Machine Learning with Titanic

Leon Shpaner

2020-07-06

This project is based on a dataset of passengers from the RMS Titanic, a famous luxury cruise liner that sank in 1912. (For more details on the Titanic, see https://en.wikipedia.org/wiki/RMS_Titanic.) The dataset comes from Kaggle (https://www.kaggle.com/c/titanic). Kaggle hosts data science competitions and is a great resource for practicing predictive analytics skills. Provided herein are test and train datasets for the titanic scenario.

Note. Models are built with train and tested with test.

Working with the train.csv dataset the following steps were followed:

- Imported the titanic.csv into a data frame in R
- Generated a series of descriptive statistics
- Determined if there were any variables with missing observations
- Generated a series of visualizations to better understand the sample

The ultimate goal of the project was to build models to determine which passengers were most likely to have survived the sinking of the Titanic. In this first part of the project, we will focus on just describing the data by providing some insight into who lived and died when the Titanic sank (variable Survived in the sample). Variables are supporting insights with descriptive statistics and visualizations generated in R.

1. Install the readr package, load the library, and load titanic.csv into the data frame named boat

```
#install.packages("readr")
library(readr)
boat <- read.csv("train.csv")
```

2. Return a vector (or matrix) in the same dimension as data using the boat data frame and class function. Use the summary function to quickly summarize the sample

```
sapply(boat,class)
## PassengerId    Survived      Pclass        Name         Sex         Age
##   "integer"   "integer"   "integer" "character" "character"   "numeric"
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##   "integer"   "integer" "character"   "numeric" "character" "character"

summary(boat)
##   PassengerId       Survived          Pclass          Name
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median :446.0   Median :0.0000   Median :3.000   Mode  :character
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##
##      Sex                 Age            SibSp           Parch
##  Length:891          Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  Class :character   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000
##  Mode  :character   Median :28.00   Median :0.000   Median :0.0000
##                      Mean   :29.70   Mean   :0.523   Mean   :0.3816
##                      3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##                      Max.   :80.00   Max.   :8.000   Max.   :6.0000
##                      NA's   :177
##     Ticket               Fare            Cabin             Embarked
##  Length:891          Min.   :  0.00   Length:891         Length:891
##  Class :character   1st Qu.:  7.91   Class :character   Class :character
##  Mode  :character   Median : 14.45   Mode  :character   Mode  :character
##                      Mean   : 32.20
##                      3rd Qu.: 31.00
##                      Max.   :512.33
##
```

3. Using relevant descriptive statistics, we can look at:

a) the average fare that the passengers paid:

```
mean(boat$Fare)
## [1] 32.20421
```

b) the average age of passengers on the Titanic while removing all missing (NA) values:

```
mean(as.numeric(boat$Age),na.rm=TRUE)
## [1] 29.69912
```

c) similarly, we can get the standard deviation as follows:

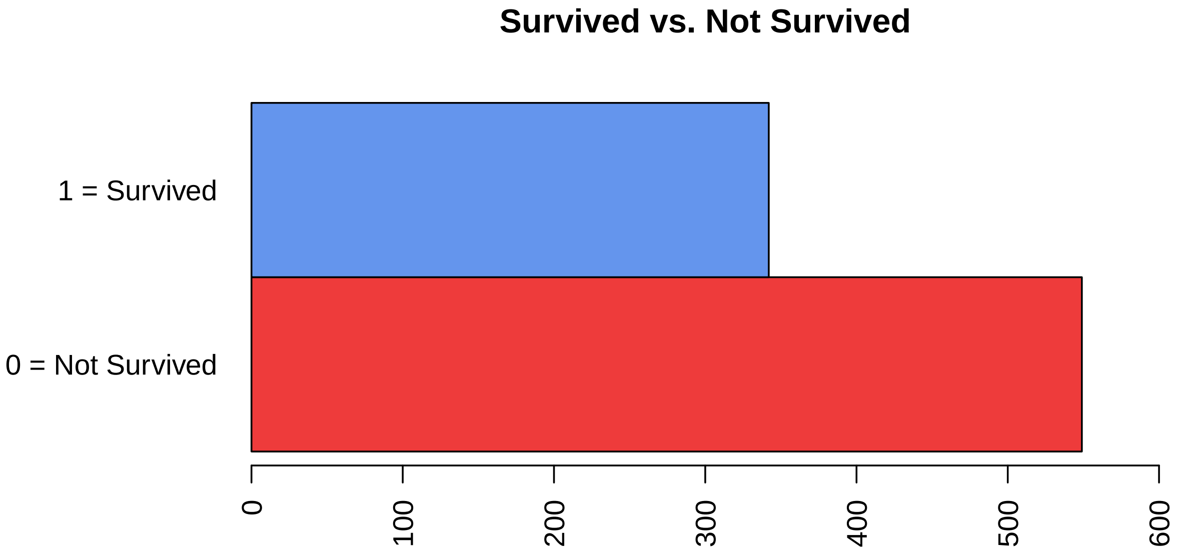
```
sd(as.numeric(boat$Age),na.rm=TRUE)
## [1] 14.5265
```

d) and the average (as a percentage) of those who survived:

```
mean(boat$Survived)
## [1] 0.3838384
```

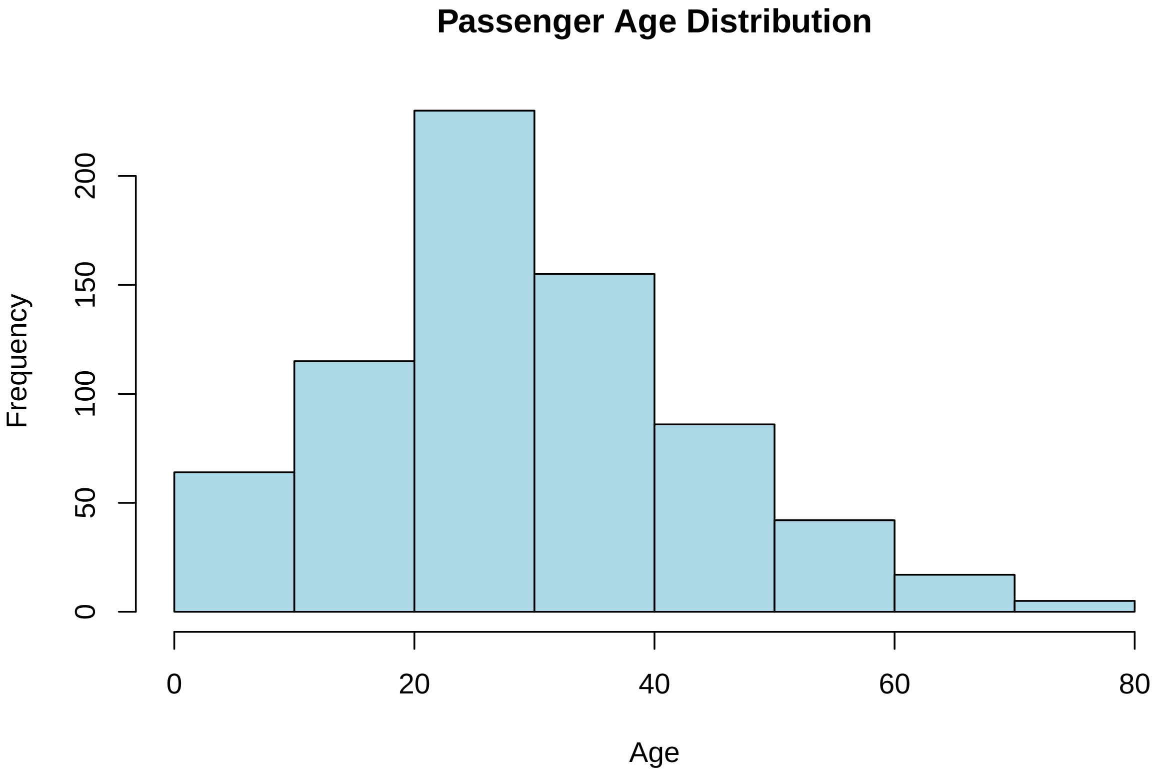
4. To get a graphical comparison of who survived vs. who did not, we can see this in the following bar chart:

```
counts <- table(boat$Survived)
par(las=2) # make label text perpendicular to axis
par(mar=c(5,8,4,2)) # increase y-axis margin.
barplot(counts, main="Survived vs. Not Survived", horiz = TRUE,
        names.arg=c("0 = Not Survived", "1 = Survived"),
        col=c("brown2",
        "cornflowerblue"),
        xlim=c(0,600),space=c(0,0))
```



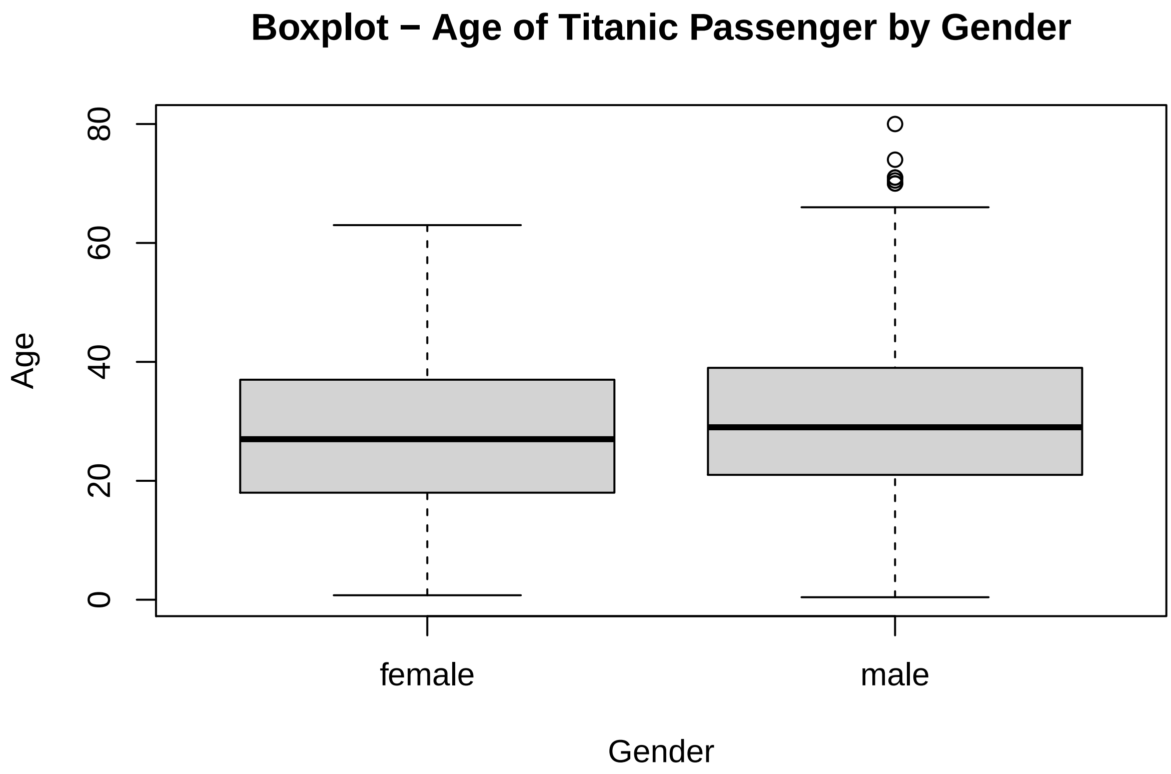
5. Now we will generate a histogram to show the age distribution of passengers on the titanic.

```r
h=hist(boat$Age,xlab="Age", ylab="Frequency",
       main="Passenger Age Distribution", col="lightblue")
```



6. To further examine the data, we can also look at the age of Titanic Passengers by Gender:

```
boxplot(boat$Age~boat$Sex,xlab="Gender", ylab="Age",
        main="Boxplot - Age of Titanic Passenger by Gender")
```



Based on this basic model, we can see that 38% of the passengers survived based on the average we calculated in #3. Now, our goal is to dive deeper and select variables that we think will influence whether passengers survived, and then use k-nearest neighbors (KNN) to build classification models that will predict who survived the Titanic.

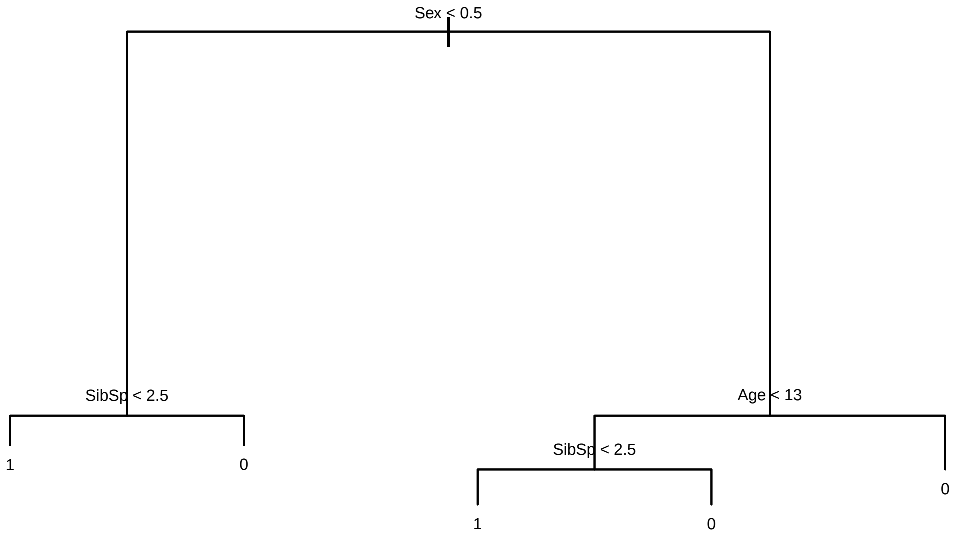
To accomplish this, we will load two additional libraries: class, and caTools. According to the documentation, class is a package that contains "various functions for classification, including k-nearest neighbour, Learning Vector Quantization and Self-Organizing Maps" (https://www.rdocumentation.org/packages/class/versions/7.3-17). CaTools "contains several basic utility functions including: moving (rolling, running) window statistic functions, read/write for GIF and ENVI binary files, fast calculation of AUC, LogitBoost classifier, base64 encoder/decoder, round-off-error-free sum and cumsum, etc." (https://www.rdocumentation.org/packages/caTools/versions/1.17.1)

```
library(class)
library(caTools)
```

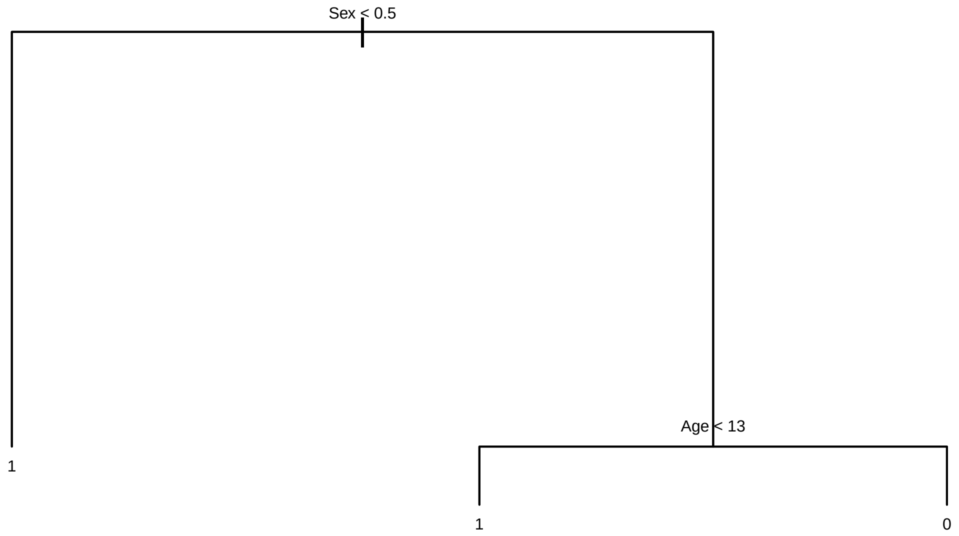
In order to classify these variables, we will use KNN as a statistical estimation/ pattern recognition tool. In a nutshell, the algorithm will classify new variables based on existing variables' current classification.

Next, we want to see if there exists a relationship between the selected variables, but not all of the variables are quantitative, and as such, we must run a logistic regression, as follows.

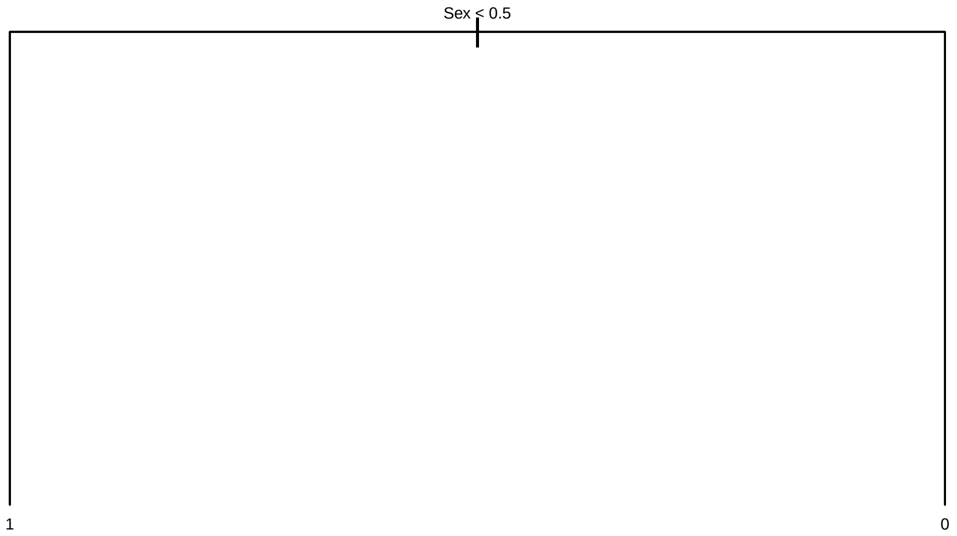
```r
library(tree)
boat$AgeAVG<-boat$Age
boat$Survived<-as.factor(boat$Survived)
boatLR<-glm(Survived~Sex+AgeAVG+SibSp+Parch,family=binomial(),data=boat)
#remove any variables from boat that you won't use for your classification
#the following code uses Survived, Sex, Age, SibSp, and Parch
# You can use your choice of variables, or fewer variables if you wish
boat<-boat[,c(2,5,6,7,8)]
sum(is.na(boat$Age))
## [1] 177
boat<-within(boat,Age[is.na(Age)]<-mean(Age,na.rm=TRUE))
boat$Sex[boat$Sex=="male"]<-1
boat$Sex[boat$Sex=="female"]<-0
set.seed(123)

sample<-sample.split(boat$Sex, SplitRatio = .80)
train<-subset(boat, sample == TRUE)
test<-subset(boat, sample == FALSE)
knn1<-knn(train[-1],test[-1],train$Sex, k=1)

train<-subset(boat, sample == TRUE)
test<-subset(boat, sample == FALSE)
knn1<-knn(train[-1],test[-1],train$Age, k=1)

train<-subset(boat, sample == TRUE)
test<-subset(boat, sample == FALSE)
knn1<-knn(train[-1],test[-1],train$Survived, k=1)

train<-subset(boat, sample == TRUE)
test<-subset(boat, sample == FALSE)
knn1<-knn(train[-1],test[-1],train$SibSp, k=1)

train<-subset(boat, sample == TRUE)
test<-subset(boat, sample == FALSE)
knn1<-knn(train[-1],test[-1],train$Parch, k=1)

CF<-table(knn1,test$Age)

Precision<-CF[2,2]/(CF[2,1]+CF[2,2])
Precision
## [1] 0.5

train<-subset(boat, sample == TRUE)
test<-subset(boat, sample == FALSE)
TrainTree<-tree(Survived ~ Sex+Age+SibSp, data=train)

plot(TrainTree)
text(TrainTree, cex=.5)
```



```
summary(TrainTree)
##
## Classification tree:
## tree(formula = Survived ~ Sex + Age + SibSp, data = train)
## Number of terminal nodes:  5
## Residual mean deviance:  0.9354 = 662.3 / 708
## Misclassification error rate: 0.1781 = 127 / 713
TrainPrune<-prune.tree(TrainTree,best = 3,newdata=test,method = "misclass")
plot(TrainPrune)
text(TrainPrune, cex=.5)
```



```
summary(TrainPrune)
##
## Classification tree:
## snip.tree(tree = TrainTree, nodes = c(2L, 6L))
## Variables actually used in tree construction:
## [1] "Sex" "Age"
## Number of terminal nodes:  3
## Residual mean deviance:  0.9857 = 699.9 / 710
## Misclassification error rate: 0.1978 = 141 / 713
TrainPrune<-prune.tree(TrainTree,best = 2,newdata=test,method = "misclass")
plot(TrainPrune)
text(TrainPrune, cex=.5)
```



```
summary(TrainPrune)
##
## Classification tree:
## snip.tree(tree = TrainTree, nodes = 2:3)
## Variables actually used in tree construction:
## [1] "Sex"
## Number of terminal nodes:  2
## Residual mean deviance:  1.028 = 731.2 / 711
## Misclassification error rate: 0.2104 = 150 / 713

PredSurv <- predict(TrainTree, test, type="class")

CF<-table(test$Survived,PredSurv)

Precision<-CF[2,2]/(CF[2,1]+CF[2,2])
Precision
## [1] 0.6984127
```