Titanic Analysis

Taras the Analyst

2023-04-09

Introduction

This is the report produced from the Kaggle notebook 'Titanic Analysis' by Taras K. from 03/18/2023.

The original inspirational source is by Hilla Behar

In this analysis the following questions were asked:

1. What is the relationship the features and a passenger's chance of survival.
2. Prediction of survival for the entire ship.

Last update: 09/04/2023 (see the list of updates at the end of this work)

Setting the environment

Packages

```
# The following packages are to be used for the current analysis
library(dplyr)          # for data manipulation
library(tidyverse)      # for working operations
library(ggplot2)        # for data visualization
library(GGally)         # Extension to 'ggplot2'
library(rpart)          # decision tree model package
library(rpart.plot)     # decision tree visualization package
library(ggcorrplot)     # to understand the correlation matrix
library(randomForest)   # planting the trees needs some methodology...:)
library(pander)         # to create pretty tables
library(knitr)          # to create pretty tables
library(tinytex)        # to use the features for file rendering to .pdf
```

Loading the data sources

```
test <- read.csv('./test.csv',stringsAsFactors = FALSE)
train <- read.csv('./train.csv', stringsAsFactors = FALSE)
dim(test)   # check the test data frame dimensions
```

```
## [1] 418  11
```

```
dim(train)  # check the train data frame dimensions
```

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

Data elaboration

Merging both datasets into a consolidated one*

bind_rows() is to be used, as **rbind()** doesn't work here due to different number of columns in *train* and *test*

```
full <- bind_rows(train,test)
dim(full)  # check the resulted data frame dimensions
```

```
## [1] 1309   12
```

```
str(full)  # check the resulted data frame structure
```

```
## 'data.frame':    1309 obs. of  12 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)" "Heik
##  $ 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" ...
```

The data is to be checked for missing values

```
## [1] "Here is missing value check:"
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0         418           0           0           0         263
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           1           0           0
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0          NA           0           0           0          NA
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0          NA        1014           2
```

So, the ouput is: N/As - left table, NULLs - right table

```
knitr::kable(list(k1, k2))
```

```
# cross-checking the empty records for Embarked
filter(full, full$Embarked == "")
```

```
##   PassengerId Survived Pclass                                        Name    Sex
## 1          62        1      1                         Icard, Miss. Amelie female
## 2         830        1      1 Stone, Mrs. George Nelson (Martha Evelyn) female
##   Age SibSp Parch Ticket Fare Cabin Embarked
## 1  38     0     0 113572   80   B28
## 2  62     0     0 113572   80   B28
```

	x
PassengerId	0
Survived	418
Pclass	0
Name	0
Sex	0
Age	263
SibSp	0
Parch	0
Ticket	0
Fare	1
Cabin	0
Embarked	0

	x
PassengerId	0
Survived	NA
Pclass	0
Name	0
Sex	0
Age	NA
SibSp	0
Parch	0
Ticket	0
Fare	NA
Cabin	1014
Embarked	2

```
# getting it into a bit more visually attractive way
kable(filter(full, full$Embarked == ""))
```

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
62	1	1	Icard, Miss. Amelie	female	38	0	0	113572	80	B28	
830	1	1	Stone, Mrs. George Nelson (Martha Evelyn)	female	62	0	0	113572	80	B28	

```
# getting the digits of missing values
paste("= N/A in full dataset:")  # that's added for some internal explanations
```

```
## [1] "= N/A in full dataset:"
```

```
pander(table(is.na(full)))  # showing aggregated "n/a" values within each column
```

FALSE	TRUE
15026	682

Cleaning & transforming the data

```
full$Embarked[full$Embarked == ""] = "C"
```

Change the empty strings in Embarked to the first choice "C"

```
apply(full, 2, function(x) length(unique(x)))
```

See how many features can be transformed to factors

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##        1309           3           3        1307           2          99
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           7           8         929         282         187           3
```

Move the attributes Survived, Pclass, Sex, Embarked to be factors

```
cols <- as.factor(c("Survived", "Pclass", "Sex", "Embarked"))
for (i in cols){
  full[, i] <- as.factor(full[, i])
}
```

```
str(full)
```

Now let's look on the structure of the full data set

```
## 'data.frame':    1309 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ 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 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" "Heik
##  $ 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      : 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   : Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
```

Move the attributes Survived, Pclass, Sex, Embarked to be factors within train data set

```
cols <- as.factor(c("Survived", "Pclass", "Sex", "Embarked"))
for (i in cols){
  train[, i] <- as.factor(train[, i])
}
```

Now let's look on the structure of the train data set str(train)

Analyse the cleaned data

The data has been loaded & cleaned a little bit so far. Now, it's time to look at the relationships between the different attributes within set and to check the correlations within factored attributes, so to see if there's something useful.

```
full_fctrs <- full[, c("Survived", "Pclass", "Sex", "Embarked")]
train_fctrs <- train[, c("Survived", "Pclass", "Sex", "Embarked")]
```

```
dim(full_fctrs)  # check if the re-shaping went well resulted in 4 columns only
```

```
## [1] 1309    4
```

```
dim(train_fctrs)
```

```
## [1] 891   4
```

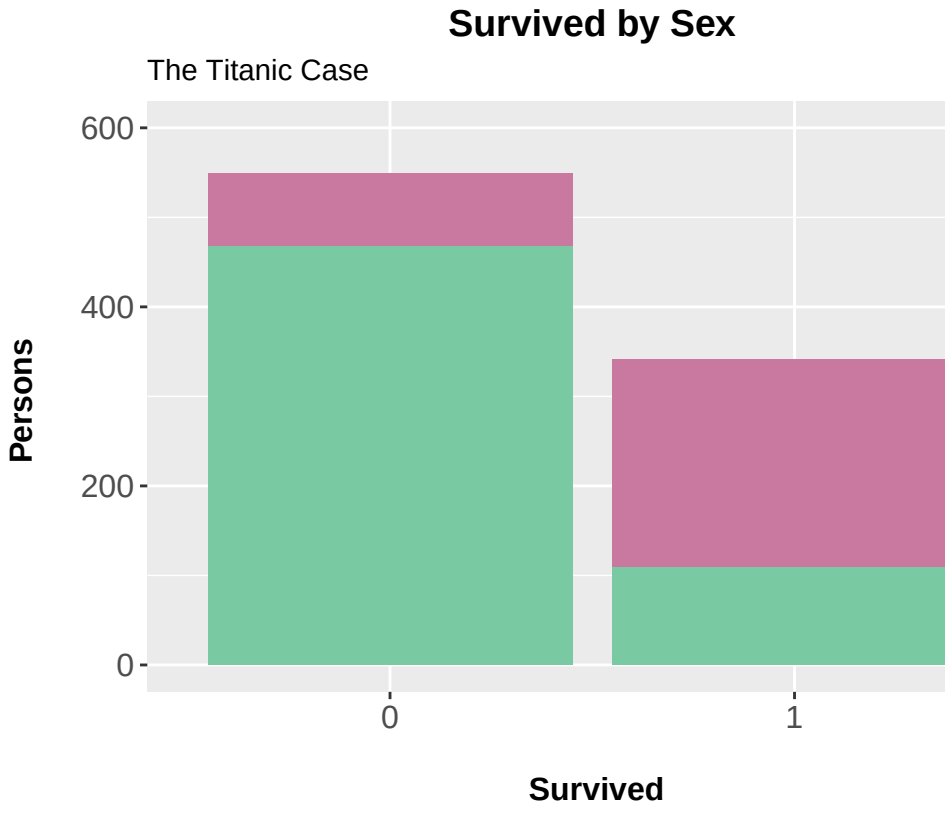
```
str(train_fctrs)  # getting the structure overview of train factors
```

```
## 'data.frame':    891 obs. of  4 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 ...
##  $ Embarked: Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

```
str(full_fctrs)   # getting the structure overview of test factors
```

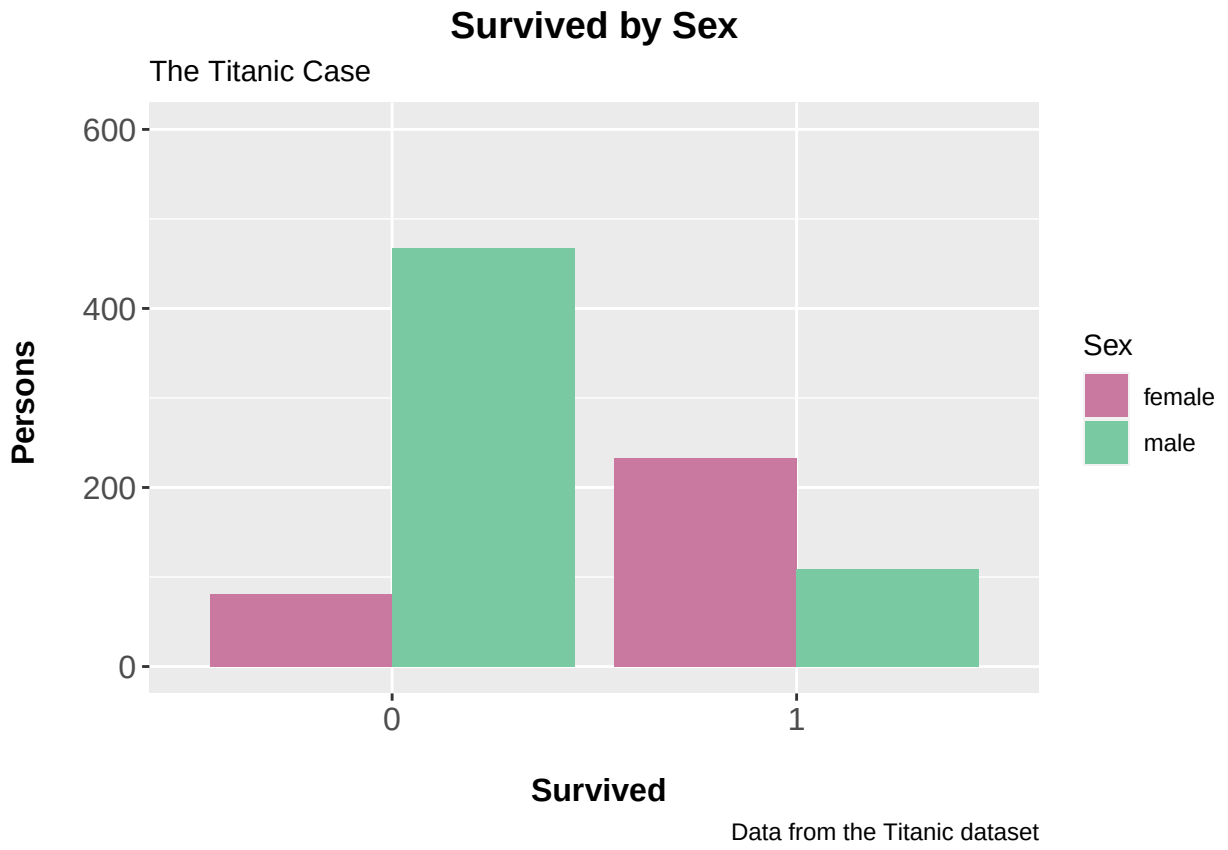
```
## 'data.frame':    1309 obs. of  4 variables:
##  $ Survived: Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass  : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Embarked: Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
```

```
# visualizing the numbers available so to get the general picture of the case
ggplot(data = train_fctrs, aes(x = Survived, fill = Sex)) + geom_bar() +
  scale_y_continuous(limits = c(0, 600)) +              # making visual limits
  scale_fill_manual(values = c("#c979a0", "#79C9A2")) + # color code for sex categories
  labs(title = "Survived by Sex",                       # setting labels
       subtitle = "The Titanic Case",
       caption = "Data from the Titanic dataset",
       x = "\n Survived", y = "Persons \n") +           # placing extra-line for better readability
  theme(axis.text = element_text(size = 12),
        axis.title = element_text(size = 12, face = "bold"),
        plot.title = element_text(size = 14, hjust = 0.5, face = "bold"))
```

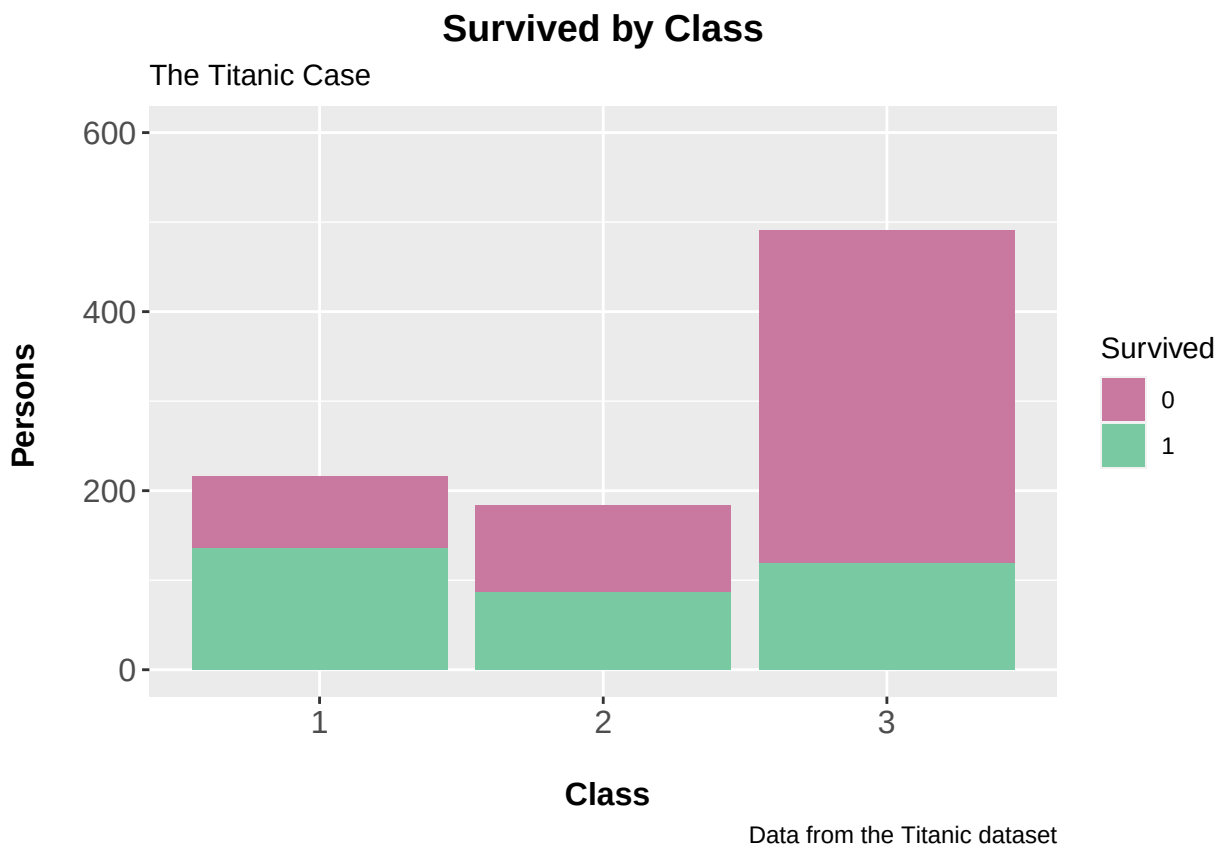


Adding some visuals to clarify the picture

```r
# side-by-side comparison to make things more understandable - Survived by Sex
ggplot(data = train_fctrs, aes(x = Survived, fill = Sex)) + geom_bar(position = "dodge") +
  scale_y_continuous(limits = c(0, 600)) +                # making visual limits
  scale_fill_manual(values = c("#c979a0", "#79C9A2")) +   # color code for sex categories
  labs(title = "Survived by Sex",                         # setting labels
       subtitle = "The Titanic Case",
       caption = "Data from the Titanic dataset",
       x = "\n Survived", y = "Persons \n") +
  theme(axis.text = element_text(size = 12),
        axis.title = element_text(size = 12, face = "bold"),
        plot.title = element_text(size = 14, hjust = 0.5, face = "bold"))
```




```r
# side-by-side comparison to make things more understandable - Survived by Sex
ggplot(data = train_fctrs, aes(x = Pclass, fill = Survived)) + geom_bar() +
  scale_y_continuous(limits = c(0, 600)) +                # making visual limits
  scale_fill_manual(values = c("#c979a0", "#79C9A2")) +   # color code for sex categories
  labs(title = "Survived by Class",
       subtitle = "The Titanic Case",
       caption = "Data from the Titanic dataset",
       x = "\n Class", y = "Persons \n")+
  theme(axis.text = element_text(size = 12),
        axis.title = element_text(size = 12, face = "bold"),
        plot.title = element_text(size = 14, hjust = 0.5, face = "bold"))
```




```
# side-by-side comparison to make things more understandable - Survived by Class
ggplot(data = train_fctrs, aes(x = Pclass, fill = Survived)) + geom_bar(position = "dodge") +
  scale_y_continuous(limits = c(0, 400)) +
  scale_fill_manual(values = c("#a79da1", "#79C9A2")) +   # color code for sex categories
  labs(title = "Survived by Class",
       subtitle = "The Titanic Case",
       caption = "Data from the Titanic dataset",
       x = "\n Class", y = "Persons \n")+
  theme(axis.text = element_text(size = 12),
        axis.title = element_text(size = 12, face = "bold"),
        plot.title = element_text(size = 14, hjust = 0.5, face = "bold"))
```