My journey on the Titanic, final destination: Kaggle

Elisa Lerner

23 February 2016

Kaggle offer a machine learning competition called "Titanic - Machine Learning From Disaster".
They have made available training and test sets. The training set contains features and survival data for 891 of the passengers. The aim is to train a model which can predict survival outcomes for the 418 passengers in the test set. The score for the predictions is the accuracy rate.

Data

```
library(caret)
#load data
train_url<-"http://s3.amazonaws.com/assets.datacamp.com/course/Kaggle/train.csv"
test_url<- "http://s3.amazonaws.com/assets.datacamp.com/course/Kaggle/test.csv"
training<-read.csv(train_url)
test<-read.csv(test_url)
test<- data.frame(PassengerId=test[,1],Survived=rep("NA",418),test[,2:11])
```

The first stage is to discover which variables are in the dataset. According to the information on the kaggle website, the following variables are included:
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)

SPECIAL NOTES:
Pclass is a proxy for socio-economic status (SES) 1st ~ Upper; 2nd ~ Middle; 3rd ~ Lower

Age is in Years; Fractional if Age less than One (1) If the Age is Estimated, it is in the form xx.5

With respect to the family relation variables (i.e. sibsp and parch) some relations were ignored. The following are the definitions used for sibsp and parch.

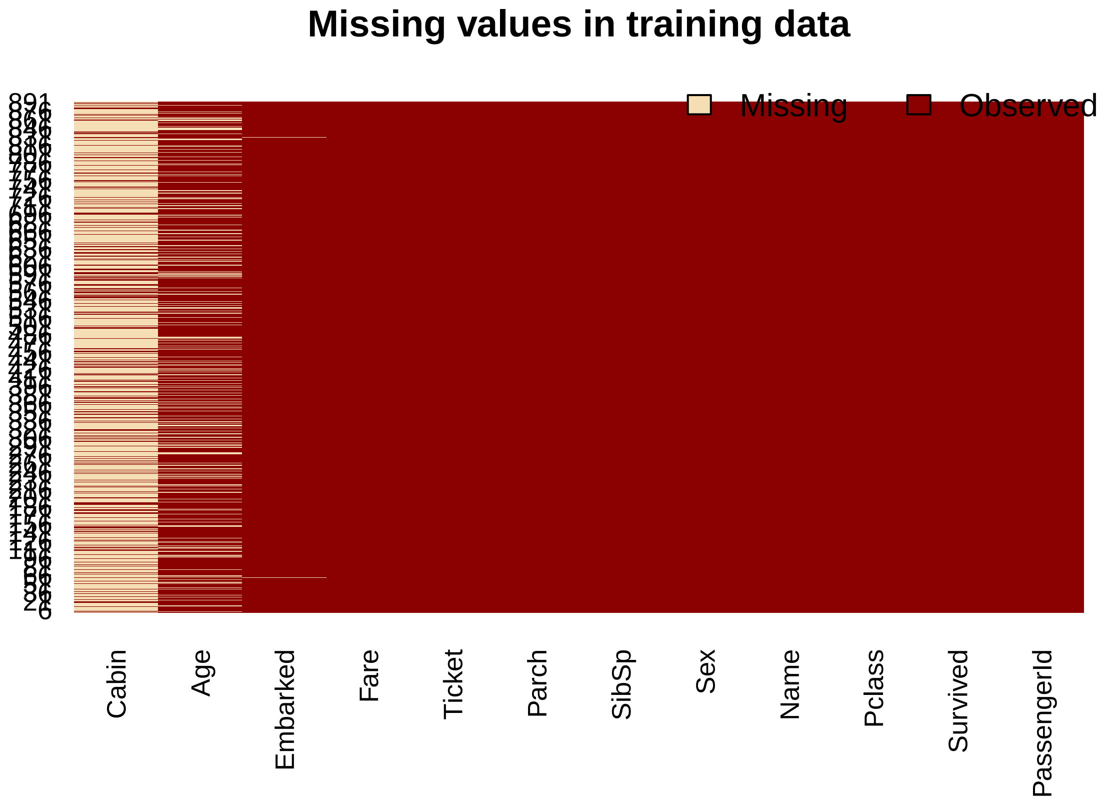
Sibling: Brother, Sister, Stepbrother, or Stepsister of Passenger Aboard Titanic
Spouse: Husband or Wife of Passenger Aboard Titanic (Mistresses and Fiances Ignored)
Parent: Mother or Father of Passenger Aboard Titanic
Child: Son, Daughter, Stepson, or Stepdaughter of Passenger Aboard Titanic

Other family relatives excluded from this study include cousins, nephews/nieces, aunts/uncles, and in-laws. Some children travelled only with a nanny, therefore parch=0 for them. As well, some travelled with very close friends or neighbors in a village, however, the definitions do not support such relations.

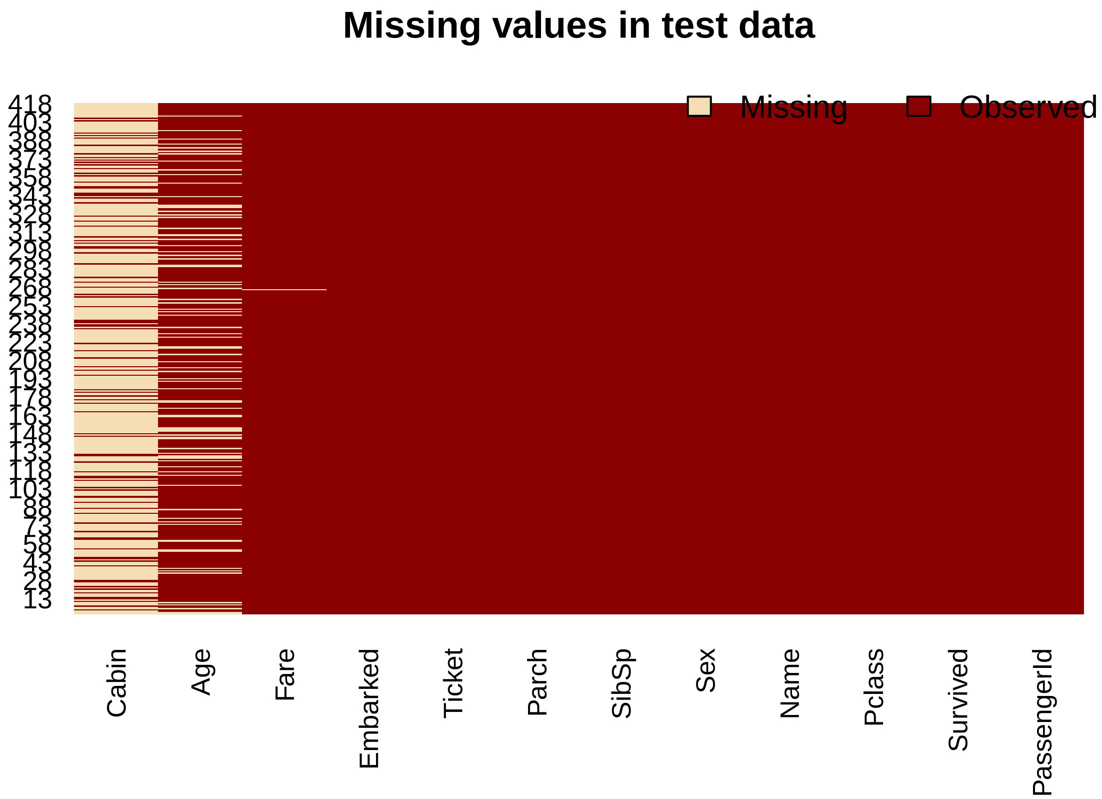
Missing Values

The next step will be to look for missing values.

```
#replace blanks by NA
training[training==""]<-NA
test[test==""]<-NA
library(Amelia)
missmap(training,main="Missing values in training data")
```



```
missmap(test,main="Missing values in test data")
```



```r
#combine training and test data sets for preprocessing
alldata <- rbind(training,test)
alldata$Pclass<-as.factor(alldata$Pclass)
```

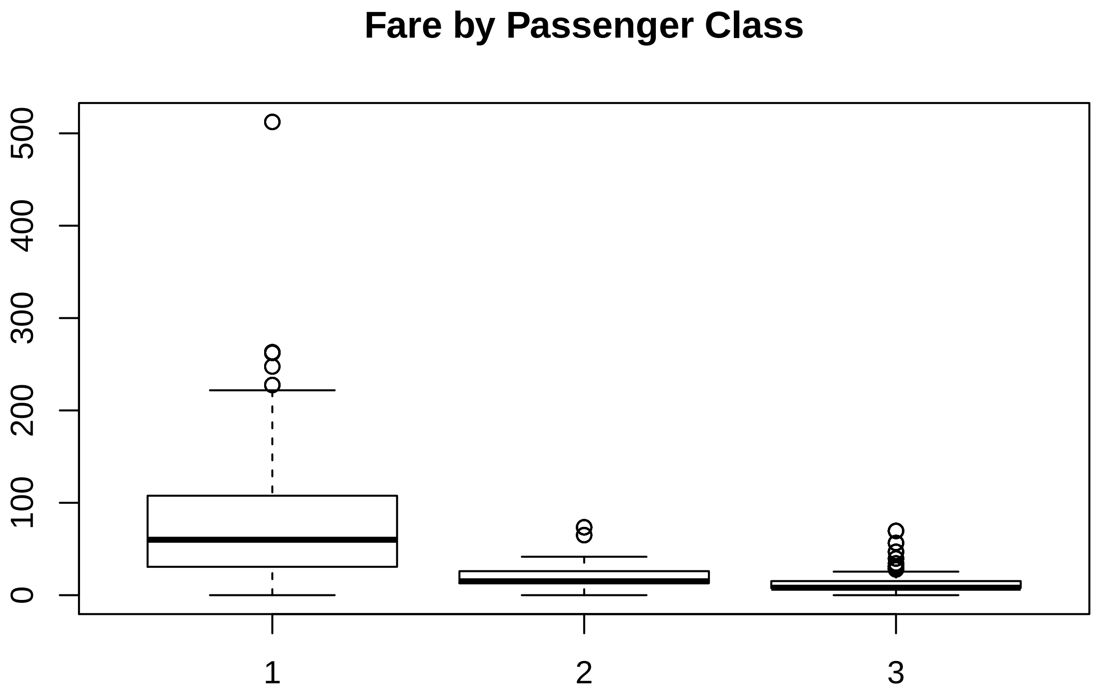
So in both training and test sets, many of the Cabin values and Age values are missing, and a couple of values are missing for Embarked and Fare.
I'll take a look first at the fares. Which passenger has no reported fare?

Fares and Tickets

```r
#which passenger has no reported fare?
alldata[which(is.na(alldata$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
## 1044   3701   NA  <NA>        S
```

```r
boxplot(Fare~Pclass,data = alldata,main="Fare by Passenger Class")
```



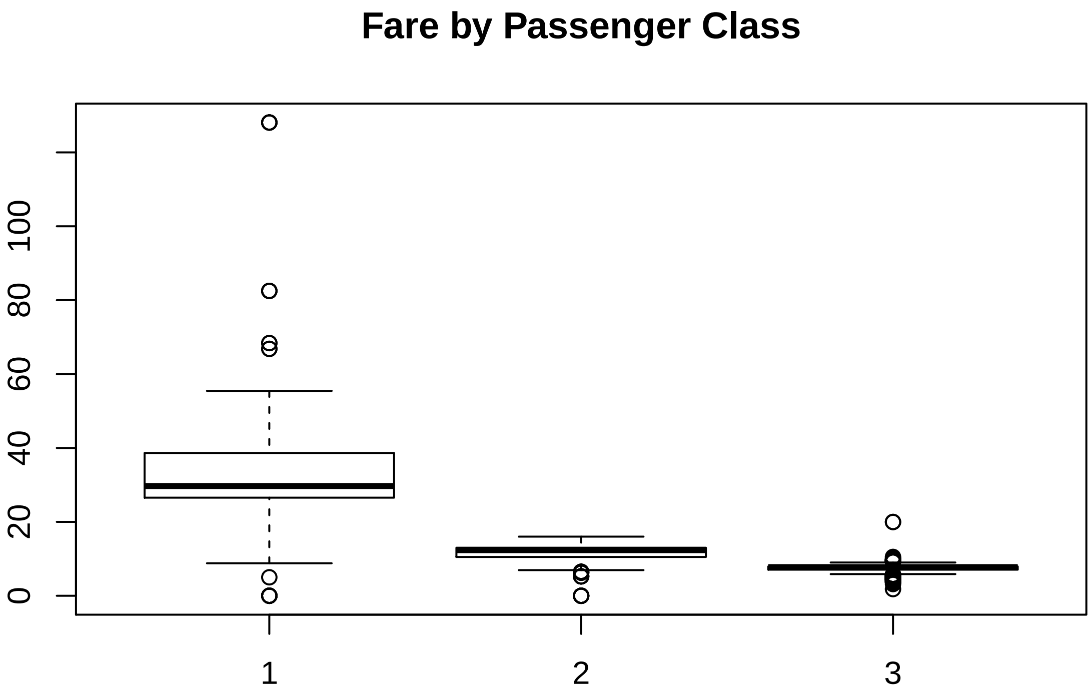
The fares seem to have very large variability even within passenger classes. Perhaps the fares are family or group fares and not the fare per passenger. In order to impute the missing fare value it seems appropriate to calculate the fare per passenger - FarePP.
I did this by looking at the ticket data. I removed the prefixes from those tickets which had prefixes and treated the remaining ticket numbers as numeric values. The reason for this was that I hypothesized that consecutive ticket numbers would probably have been purchased consecutively, some of them by family of friends of the previous ticket holders, and this information might have influence in the final model. For the mean time what I needed to know was how many passengers were travelling on each ticket, and then the fare per passenger would be the Fare value divided by this number.

```
tck<-sapply(as.character(alldata$Ticket),function(s) strsplit(s," "))
tick<-as.numeric(sapply(tck,function(s){
        if(length(s)==1) s
        else if (length(s)==2) s[2]
        else s[3]
        }))
alldata$Ticket <-tick

###it appears that fares for multiple tickets are a total fare for the group
###need to divide fare by number of times ticket number appears
tc<-table(alldata$Ticket)
counts<- data.frame(TickNum=names(tc),counts<-as.numeric(tc))
for (i in 1:nrow(alldata)){
        if (!is.na(alldata$Fare[i])&!is.na(alldata$Ticket[i]))
        alldata$FarePP[i]<-alldata$Fare[i]/counts[which(counts[,1]==alldata$Ticket[i]),2]
        }
```

I have seen attempts by other people to calculate the fare per passenger by deviding the total fare by the family size (which can be calculated from Parch and SibSp), but I preferred this approach after I found a group ticket whose passengers were not members of the same family.
Now I can show the distribution of fares per passenger:

```
boxplot(FarePP~Pclass,data = alldata,main="Fare by Passenger Class")
```



Much of the variability in 2nd and 3rd class fares has been reduced, and there is a considerable reduction also in the 1st class variability of fares.
Passenger number 1044 is the only passenger travelling on his ticket number. Since he travelled 3rd class from Southhampton, it seems reasonable to impute his missing fare with the median fare per passenger in this class.

```
fares<-alldata$FarePP[alldata$Pclass==3 & alldata$Embarked =="S"]
alldata$Fare[1044]<-median(fares,na.rm=T)
alldata$FarePP[1044]<-median(fares,na.rm=T)
alldata[which(alldata$Ticket==3701),]
```

```
##      PassengerId Survived Pclass               Name  Sex  Age SibSp Parch
## 1044        1044       NA      3 Storey, Mr. Thomas male 60.5     0     0
##      Ticket   Fare Cabin Embarked FarePP
## 1044   3701 7.7958  <NA>        S 7.7958
```

Port of Embarkment and Cabins

Let's see which passengers have unknown embarkment data:

```
alldata[is.na(alldata$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 FarePP
## 62  female  38     0     0 113572   80   B28     <NA>     40
## 830 female  62     0     0 113572   80   B28     <NA>     40
```

These two passengers travelled on the same ticket and shared a cabin, so they must all have embarked at the same port. They are both first class passengers. Their fares are relatively high. I'll check if any of the ports can be ruled out because of the fare they payed:

```
summary(alldata$FarePP[alldata$Pclass==1 & alldata$Embarked=="S"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    0.00   26.00   26.55   30.04   34.02   66.82       2 
```

```
summary(alldata$FarePP[alldata$Pclass==1 & alldata$Embarked=="Q"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##      30      30      30      30      30      30       2 
```

```
summary(alldata$FarePP[alldata$Pclass==1 & alldata$Embarked=="C"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   15.50   27.72   34.65   38.32   39.60  128.10       2 
```

So it's very unlikely that with a fare=40, that the port of embarkment was Q. The cabin data is quite sparse, but maybe there will be a clue there about the distribution of ports of embarkment by cabin data. I'll separate out the letter prefix of the cabin variable, which represents the deck of the cabin, and have a look at which port passengers on deck B embarked:

```
alldata$Deck<- as.factor(sapply(alldata$Cabin,function(s) substr(s,1,1)))
table(alldata$Embarked[alldata$Deck=="B"])
```

```
## 
##     C  Q  S 
##  0 32  0 31 
```

Still no wiser!
Perhaps finer details will help. The passengers whose port of embarkment is unknown were in cabin B28. Let's have a look where all their neighbours embarked. I'll look at all the passengers in cabins B20-B28, and those in cabins B30-B38.

```
alldata$CabNum<-sapply(alldata$Cabin,function(s) substr(s,1,2))
alldata$Embarked[which(alldata$CabNum=="B2" )]
```

```
## [1] <NA> S    S    S    S    <NA> S    S   
## Levels:  C Q S
```

```
alldata$Embarked[which(alldata$CabNum=="B3" )]
```

```
## [1] C C C S C C S C
## Levels:  C Q S
```

It seems that all other passengers in B deck cabins with numbers in the twenties embarked at port S, whereas most of the passengers in cabins with numbers in the thirties embarked at port C. I'm going to impute the missing ports of embarkment as "S".

```
alldata$Embarked[c(62,830)]<-"S"
```

Age and Name

There are many passengers whose age is not reported. The name variable consists of Title, Surname and First names. The title gives some clue to the age. "Master" refers to a young male who is not old enough to be called "Mr" and "Miss" is an unmarried female. It's easy enough to extract the title information out of the name and store it as a variable which can be used as a predictor for age. Some of the titles are equivalent to "Mr" or "Mrs" in other languages, so I replaced them by the English equivalent. Other titles expressing nobility were combined into special male and female categories - not so much because they might be older, but because later when I will use this as a feature for predicting survival outcomes, such passengers might have had precedence for getting in a lifeboat. At the same time as creating the "Title" variable , I created another variable storing the surnames - which will be used in the final model to identify families.

```
###first make Name into character vector instead of factor
alldata$Name<-as.character(alldata$Name)
namesplit<-sapply(alldata$Name,function(x)unlist(strsplit(x,", ")))
alldata$Surname<-as.factor(namesplit[1,])


alldata$Title<-sapply(namesplit[2,],function(n)unlist(strsplit(n,".",fixed=TRUE))[1])
###combine "Capt","Col","Jonkheer","Major","Rev","Sir","Dr" into SpecialMr
t<-c("Capt","Col","Jonkheer","Major","Rev","Sir","Dr")
for (i in 1:7) alldata$Title<-gsub(t[i],"SpecialMr",alldata$Title)
###combine "Lady", "the Countess" into "SpecialMrs"
t<- c("Lady", "the Countess")
for (i in 1:2) alldata$Title<-gsub(t[i],"SpecialMrs",alldata$Title)
###replace "Don" by "Mr", "Mlle" by "Miss", "Mme" by "Mrs"
alldata$Title<-gsub("Dona","Mrs",alldata$Title)
alldata$Title<-gsub("Don","Mr",alldata$Title)
alldata$Title<-gsub("Mme","Mrs",alldata$Title)
alldata$Title<-gsub("Mlle","Miss",alldata$Title)
###make factor
alldata$Title <- as.factor(alldata$Title)
table(alldata$Title)
```

```
##
##     Master       Miss         Mr        Mrs         Ms  SpecialMr
##         61        262        758        199          2         25
## SpecialMrs
##          2
```

I decided to try and predict the missing ages by training a model on all the passengers whose ages are known, using Title, Sex, SibSp and Parch as predictor variables. After trying several regression algorithms I found "gam" in the Caret package to give results that looked satisfactory.

```
#Use gam to predict missing values, based on Title, Sex, SibSp, Parch
fit <- train(Age~SibSp+Parch+Title+Sex,data=alldata[!is.na(alldata$Age),],method="gam")
fit$finalModel
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ TitleMiss + TitleMr + TitleMrs + Sexmale + SibSp +
##     Parch
## Total model degrees of freedom 7
##
## GCV score: 167.4158
```

```
newAge <- predict(fit,alldata[is.na(alldata$Age),])
alldata[is.na(alldata$Age),6]<-newAge
```

After predicting the missing ages with the model constructed by the algorithm I decided to transform the Age variable into a factor variable with levels 0 representing an adult, and 1 representing a child. I did this because there were so many missing ages that had to be predicted that it is not really meaningful to rely on the accuracy of the prediction. This, together with the well known policy adopted on the Titanic of saving women and children first, seemed to be a more meaningful choice of feature. I thought there might be a different age for with a person might be considered a child or an adult according to their class. Lower class children might be considered adults at a younger age than upper class children who would be more protected. I used kfold cv to find cutoff age for child/adult for 1st-2nd class and 3rd class separately.

```
 set.seed(1968)
 flds<-createFolds(alldata$PassengerId[1:891], k = 3, list = TRUE, returnTrain = FALSE)
 a<-c(13,14,15,16,17,18)

 for (i in 1:3){
         for(j in seq_along(a)){
                 for (k in seq_along(a)){
                         cvtrain<-alldata[-flds[[i]],]
                         cvtest <- alldata[flds[[i]],]
                         #make age a factor variable young (<-a[j])or not young
                         cvtrain$Age[cvtrain$Pclass != "3" & cvtrain$Age<=a[j]]<-1
                         cvtrain$Age[cvtrain$Pclass=="3"& cvtrain$Age<=a[k]]<-1
                         cvtrain$Age[cvtrain$Pclass!="3" & cvtrain$Age >a[j] ]<-0
                         cvtrain$Age[cvtrain$Pclass=="3" & cvtrain$Age >a[k] ]<-0
                         cvtrain$Age <-as.factor(cvtrain$Age)
                         cvtest$Age[cvtest$Pclass != "3" & cvtest$Age<=a[j]]<-1
                         cvtest$Age[cvtest$Pclass=="3"& cvtest$Age<=a[k]]<-1
                         cvtest$Age[cvtest$Pclass!="3" & cvtest$Age >a[j] ]<-0
                         cvtest$Age[cvtest$Pclass=="3" & cvtest$Age >a[k] ]<-0
                         cvtest$Age <-as.factor(cvtest$Age)
                         data<-cvtrain[,c(2,3,5,6,7,8,10,12,13,14,17,18)]
                         set.seed(115)
                         cvfit<- train(data=data,Survived~.,method="rf")
                         surv <- predict(cvfit,newdata=cvtest)
                         print(c(i,a[j],a[k], sum(surv==cvtest[,2])/length(surv)))
                 }
         }
 }

##the best result for j=16,k=13, ie class 1-2, child is <=16, class 3 , child is <=13
```

```
#make age a factor variable child or not child

alldata$Age[alldata$Pclass != "3" & alldata$Age<=16]<-1
alldata$Age[alldata$Pclass=="3"& alldata$Age<=13]<-1
alldata$Age[alldata$Pclass!="3" & alldata$Age >16 ]<-0
alldata$Age[alldata$Pclass=="3" & alldata$Age >13 ]<-0
alldata$Age <-as.factor(alldata$Age)
table(alldata$Age)
```

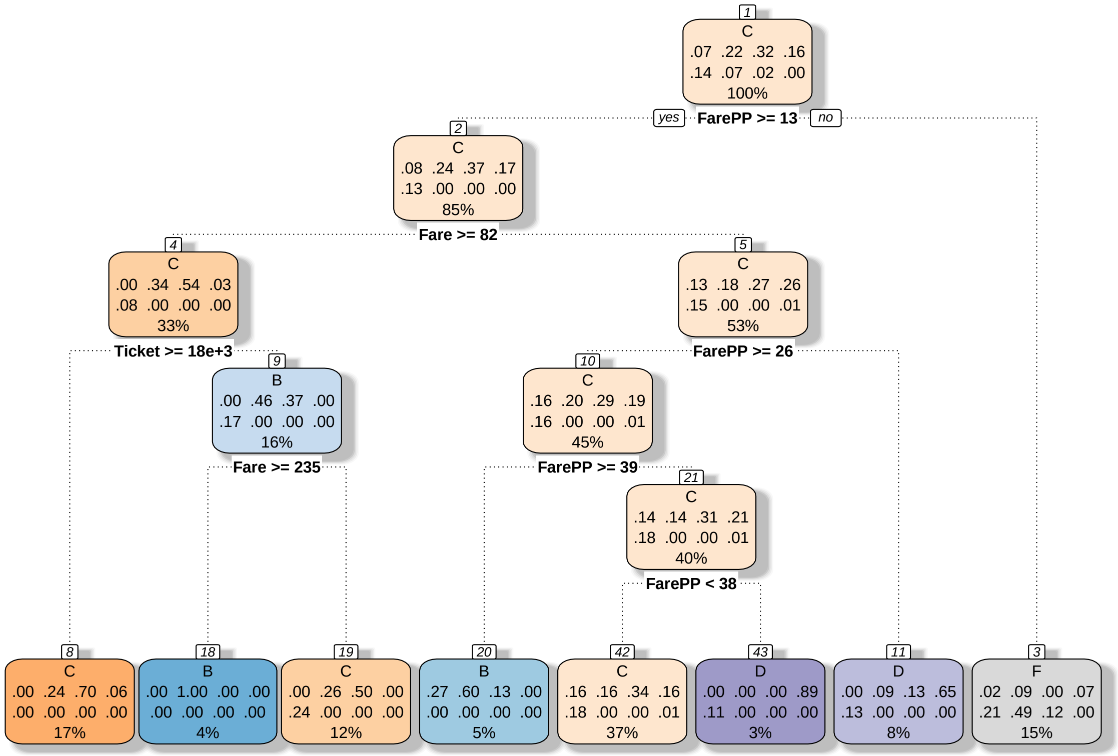
```
##
##    0    1
## 1192  117
```

I could have tried to classify the passengers directly as child or adult, instead of imputing the ages and then classifying, but I found this method to give more accurate predictions of survival.

Imputing missing Cabin data

I used a decision tree (rpart in caret package) to predict missing cabin decks. As predictors I used Pclass,FarePP,Embarked as features that surely influence the deck allocated to the passenger, and Fare,Surname,Ticket to allow family/group identifcation to assist predicting a deck for passengers whose deck is unknown using information known for other members of the same group.

```
library(rpart)
library(rpart.plot)
library(rattle)
set.seed(192)
cabfit <- train(Deck~Pclass+Fare+Surname+Ticket+FarePP+Embarked,data=alldata[!is.na(alldata$Deck),],met
fancyRpartPlot(cabfit$finalModel,main="", sub="Decision tree for predicting missing cabin decks")
```



Decision tree for predicting missing cabin decks

```
alldata$Deck[is.na(alldata$Deck)]<-predict(cabfit,newdata=alldata[is.na(alldata$Deck),],type="raw")
table(alldata$Deck)
```

```
##
##    A   B   C   D   E   F   G   T
##   22  73 141  84  41 942   5   1
```

Next, I wondered what to do about the cabin numbers. I don't think it is meaningful to try and predict the exact cabin numbers. I extracted rather the position of the cabin on the deck, that is whether the cabin number is in the twenties or thirties etc., for earlier preprocessing, but I don't think there is much point trying to impute the missing ones, so I'll leave this variable out.

Family Size

Another feature that many people created is the family size. This tries to take into account the influence of family members worrying about each other on survival. Other people created a feature of mother/child, but I chose to incorporate family size into the model. FamilySize is number of parents/children + number of spouse/siblings + oneself.

```
###FamilySize is number of parents/children + number of spouse/siblings + oneself
alldata$FamilySize <- 1+alldata$Parch + alldata$SibSp
```

Final Model

Now it is time to build the model which will try to learn from the features of the training set how to predict survival for passengers. To facilitate the model I resplit the data set into training and test sets including only the variables which will be included in the model. These variables will be : Pclass, Sex, Age, SibSp, Parch, Ticket, Fare, Embarked, FarePP, Deck, Surname, Title, Family size. I used random forest within the caret package.

```
set.seed(375)
newtrain<- alldata[1:891,c(2,3,5,6,7,8,9,10,12,13,14,16,17,18)]
newtrain$Survived<-as.factor(newtrain$Survived)
newtest<- alldata[892:1309,c(2,3,5,6,7,8,9,10,12,13,14,16,17,18)]
fit1 <- train(Survived~.,data=newtrain,method="rf", do.trace=100)
```

```
##   mtry  Accuracy     Kappa AccuracySD    KappaSD
## 1    2 0.6214840 0.0000000 0.02025377 0.00000000
## 2   42 0.8139995 0.5904302 0.01290315 0.03192431
## 3  900 0.8217636 0.6100183 0.01625610 0.03849189
```

```
fit1$results


Survived<-predict(fit1,newdata=newtest)
PassengerId <- test$PassengerId
solution <- data.frame(PassengerId,Survived)
write.csv(solution,file="submit.csv",row.names=FALSE)
```

The accuracy reported by the model is approximately 82%, and the accuracy obtained on the Kaggle site public leaderboard after submitting the prediction made by this model on the test set was about 80%.