Kaggle: Titanic

Suman Adhikari

August 14, 2017

Executive Summary:

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 challenge, we will perform analysis of what sorts of people were likely to survive. In particular, we have done feature engineering, imputation of missing values and random forest machine learning to predict which passengers survived the tragedy.

Libaries and Data

Variables and their description:

Variable	Definition	Key
survival	Survival	0 if Survival = No, 1 if Survival = Yes
pclass	Ticket class accorinding to socio-economic status	1 = Upper, 2 = Middle, 3 = Lower
sex	Sex	
Age	Age of passengers in years	
sibsp	No. of siblings / spouses aboard the Titanic	
parch	No. of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

```
#### Load libraries
library(caret)
library(randomForest)
library(rpart)
library(rpart.plot)
library(corrplot)
library(dplyr)
genderSub <- read.csv("/R/workspace/kaggle/titanic/gender_submission.csv", sep = ",",
    stringsAsFactors = F, header = T)
testData <- read.csv("/R/workspace/kaggle/titanic/test.csv", sep = ",", stringsAsFactors = F,
    header = T)
trainData <- read.csv("/R/workspace/kaggle/titanic/train.csv", sep = ",", stringsAsFactors = F,
    header = T)
```

```r
testData$Survived <- NA

## Tagging Data
trainData$sample = "training"
testData$sample = "testing"

## Merging the traning and testing data
trailData <- bind_rows(trainData, testData)
tempData <- trailData

## Changing Veriables to factor
trailData$Pclass <- as.factor(trailData$Pclass)
trailData$Sex <- as.factor(trailData$Sex)
trailData$Embarked <- as.factor(trailData$Embarked)

## Results hidden. Please refer appendix
str(trailData)
summary(trailData)
```

Cleaning Data and Imputation

From the summary of the data above we got our data have some missing values. In this section we are going to fix and imputate the missing values.

Cleaning and imputation of AGE

From Summary of the data we can observe that the Age variable has some missing values. We will be developing and implementing workaround to deal with missing values.

```r
sum(is.na(trailData$Age))
## [1] 263
```

To impute the age, lets introduce a new variable Age categorey using below rule:

1) If (number of spouses or siblings > 1) then it's probably a child and represented by 0
2) If (number of parents or children > 2) then it's probably an adult and represented by 1
3) If all above conditions are not met lets call it is unterermined and represented by 2

```r
trailData <- mutate(trailData, AgeCat = ifelse(SibSp > 1, 0, ifelse(Parch >
    2, 1, 2)))
```

Building model to predict age.

```r
ageModel = rpart(Age ~ Fare + Pclass + SibSp + Parch + AgeCat, data = trailData)
trailData$predictAge = predict(ageModel, trailData)
trailData$Age <- ifelse(is.na(trailData$Age), trailData$predictAge, trailData$Age)

## Check if any NA values are there
numOfNa <- sum(is.na(trailData$Age))
numOfNa
## [1] 0
```

With successful imputation of age variables, there are 0 NA's in AGE variables.

Cleaning and imputation of Embarked

There are smome empty values for variable Embarked. In this section we will expore and clean the missing data.

```
## Check if Embarked is missing & if missing show respective index
which(trailData$Embarked == "")
## [1]  62 830
```

From above we can see two values for embarked missing. We will use the existing data to predict the empty data.

```
trailData[trailData$Embarked == "", ]
##     PassengerId Survived Pclass                                      Name
## 62           62        1      1                       Icard, Miss. Amelie
## 830         830        1      1 Stone, Mrs. George Nelson (Martha Evelyn)
##        Sex Age SibSp Parch Ticket Fare Cabin Embarked   sample AgeCat
## 62  female  38     0     0 113572   80   B28          training      2
## 830 female  62     0     0 113572   80   B28          training      2
##     predictAge
## 62    40.54743
## 830   40.54743
```

Going in detail we find that the data with missing embarked have same ticked num and same cabin are from same passenger class. Hence lets build a model based on fare and passenger class.

```
trailData$Embarked <- as.factor(trailData$Embarked)
emptyEmbarked <- which(trailData$Embarked == "")
embarkedModel <- train(Embarked ~ Pclass + Fare, data = trailData, method = "rpart",
    na.action = na.pass)
trailData$Embarked[emptyEmbarked] <- predict(embarkedModel, trailData[emptyEmbarked,
    ])
```

Cleaning and imputation of Fare

Now its turn to clean and impute missing values for Fare variables. There is 1 missing values for Fear variable. First let's look for the while row.

```
sum(is.na(trailData$Fare))
## [1] 1
trailData[is.na(trailData$Fare), ]
##      PassengerId Survived Pclass               Name  Sex  Age SibSp Parch
## 1044        1044       NA      3 Storey, Mr. Thomas male 60.5     0     0
##      Ticket Fare Cabin Embarked  sample AgeCat predictAge
## 1044   3701   NA              S testing      2   27.43182
```

We are going to build a model to predict the fare on basis of variables:

```
fareModel <- rpart(Fare ~ Age + Pclass + Embarked + SibSp + Parch, data = trailData)
emptyFare <- which(is.na(trailData$Fare))
trailData$Fare[emptyFare] <- predict(fareModel, trailData[emptyFare, ])
sum(is.na(trailData$Fare))
## [1] 0
```

Cleaning and imputation of Cabin

First lets count the number of missing values for the Cabin variable.

```r
count <- length(which(trailData$Cabin == ""))
count
## [1] 1014
```

The toatl empty values for variable Cabin is: 1014. There are too many empty values so we will drop this feature entirely to avoid the risk of adding noise by filling in predicted values.

Exploratory Data analysis

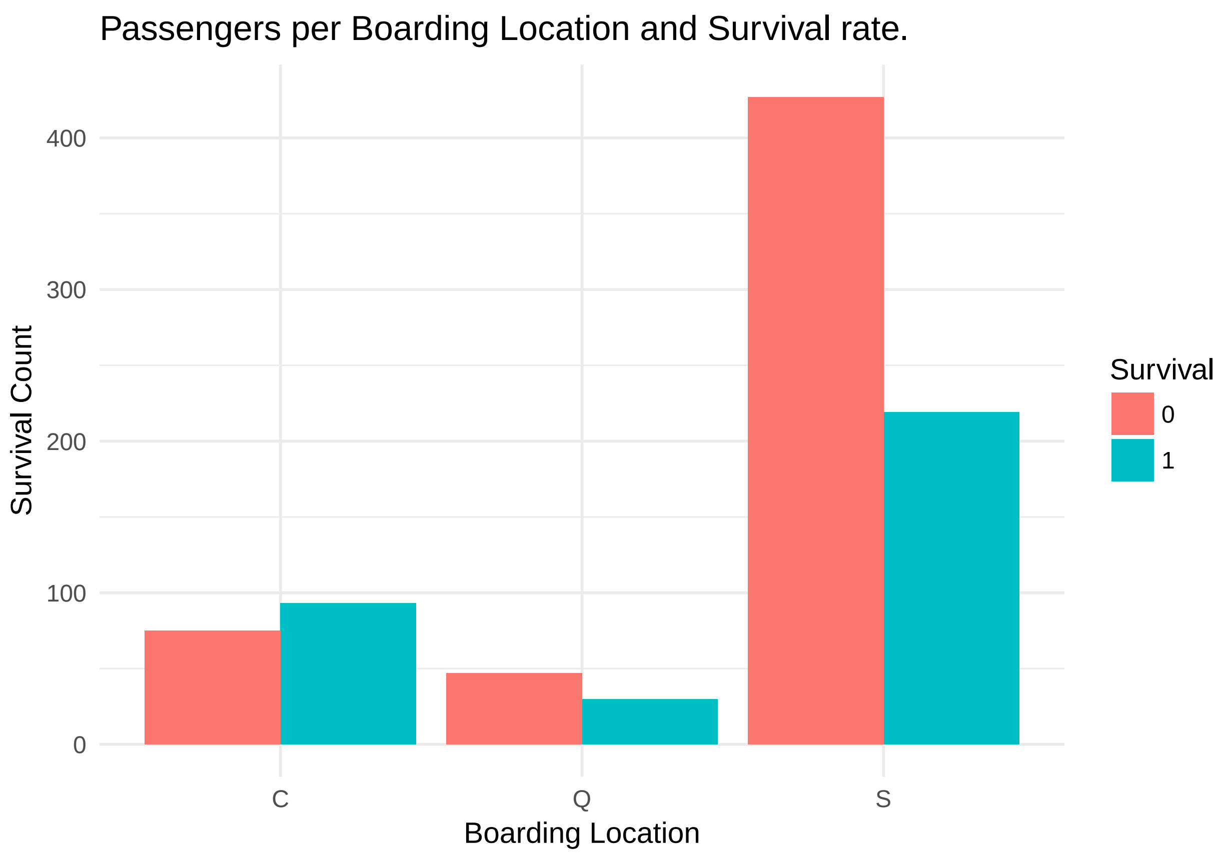
In this section we will visualize the given data for some exploratory data analysis and visualize relationship between other features and Survival. All the plots are based train data.

Men and Women Survived by Passenger Class

We had draw a plot on for Men/Women Survived by passengre class. For the plot below we can visualize that there were more female survival for each passenger class. [Refer appendix for code.]

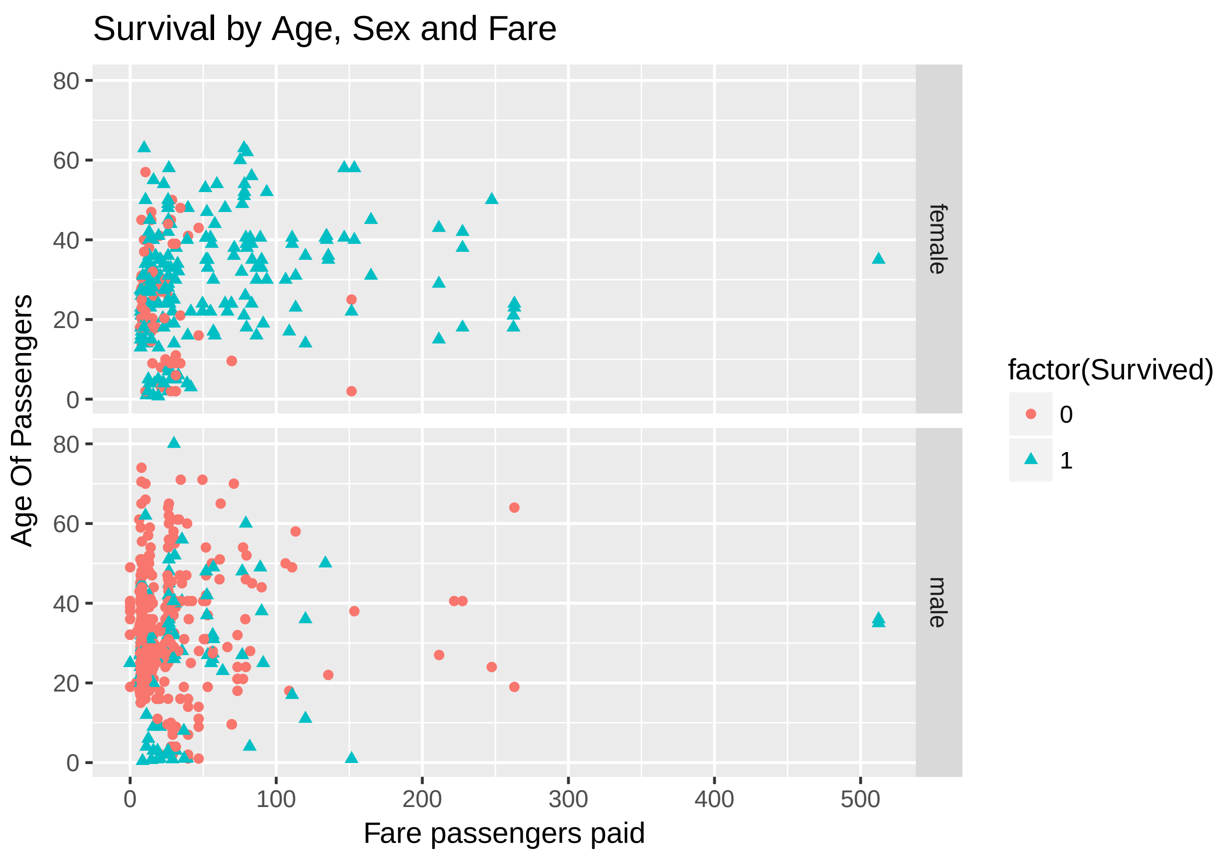
Passengers per Boarding Location

Next we have ploted plot with passengers with boarding location and their survival count and there ratio of survial was high for boarding location “C”. [Refer appendix for code.]



Survival by Age, Sex and Fare

From plot below, Survival by Age, Sex and Fare we can obeserve female passengers with age 16-50 survived more and also who paid high fare survivied more. [Refer appendix for code.]



Feature Engineering:

Stronger vs Weak

It is obvious that, we can assume that stronger can swim more than weaker ones. So I would like to categorize age into Child/Adult/Old variables. Categorizing is done:

1. age <= 50 => Old
2. age < 16 => Child
3. Other => Adult

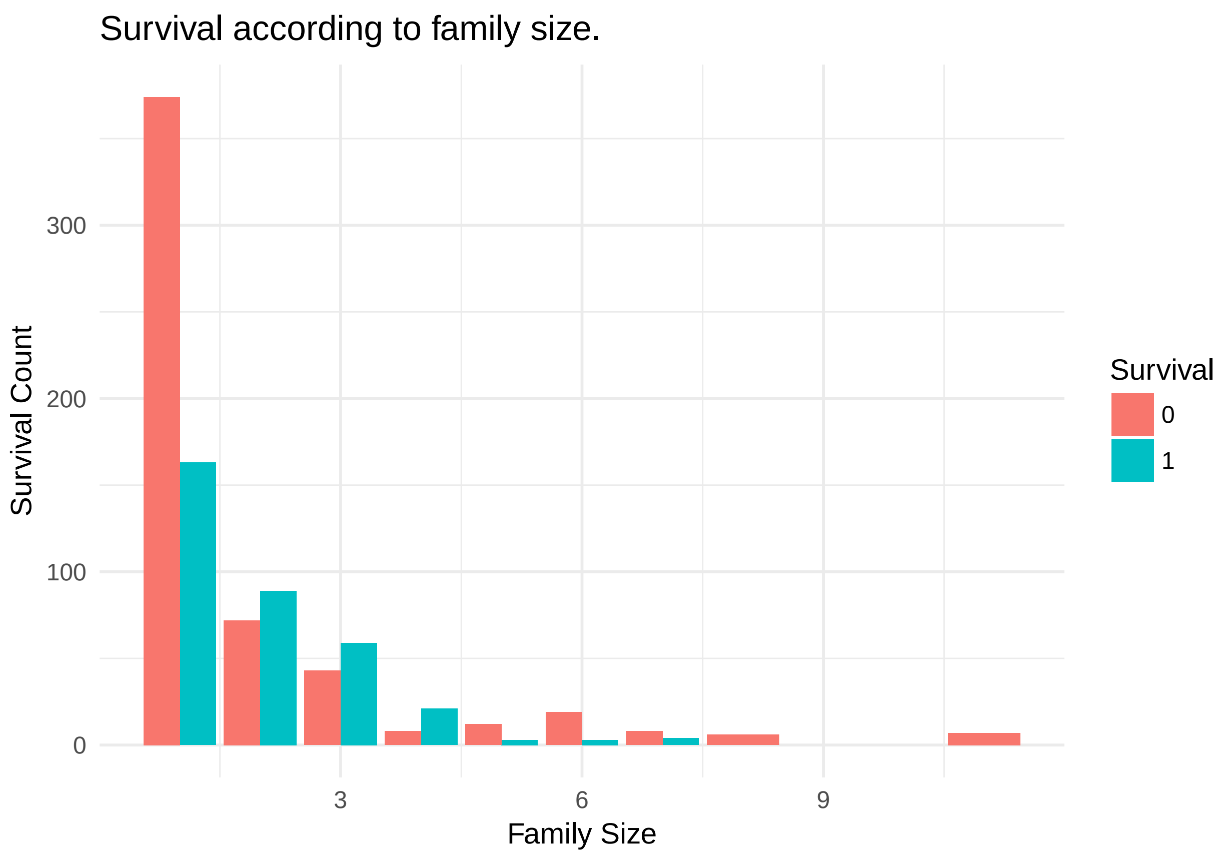
```
trailData <- mutate(trailData, fitness = as.factor(ifelse(trailData$Age <= 16,
    "Child", ifelse(trailData$Age >= 50, "Old", "Adult"))))

## View the Fitness Vs Sex distribution
table(trailData$Sex, trailData$fitness)
##
##          Adult Child Old
##   female   357    71  38
##   male     696    75  72
```

Now its more obvious that there were more Adult and Child survivals and can also be witnessed by below plot.

Family Size

Now I am focusing on the passengers if they are travelling in family and their survival rate. If they were travelling in family the male member of the family may give importance to other family members.

```
## Introduce new variable familySize, (+1 is for the individual him or her
## self)
trailData <- mutate(trailData, familySize = trailData$SibSp + trailData$Parch +
    1)
```



We can observe from the above barchart that the survival rate is high for fmily travelling alone or with fmaily size greater than 4. Hence we will take a step further and categorize the Family size according to following rule:

1. familySize = 1 => Single
2. familySize > 1 && <=4 => Small
3. familySize > 4 => Big

```
trailData <- mutate(trailData, familyType = as.factor(ifelse(familySize == 1,
    "Single", ifelse(familySize > 1 & familySize <= 4, "Small", "Big"))))
```

Passengers Name:

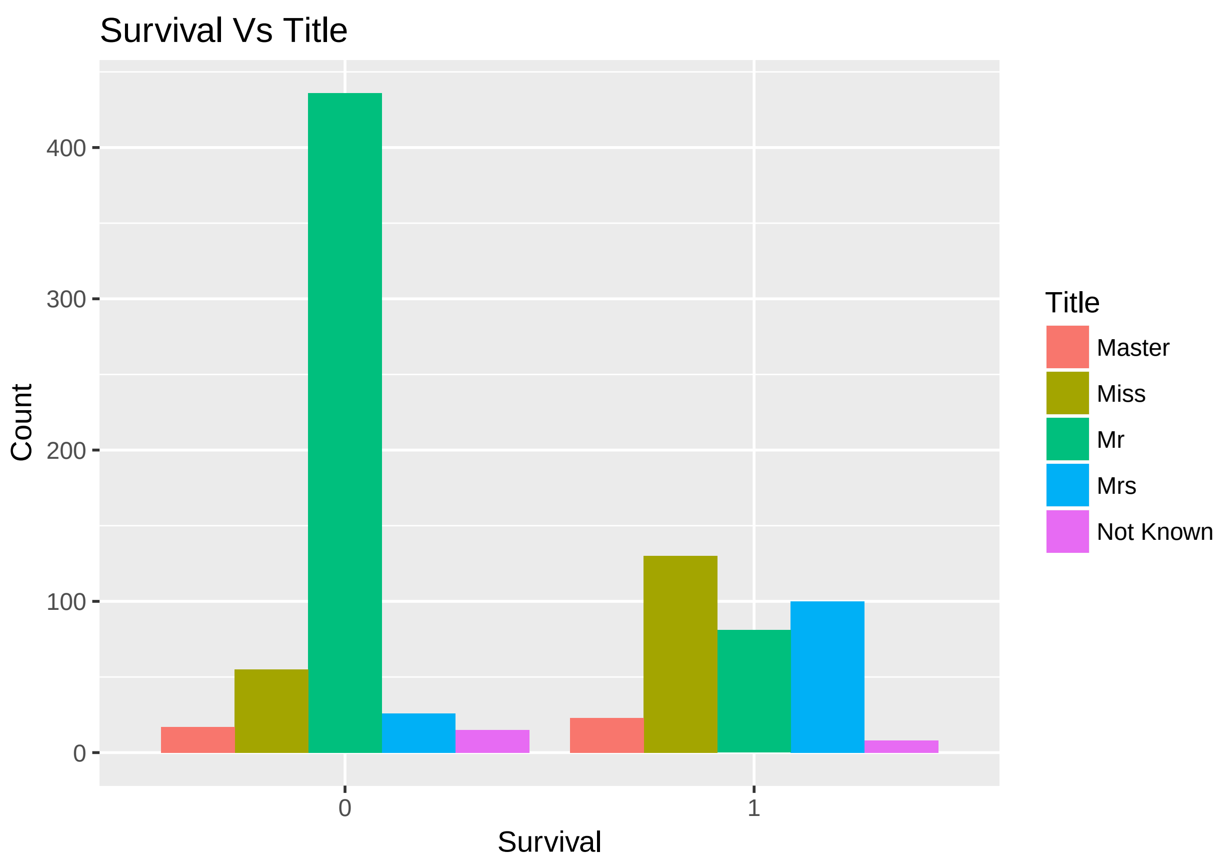
From plot Survival by Age, Sex and Fare, we got idea that female who paid high fare mostly survived. The Name variable consists of three parts Surname, Title and First Name. We are more interested in title rather than First Name and Surname. In this section we will break the name and extract title.

```
trailData$Title <- gsub("(.*, )|(\\..*)", "", trailData$Name)

## View the raw result: Title Vs Sex distribution
table(trailData$Sex, trailData$Title)
##
##          Capt Col Don Dona  Dr Jonkheer Lady Major Master Miss Mlle Mme
##   female    0   0   0    1   1        0    1     0      0  260    2   1
##   male      1   4   1    0   7        1    0     2     61    0    0   0
##
##           Mr Mrs  Ms Rev Sir the Countess
```

```
##   female    0 197   2   0   0            1
##   male    757   0   0   8   1            0
```

With Successful segregation of of title, lets combine similar titles.

```r
trailData$Title <- as.factor(ifelse(trailData$Title == "Mlle", "Miss", ifelse(trailData$Title ==
    "Ms", "Miss", ifelse(trailData$Title == "Mme", "Mrs", ifelse(trailData$Title ==
    "Miss", "Miss", ifelse(trailData$Title == "Mrs", "Mrs", ifelse(trailData$Title ==
    "Mr", "Mr", ifelse(trailData$Title == "Master", "Master", "Not Known"))))))))

## View the Title Vs Sex distribution
table(trailData$Sex, trailData$Title)
##
##          Master Miss  Mr Mrs Not Known
##   female      0  264   0 198         4
##   male       61    0 757   0        25
```



Prediction

We have cleaning and imputation of data, done some exploratory data analysis and feature engineering. In this section we will use the data and do prediction.

Slicing of data

As first step we will first slice the data, that before we merged in respective testing and train datasets.

```r
## Set the seed for reproducibililty
set.seed(22519)

## Splitting data sets to respective traning and test sets
tstData <- trailData[trailData$sample == "testing", ]
tranData <- trailData[trailData$sample == "training", ]
```

Data Modelling

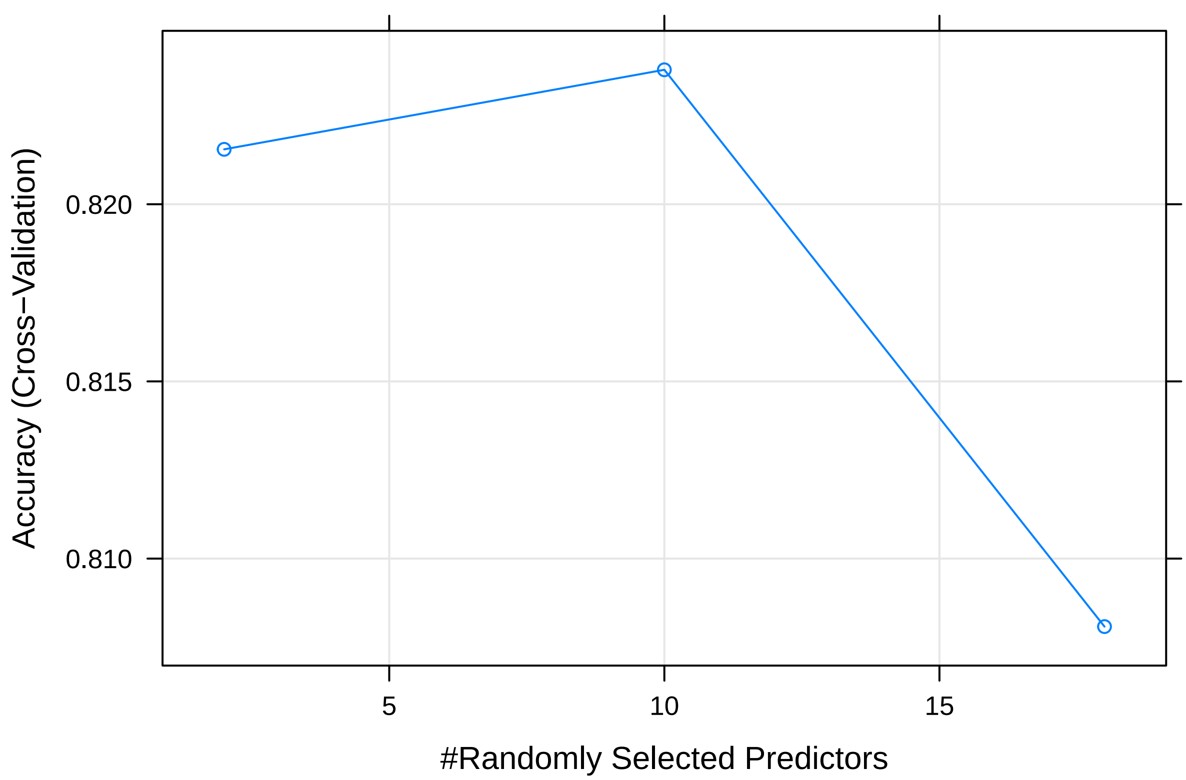
Now we will bulit model on basis of traning data set using Random Forest model with 3-fold cross validation.

```r
# Train the model using 3-fold CV
fitControl <- trainControl(method = "cv", number = 3, verboseIter = F)

fit <- train(as.factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch + Fare +
    Embarked + fitness + familyType + Title, data = tranData, method = "rf",
    trControl = fitControl, importance = TRUE)
```

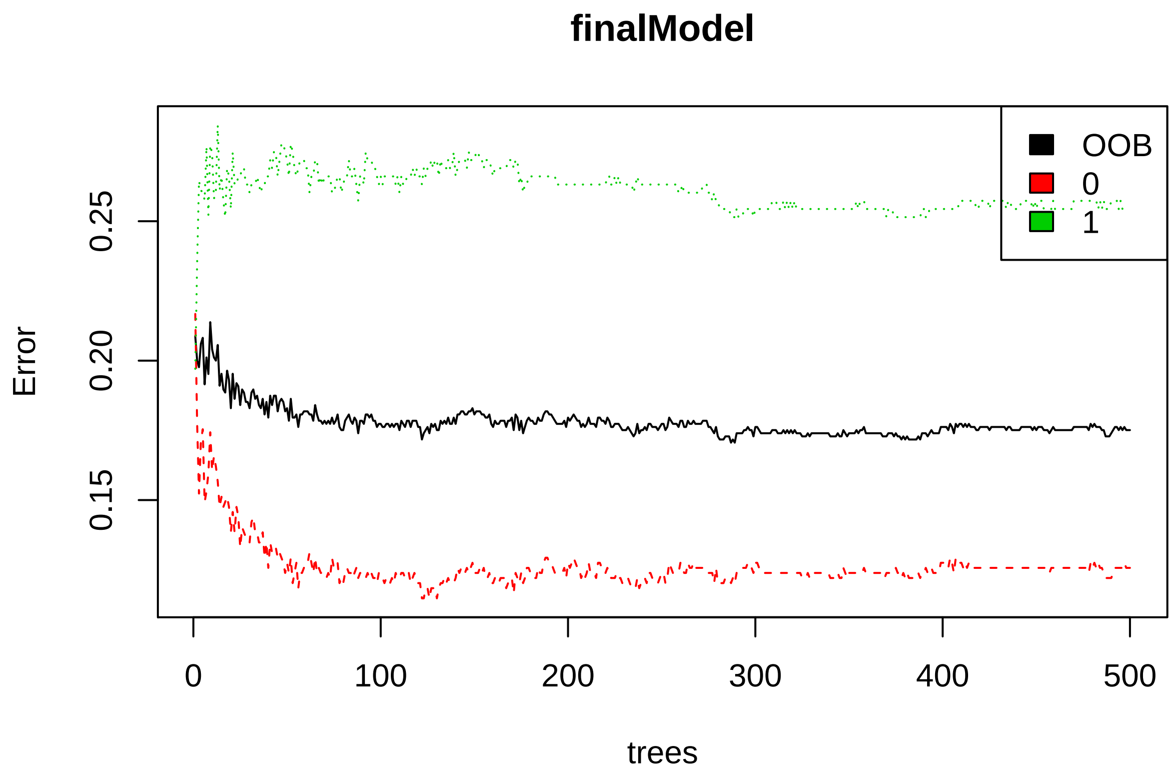
Evaluation and Selection of Model

With building the model now, we will use the model to predict the Survival of passenger from tstData and view the confusion matrix and make comparison wih actual Survival.

We can see from the plot below that the most accurate value for mtry was 10 with an accuracy of 82.38%.



Now lets look at the model error. Form the plot below we can observe the mean prediction error (OOB) and error rate for both survival and dead respectively in black, green and red for the model developed.



Lets print the whole mode. The mean prediction error (OOB) for the mode is 17.51%.

```
fit$finalModel
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 10
##
##         OOB estimate of  error rate: 17.51%
## Confusion matrix:
##     0   1 class.error
## 0 480  69   0.1256831
## 1  87 255   0.2543860
```

Prediction on Test Data set.

Now, I apply the model to the original testing data set downloaded from the data source and write those predictions to output file.

```
# Using model to predict Survival for test data set
predict <- predict(fit, newdata = tstData)
# predict
```

```r
solution <- data.frame(PassengerID = tstData$PassengerId, Survived = predict)
write.csv(solution, file = "predictionSurvivalTitanic.csv", row.names = F)
```

Conclusion

As the prediction of survival for titanic dataset is complete. Though mean prediction error is around 18% (approx), its sound for me to accept the solution. As I paced first step in machine learning, will apply more sophisticated algorithms to achive better and accurate predictions.

Appendix

Display the internal structure of an DataSet:

```
## 'data.frame':    1309 obs. of  13 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       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ Sex        : chr  "male" "female" "female" "female" ...
##  $ 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     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
##  $ sample     : chr  "training" "training" "training" "training" ...
```

Summary of the DataSet:

```
##   PassengerId      Survived          Pclass          Name          
##  Min.   :   1   Min.   :0.0000   Min.   :1.000   Length:1309       
##  1st Qu.: 328   1st Qu.:0.0000   1st Qu.:2.000   Class :character  
##  Median : 655   Median :0.0000   Median :3.000   Mode  :character  
##  Mean   : 655   Mean   :0.3838   Mean   :2.295                     
##  3rd Qu.: 982   3rd Qu.:1.0000   3rd Qu.:3.000                     
##  Max.   :1309   Max.   :1.0000   Max.   :3.000                     
##                 NA's   :418                                        
##      Sex                 Age            SibSp            Parch      
##  Length:1309        Min.   : 0.17   Min.   :0.0000   Min.   :0.000  
##  Class :character   1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.000  
##  Mode  :character   Median :28.00   Median :0.0000   Median :0.000  
##                     Mean   :29.88   Mean   :0.4989   Mean   :0.385  
##                     3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.000  
##                     Max.   :80.00   Max.   :8.0000   Max.   :9.000  
##                     NA's   :263                                     
##     Ticket               Fare            Cabin          
##  Length:1309        Min.   :  0.000   Length:1309       
##  Class :character   1st Qu.:  7.896   Class :character  
##  Mode  :character   Median : 14.454   Mode  :character  
##                     Mean   : 33.295                     
##                     3rd Qu.: 31.275                     
##                     Max.   :512.329                     
```

```
##                      NA's   :1
##    Embarked             sample
##  Length:1309        Length:1309
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
##
```

Summary of the Fit Model:

```
## Random Forest
##
## 891 samples
##  10 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 594, 594, 594
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.8215488  0.6099925
##   10    0.8237935  0.6206204
##   18    0.8080808  0.5882949
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 10.
```

R code: Men and Women Survived by Passenger Class

```
trnData <- trailData[trailData$sample == "training", ]
plot <- ggplot(trnData, aes(Pclass, fill = factor(Survived)))
plot <- plot + geom_bar(stat = "count", position = "dodge")
plot <- plot + facet_grid(Sex ~ .) + coord_flip()
plot <- plot + labs(title = "Men and Women Survived by Passenger Class", x = "Ticket class accorinding
    y = "Survival Count", fill = "Survival")
plot
```

R code: Passengers per Boarding Location and Survival rate.

```
plot2 <- ggplot(trnData, aes(x = Embarked, fill = factor(Survived)))
plot2 <- plot2 + geom_bar(stat = "count", position = "dodge")
plot2 <- plot2 + ggtitle("Passengers per Boarding Location and Survival rate.")
plot2 <- plot2 + ylab("Survival Count")
plot2 <- plot2 + xlab("Boarding Location") + theme_minimal()
plot2 <- plot2 + scale_fill_discrete(name = "Survival")
plot2
```

R code: Survival by Age, Sex and Fare.

```
plot <- ggplot(trnData, aes(x = Age, y = Fare))
plot <- plot + geom_point(aes(shape = factor(Survived), colour = factor(Survived)))
plot <- plot + facet_grid(Sex ~ .) + coord_flip()
plot <- plot + labs(title = "Survival by Age, Sex and Fare", x = "Age Of Passengers",
```

```
    y = "Fare passengers paid", fill = "Survival")
plot
```

R code: Men and Women Survived by Passenger Class and Physical fitness.

```
trnData <- trailData[trailData$sample == "training", ]
plot <- ggplot(trnData, aes(Pclass, fill = factor(Survived)))
plot <- plot + geom_bar(stat = "count", position = "dodge")
plot <- plot + facet_grid(Sex ~ fitness)
plot <- plot + labs(title = "Men and Women Survived by Passenger Class and Physical fitness",
    x = "Ticket class accorinding to socio-economic status", y = "Survival Count",
    fill = "Survival")
plot
```

R code: Survival according to family size.

```
trnData <- trailData[trailData$sample == "training", ]
plot1 <- ggplot(trnData, aes(x = familySize, fill = factor(Survived)))
plot1 <- plot1 + geom_bar(stat = "count", position = "dodge")
plot1 <- plot1 + ggtitle("Survival according to family size.")
plot1 <- plot1 + ylab("Survival Count")
plot1 <- plot1 + xlab("Family Size") + theme_minimal()
plot1 <- plot1 + scale_fill_discrete(name = "Survival")
plot1
```

R code: Survival Vs Title.

```
trnData <- trailData[trailData$sample == "training", ]
plot <- ggplot(trnData, aes(factor(Survived), fill = Title)) + geom_bar(position = "dodge")
plot <- plot + labs(title = "Survival Vs Title", x = "Survival", y = "Count",
    fill = "Title")
plot
```