Titanic Survival

Problem Description: Titanic - Machine Learning from Disaster

The dataset contains information on passengers of the RMS Titanic which sank in 1912 after colliding with an iceberg. Around 2/3 of the people onboard were killed with groups of people, such as women, children or upper-class, having a much higher rate of survival than others. The aim of the challenge is to most accurately predict for each passenger whether the passenger survived or not.
The data is comprised of fields to identify each passenger by name and id as well as his/her gender, age, travel class, number of family relations on board, cabin number and travel details such as fare, embarkation port and ticket number. The training set also has a value of 1 if the passenger survived or 0 if he/she didn't. This value is missing in the test set and the percentage of correct predictions of this value is used as the measure of quality for the analysis.

```
# Loading and processing the data set
passengers <- read.csv("train.csv")
passengers$Name <- as.character(passengers$Name)
passengers$Pclass <- as.factor(passengers$Pclass)
drops <- c("Name","Ticket","Embarked")
passengers <- passengers[,!(names(passengers) %in% drops)]
```

I decided to immediately drop the information on Name, Ticket number and place of Embarkation. It doesn't seem logical to me that these would provide any helpful input to the analysis.

Exploration of the data

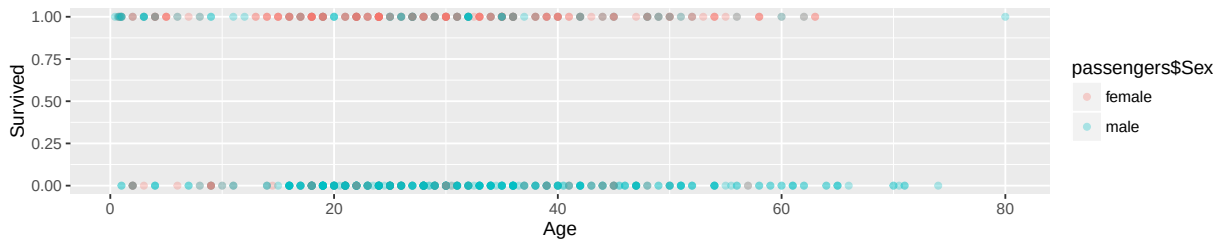
Overview of survival based on sex, age and class

```
table(passengers$Sex,passengers$Survived)
```

```
##
##              0   1
##   female  81 233
##   male   468 109
```

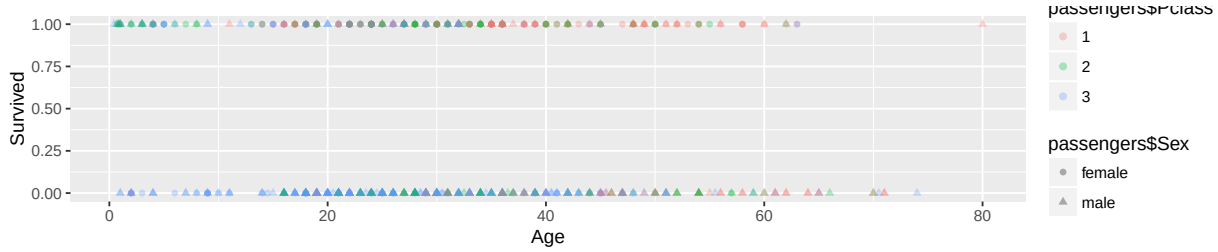
```
qplot(Age, Survived, data=passengers, alpha=I(0.3), color=passengers$Sex)
```

```
## Warning: Removed 177 rows containing missing values (geom_point).
```



```
qplot(Age, Survived, data=passengers, alpha=I(0.3), color=passengers$Pclass,
      pch=passengers$Sex)
```

```
## Warning: Removed 177 rows containing missing values (geom_point).
```

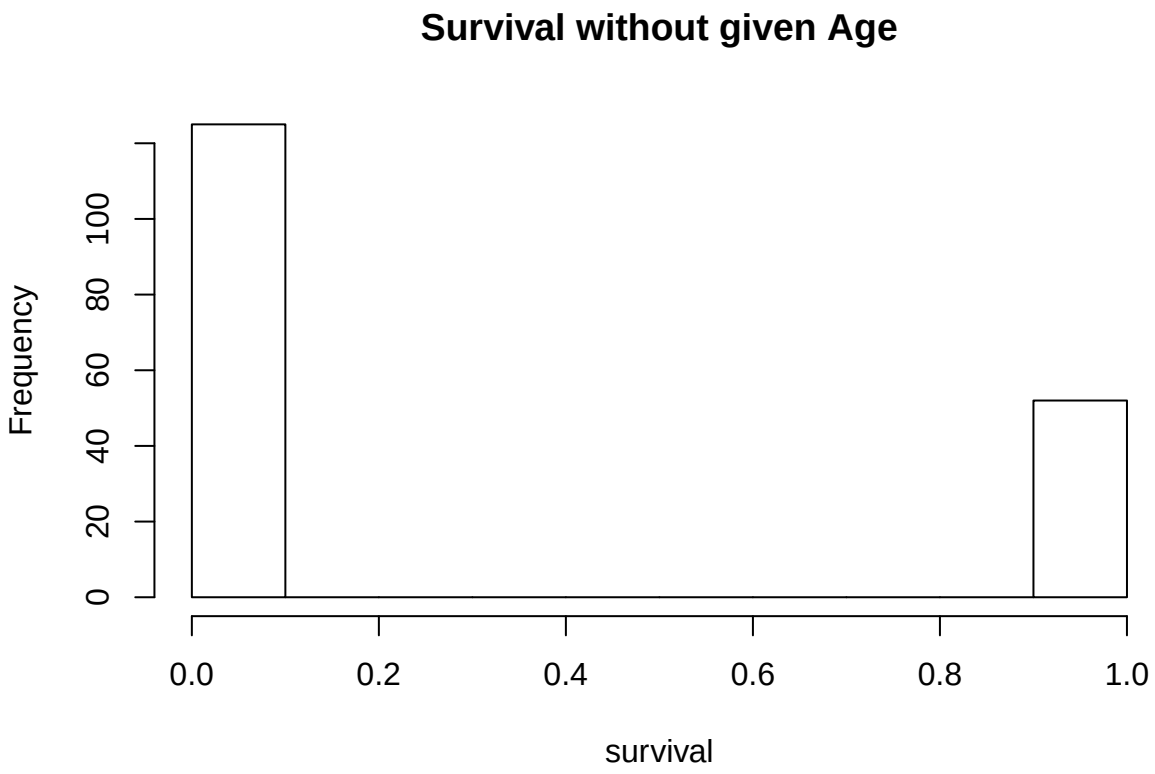


Based on the initial exploration, it appears that most adult men died while most adult women survived. Most children of class 1 or 2 seem to have survived and very few passengers of class 1 that were younger than 45 died. As a first approach to analysing the problem, I've decided to set up a rule based system following these observation:

1. If the passenger is male
    - and older than 16 or class 3, he died
    - else he survived
2. If the passenger is female
    - and younger than 10 and class 3, she died
    - else she survived

Since roughly 20% of the Age values are missing, it might be interesting to see the relationship between missing Age value and survival:

```
hist(passengers$Survived[which(is.na(passengers$Age))], main="Survival without given Age", xlab="surviva
```



As it is more likely that Age data was missing for passengers of class 3, it makes sense for a simple approach to just replace it with an arbitrary adult age. About 2/3 of people missing age died and it is likely that ones that did survive were adult women.

```
#replace NA values for this analysis
nonapassengers <- passengers
nonapassengers$Age[which(is.na(passengers$Age))] <- 20

#helper function for rules
survivalbyrules <- function(x){
    if(x['Sex'] == "male"){
        if(x['Age'] > 16 | x['Pclass'] == 3)
            return (0);
    }
    else if(x['Sex'] == "female"){
        if(x['Age'] <= 10 & x['Pclass'] == 3)
            return (0);
    }
    return (1);
}

#making predictions based on rules
nonapassengers$PredictedSurvival <- apply(nonapassengers,1,survivalbyrules)

confusionMatrix(table(nonapassengers$Survived,
                      nonapassengers$PredictedSurvival))
```

```
## Confusion Matrix and Statistics
##
##
##       0   1
##   0 478  71
##   1 108 234
##
##                Accuracy : 0.7991
##                  95% CI : (0.7713, 0.8249)
##     No Information Rate : 0.6577
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5664
##  Mcnemar's Test P-Value : 0.007129
##
##             Sensitivity : 0.8157
##             Specificity : 0.7672
##          Pos Pred Value : 0.8707
##          Neg Pred Value : 0.6842
##              Prevalence : 0.6577
##          Detection Rate : 0.5365
##    Detection Prevalence : 0.6162
##       Balanced Accuracy : 0.7915
##
##        'Positive' Class : 0
##
```

The prediction accuracy of this approach is 79.9% which turns out to give a surprisingly good result given that it's based on a very small and simple set of rules. Obviously, most problems follow a simple logic that can easily be grasped from a few two-dimensional plots and furthermore, there is still an error rate of slightly more than 20%. So, there's definitely room for improvement. I think incorporating the information on family relationships might help to make a better prediction.

```
set.seed(111)

#partitinioning the training data
inT <- createDataPartition(passengers$PassengerId,times=1,p=0.8, list=FALSE)
passengersTrain <- nonapassengers[inT,]
passengersTest <- passengers[-inT,]

#model and prediction based on rpart function
modelrpart <- rpart(formula(Survived ~ Pclass + Sex + Age + SibSp
                            + Parch + Fare),
                    method="class",data=passengersTrain)

print(modelrpart)
```

Rpart

```
## n= 715
##
```

```
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##   1) root 715 282 0 (0.60559441 0.39440559)
##     2) Sex=male 457  91 0 (0.80087527 0.19912473)
##       4) Age>=13 422  70 0 (0.83412322 0.16587678) *
##       5) Age< 13 35  14 1 (0.40000000 0.60000000)
##        10) SibSp>=2.5 14   1 0 (0.92857143 0.07142857) *
##        11) SibSp< 2.5 21   1 1 (0.04761905 0.95238095) *
##     3) Sex=female 258  67 1 (0.25968992 0.74031008)
##       6) Pclass=3 117  58 1 (0.49572650 0.50427350)
##        12) Fare>=24.80835 15   1 0 (0.93333333 0.06666667) *
##        13) Fare< 24.80835 102  44 1 (0.43137255 0.56862745)
##          26) Age>=27.5 20   6 0 (0.70000000 0.30000000) *
##          27) Age< 27.5 82  30 1 (0.36585366 0.63414634)
##            54) Fare>=7.9021 49  22 1 (0.44897959 0.55102041)
##             108) Fare< 10.825 14   4 0 (0.71428571 0.28571429) *
##             109) Fare>=10.825 35  12 1 (0.34285714 0.65714286)
##               218) Fare>=13.93545 28  12 1 (0.42857143 0.57142857)
##                 436) Fare< 15.3729 8   1 0 (0.87500000 0.12500000) *
##                 437) Fare>=15.3729 20   5 1 (0.25000000 0.75000000) *
##               219) Fare< 13.93545 7   0 1 (0.00000000 1.00000000) *
##            55) Fare< 7.9021 33   8 1 (0.24242424 0.75757576) *
##       7) Pclass=1,2 141   9 1 (0.06382979 0.93617021) *
```

```
predsurvrpart <- predict(modelrpart,passengersTest)
predsurvrpart <- max.col(predsurvrpart)
predsurvrpart <- predsurvrpart - 1
confusionMatrix(table(passengersTest$Survived,predsurvrpart))
```

```
## Confusion Matrix and Statistics
##
##    predsurvrpart
##       0   1
##   0 113   3
##   1  23  37
##
##                Accuracy : 0.8523
##                  95% CI : (0.7911, 0.9012)
##     No Information Rate : 0.7727
##     P-Value [Acc > NIR] : 0.0057645
##
##                   Kappa : 0.6425
##  Mcnemar's Test P-Value : 0.0001944
##
##             Sensitivity : 0.8309
##             Specificity : 0.9250
##          Pos Pred Value : 0.9741
##          Neg Pred Value : 0.6167
##              Prevalence : 0.7727
##          Detection Rate : 0.6420
##    Detection Prevalence : 0.6591
##       Balanced Accuracy : 0.8779
##
```

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

```
#partitinioning the training data using the data with imputed Age values
passengersTrain <- nonapassengers[inT,]

#model and prediction based on rpart function
modelforest <- randomForest(formula(as.factor(Survived) ~ Pclass + Sex + Age
                                    + SibSp + Parch), data=passengersTrain)

predsurvforest <- predict(modelforest,passengersTest)
predsurvforest[which(is.na(predsurvforest))] <- 0
confusionMatrix(table(passengersTest$Survived,predsurvforest))
```

Random Forest

```
## Confusion Matrix and Statistics
##
##    predsurvforest
##       0   1
##   0 110   6
##   1  26  34
##
##                Accuracy : 0.8182
##                  95% CI : (0.7531, 0.8722)
##     No Information Rate : 0.7727
##     P-Value [Acc > NIR] : 0.0862223
##
##                   Kappa : 0.56
##  Mcnemar's Test P-Value : 0.0007829
##
##             Sensitivity : 0.8088
##             Specificity : 0.8500
##          Pos Pred Value : 0.9483
##          Neg Pred Value : 0.5667
##              Prevalence : 0.7727
##          Detection Rate : 0.6250
##    Detection Prevalence : 0.6591
##       Balanced Accuracy : 0.8294
##
##        'Positive' Class : 0
##
```

```
#model and prediction based on rpart function
modelforest2 <- randomForest(formula(as.factor(Survived) ~ Pclass + Sex + Age
                                     + SibSp + Fare),
               data=passengersTrain)

predsurvforest2 <- predict(modelforest2,passengersTest)
predsurvforest2[which(is.na(predsurvforest2))] <- 0
confusionMatrix(table(passengersTest$Survived,predsurvforest2))
```

```
## Confusion Matrix and Statistics
##
##    predsurvforest2
##       0   1
##   0 111   5
##   1  24  36
##
##                Accuracy : 0.8352
##                  95% CI : (0.772, 0.8868)
##     No Information Rate : 0.767
##     P-Value [Acc > NIR] : 0.0173847
##
##                   Kappa : 0.603
##  Mcnemar's Test P-Value : 0.0008302
##
##             Sensitivity : 0.8222
##             Specificity : 0.8780
##          Pos Pred Value : 0.9569
##          Neg Pred Value : 0.6000
##              Prevalence : 0.7670
##          Detection Rate : 0.6307
##    Detection Prevalence : 0.6591
##       Balanced Accuracy : 0.8501
##
##        'Positive' Class : 0
##
```

Of all the more sophisticated approaches that I tried, the one using rpart to figure out the internal rule system gives the best result at around 85%. The rules show that both the Fare and the relationship to siblings or spouses on board also play a significant role when determining survival. For boys, having fewer siblings led to improved chances of survival which seems logical. For women in class 3, the decision is very finely tuned to different Fare categories which might indicate a problem with overfitting.

Training best method on the full dataset and given test set

```
#parsing test dataset
testset <- read.csv("test.csv")
testset$Name <- as.character(testset$Name)
testset$Pclass <- as.factor(testset$Pclass)
drops <- c("Name","Ticket","Embarked")
testset <- testset[,!(names(testset) %in% drops)]
#model and prediction based on rpart function
modelfinal <- rpart(formula(Survived ~ Pclass + Sex + Age + SibSp
                            + Fare ),
                    method="class",data=passengers)

print(modelfinal)
```

```
## n= 891
##
## node), split, n, loss, yval, (yprob)
```

```
##       * denotes terminal node
##
##   1) root 891 342 0 (0.61616162 0.38383838)
##     2) Sex=male 577 109 0 (0.81109185 0.18890815)
##       4) Age>=6.5 553  93 0 (0.83182640 0.16817360) *
##       5) Age< 6.5 24   8 1 (0.33333333 0.66666667)
##        10) SibSp>=2.5 9   1 0 (0.88888889 0.11111111) *
##        11) SibSp< 2.5 15   0 1 (0.00000000 1.00000000) *
##     3) Sex=female 314  81 1 (0.25796178 0.74203822)
##       6) Pclass=3 144  72 0 (0.50000000 0.50000000)
##        12) Fare>=23.35 27   3 0 (0.88888889 0.11111111) *
##        13) Fare< 23.35 117  48 1 (0.41025641 0.58974359)
##          26) Age>=16.5 93  42 1 (0.45161290 0.54838710)
##            52) Fare>=7.8875 56  25 0 (0.55357143 0.44642857)
##             104) Fare< 14.8729 33  10 0 (0.69696970 0.30303030) *
##             105) Fare>=14.8729 23   8 1 (0.34782609 0.65217391) *
##            53) Fare< 7.8875 37  11 1 (0.29729730 0.70270270) *
##          27) Age< 16.5 24   6 1 (0.25000000 0.75000000) *
##       7) Pclass=1,2 170   9 1 (0.05294118 0.94705882) *
```

```r
Survived <- predict(modelfinal,testset)
Survived <- max.col(Survived)
Survived <- Survived - 1
testset <- cbind(testset,Survived)
keep <- c("PassengerId","Survived")
testoutput <- testset[,keep]
write.csv(testoutput,file="prediction3.csv",row.names = FALSE, quote = FALSE)
```