1.5921     0.3757  -4.238 2.25e-05 ***
## SibSp        -0.3566     0.1285  -2.776  0.00550 **
## EmbarkedQ     0.4650     0.4452   1.045  0.29620
## EmbarkedS    -0.2634     0.2832  -0.930  0.35240
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 832.49  on 624  degrees of freedom
## Residual deviance: 569.99  on 615  degrees of freedom
## AIC: 589.99
##
## Number of Fisher Scoring iterations: 5
```

Calculating predicted pribabilties for training set of Survived being equal to 1

```
training_titanic$predicted_prob<-fitted(titanic_model)
head(training_titanic)
```

```
## # A tibble: 6 x 7
##   Survived Pclass Sex    Age   SibSp[,1] Embarked predicted_prob
##   <fct>    <fct>  <fct>  <fct>     <dbl> <fct>             <dbl>
## 1 0        3      male   c2        0.433 S                0.0589
## 2 1        3      female c2       -0.474 S                0.530
## 3 1        1      female c3        0.433 S                0.908
## 4 0        3      male   c3       -0.474 S                0.102
## 5 0        3      male   c2       -0.474 Q                0.152
## 6 0        1      male   c4       -0.474 S                0.300
```

Checking the ROC curve for cut-off:

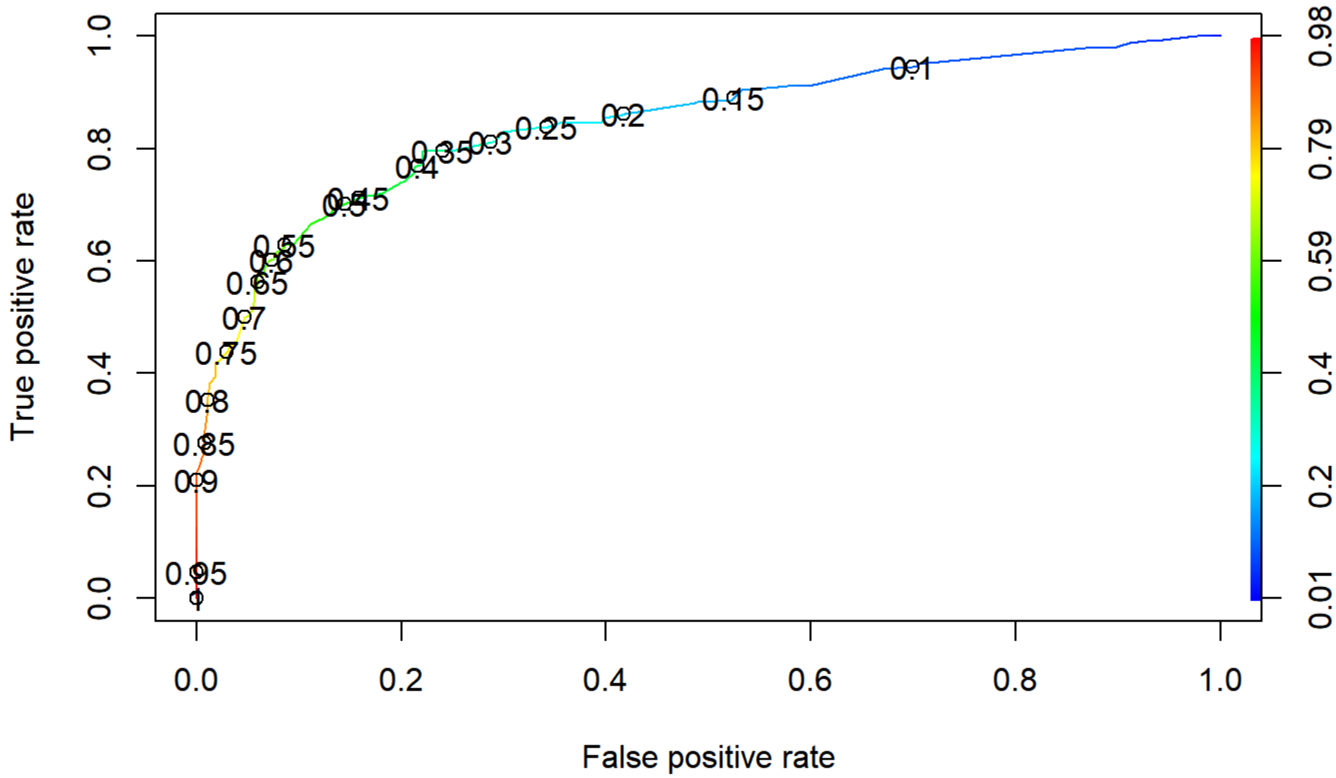
```
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

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

```
pred<-prediction(training_titanic$predicted_prob,training_titanic$Survived)
perf<-performance(pred,"tpr","fpr")
plot(perf,colorize=T,print.cutoffs.at=seq(0.1,by=0.05))
```



After looking at the graph we assign cutoff as 0.45 Hence we find the probabilties based on this cutoff

```
# we add a new column which has survival as 0 or 1
training_titanic$predicted_survived<-ifelse(training_titanic$predicted_prob<0.45,0,1)
head(training_titanic)
```

```
## # A tibble: 6 x 8
##   Survived Pclass Sex   Age   SibSp[,1] Embarked predicted_prob
##   <fct>    <fct>  <fct> <fct>     <dbl> <fct>             <dbl>
## 1 0        3      male  c2        0.433 S                0.0589
## 2 1        3      fema~ c2       -0.474 S                0.530
## 3 1        1      fema~ c3        0.433 S                0.908
## 4 0        3      male  c3       -0.474 S                0.102
## 5 0        3      male  c2       -0.474 Q                0.152
## 6 0        1      male  c4       -0.474 S                0.300
## # ... with 1 more variable: predicted_survived <dbl>
```

Confusion matrix:

```
table(training_titanic$Survived,training_titanic$predicted_survived)
```

```
##
##       0   1
##   0 326  59
##   1  70 170
```

Another way to get Accuracy and other measures is:

```
# first we convert the predicted_survived column to factor
training_titanic$predicted_survived<-as.factor(training_titanic$predicted_survived)

library(caret)
confusionMatrix(training_titanic$predicted_survived,training_titanic$Survived)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 326  70
##          1  59 170
##
##                Accuracy : 0.7936
##                  95% CI : (0.7597, 0.8247)
##     No Information Rate : 0.616
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.5599
##
##  Mcnemar's Test P-Value : 0.3786
##
##             Sensitivity : 0.8468
##             Specificity : 0.7083
##          Pos Pred Value : 0.8232
##          Neg Pred Value : 0.7424
##              Prevalence : 0.6160
##          Detection Rate : 0.5216
##    Detection Prevalence : 0.6336
##       Balanced Accuracy : 0.7775
##
##        'Positive' Class : 0
##
```

Applying the same logic for testing set:

```
testing_titanic$predicted_prob<-predict(titanic_model,testing_titanic,type = "response")
testing_titanic$predicted_survived<-ifelse(testing_titanic$predicted_prob<0.45,0,1)

testing_titanic$predicted_survived<-as.factor(testing_titanic$predicted_survived)

confusionMatrix(testing_titanic$predicted_survived,testing_titanic$Survived)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 141  26
##          1  23  76
##
##                Accuracy : 0.8158
##                  95% CI : (0.7639, 0.8605)
##     No Information Rate : 0.6165
##     P-Value [Acc > NIR] : 1.592e-12
##
##                   Kappa : 0.6082
##
##  Mcnemar's Test P-Value : 0.7751
##
##             Sensitivity : 0.8598
##             Specificity : 0.7451
##          Pos Pred Value : 0.8443
##          Neg Pred Value : 0.7677
##              Prevalence : 0.6165
##          Detection Rate : 0.5301
##    Detection Prevalence : 0.6278
##       Balanced Accuracy : 0.8024
##
##        'Positive' Class : 0
##
```

Applying Random Forest:

```
# first we remove the extra columns we created
training_titanic<-training_titanic[,1:6]
testing_titanic<-testing_titanic[,1:6]

library(randomForest)
```

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

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

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
rf<-randomForest(Survived~.,data = training_titanic,ntree=60)

pred_test_rf<-predict(rf,testing_titanic)

confusionMatrix(pred_test_rf,testing_titanic$Survived)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 154  37
##          1  10  65
##
##                Accuracy : 0.8233
##                  95% CI : (0.7721, 0.8672)
##     No Information Rate : 0.6165
##     P-Value [Acc > NIR] : 1.974e-13
##
##                   Kappa : 0.6066
##
##  Mcnemar's Test P-Value : 0.0001491
##
##             Sensitivity : 0.9390
##             Specificity : 0.6373
##          Pos Pred Value : 0.8063
##          Neg Pred Value : 0.8667
##              Prevalence : 0.6165
##          Detection Rate : 0.5789
##    Detection Prevalence : 0.7180
##       Balanced Accuracy : 0.7881
##
##        'Positive' Class : 0
##
```

Applying SVC:

```
library(e1071)
svc<-svm(Survived~.,data = training_titanic, kernel='poly', degree=3)
pred_svc<-predict(svc,testing_titanic)
confusionMatrix(pred_svc,testing_titanic$Survived)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 146  28
##          1  18  74
##
##                Accuracy : 0.8271
##                  95% CI : (0.7762, 0.8705)
##     No Information Rate : 0.6165
##     P-Value [Acc > NIR] : 6.692e-14
##
##                   Kappa : 0.6274
##
##  Mcnemar's Test P-Value : 0.1845
##
##             Sensitivity : 0.8902
##             Specificity : 0.7255
##          Pos Pred Value : 0.8391
##          Neg Pred Value : 0.8043
##              Prevalence : 0.6165
##          Detection Rate : 0.5489
##    Detection Prevalence : 0.6541
##       Balanced Accuracy : 0.8079
##
##        'Positive' Class : 0
##
```

Applying Naive Bayes:

```
nvb<-naiveBayes(Survived~.,data = training_titanic)
pred_nvb<-predict(nvb,testing_titanic)
confusionMatrix(pred_nvb,testing_titanic$Survived)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0 135  23
##          1  29  79
##                                           
##                Accuracy : 0.8045          
##                  95% CI : (0.7517, 0.8504)
##     No Information Rate : 0.6165          
##     P-Value [Acc > NIR] : 3.037e-11       
##                                           
##                   Kappa : 0.5911          
##                                           
##  Mcnemar's Test P-Value : 0.4881          
##                                           
##             Sensitivity : 0.8232          
##             Specificity : 0.7745          
##          Pos Pred Value : 0.8544          
##          Neg Pred Value : 0.7315          
##              Prevalence : 0.6165          
##          Detection Rate : 0.5075          
##    Detection Prevalence : 0.5940          
##       Balanced Accuracy : 0.7988          
##                                           
##        'Positive' Class : 0               
## 
```

Conclusion:

We observe that the Support vector classifier yeilds us an accuracy of 82.71% and also the difference between sensitivity and specificity is less as compared the difference between them in random forest. Hence we conclude SVC model best for predicting the Survival class of the Titatnic data.