surviving predictive model titanic

andres tomatis

2022-07-16

Introduction

In this opportunity I am going to work with Titanic data set, which contains two data frames, both have features about a certain quantity of passengers, but the first one (train data) has a column named *survived* that tells us if the passenger survived or not. Second one (test data) doesn't. So, the task is to determine for every passenger in *test data* if survives or not. In order to do that, I have to use *train data* to create a predictive model.

install and library useful packages

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("caret", dependencies = TRUE)
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("randomForest")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──
```

```
## ✔ tibble  3.1.7     ✔ dplyr   1.0.9
## ✔ tidyr   1.2.0     ✔ stringr 1.4.0
## ✔ readr   2.1.2     ✔ forcats 0.5.1
## ✔ purrr   0.3.4
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::combine()       masks randomForest::combine()
## ✖ dplyr::filter()        masks stats::filter()
## ✖ dplyr::lag()           masks stats::lag()
## ✖ purrr::lift()          masks caret::lift()
## ✖ randomForest::margin() masks ggplot2::margin()
```

Upload the data set

upload data and take a look at it

```
train <- read.csv("train.csv")
test <- read.csv("test.csv")
```

```
head(train)
```

```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
## 4           4        1      1
## 5           5        0      3
## 6           6        0      3
##                                                 Name    Sex Age SibSp Parch
## 1                            Braund, Mr. Owen Harris   male  22     1     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1     0
## 3                             Heikkinen, Miss. Laina female  26     0     0
## 4       Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1     0
## 5                           Allen, Mr. William Henry   male  35     0     0
## 6                                   Moran, Mr. James   male  NA     0     0
##            Ticket     Fare Cabin Embarked
## 1       A/5 21171  7.2500              S
## 2        PC 17599 71.2833   C85        C
## 3 STON/O2. 3101282  7.9250              S
## 4          113803 53.1000  C123        S
## 5          373450  8.0500              S
## 6          330877  8.4583              Q
```

```
head(test)
```

```
##   PassengerId Pclass                                          Name    Sex  Age
## 1         892      3                              Kelly, Mr. James   male 34.5
## 2         893      3                  Wilkes, Mrs. James (Ellen Needs) female 47.0
## 3         894      2                     Myles, Mr. Thomas Francis   male 62.0
## 4         895      3                              Wirz, Mr. Albert   male 27.0
## 5         896      3 Hirvonen, Mrs. Alexander (Helga E Lindqvist) female 22.0
## 6         897      3                    Svensson, Mr. Johan Cervin   male 14.0
##   SibSp Parch  Ticket    Fare Cabin Embarked
## 1     0     0  330911  7.8292              Q
## 2     1     0  363272  7.0000              S
## 3     0     0  240276  9.6875              Q
## 4     0     0  315154  8.6625              S
## 5     1     1 3101298 12.2875              S
## 6     0     0    7538  9.2250              S
```

There are a few column names that aren't descriptive enough. So I'm creating a descriptive data frame.

```
variable_name <- c('Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp', 'Parch', 'Ticket', 'Fare', 'Cabine', 'Embarked')
description <- c('Survived (1) or died (0)', 'Passenger's class', 'Passenger's name', 'Passenger's sex', 'Passenger's age', 'Number of siblings/spouses aboa
rd', 'Number of parents/children aboard', 'Ticket number', 'Fare', 'Cabine', 'Port of embarkation')
variable_description <- data.frame(variable_name, description)
head(variable_description, 11)
```

```
##    variable_name                  description
## 1       Survived         Survived (1) or died (0)
## 2         Pclass               Passenger's class
## 3           Name                Passenger's name
## 4            Sex                Passenger's sex
## 5            Age                Passenger's age
## 6          SibSp Number of siblings/spouses aboard
## 7          Parch Number of parents/children aboard
## 8         Ticket                   Ticket number
## 9           Fare                            Fare
## 10       Cabine                          Cabine
## 11     Embarked              Port of embarkation
```

Choosing variables to build a model

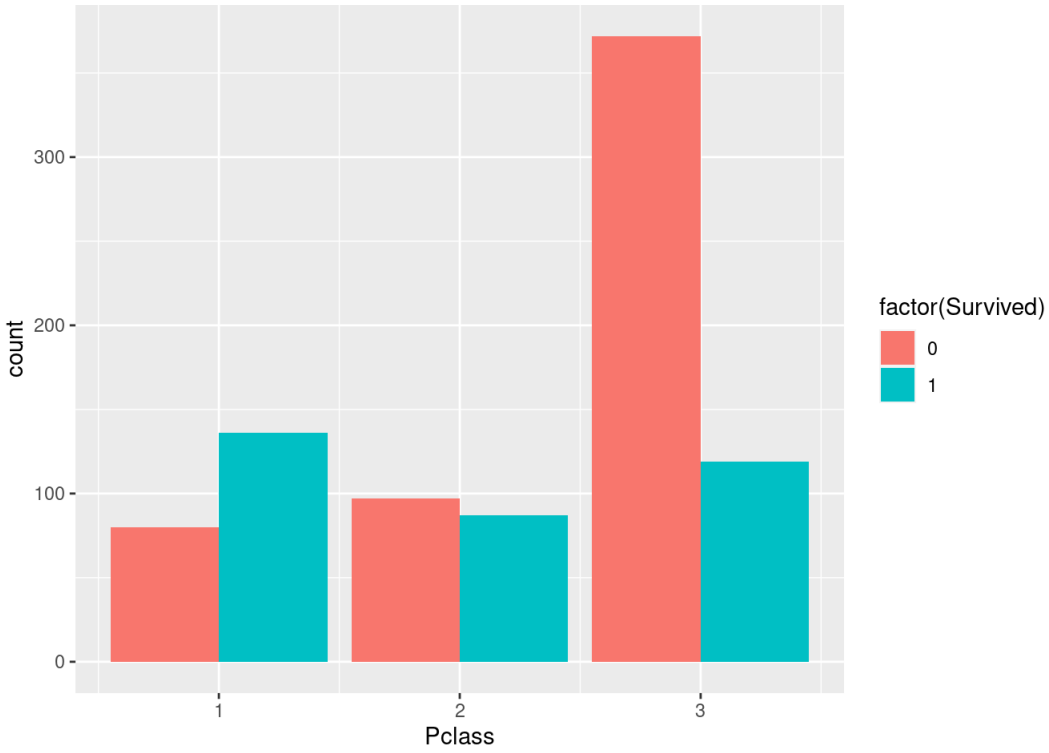
So, now I have to pick the best features to use in the model. And, I'm going to do it using cross-tabs for categorical variables and conditional box

plots for continuous ones.

```
table(train[,c('Survived', 'Pclass')])
```

```
##          Pclass
## Survived   1   2   3
##        0  80  97 372
##        1 136  87 119
```

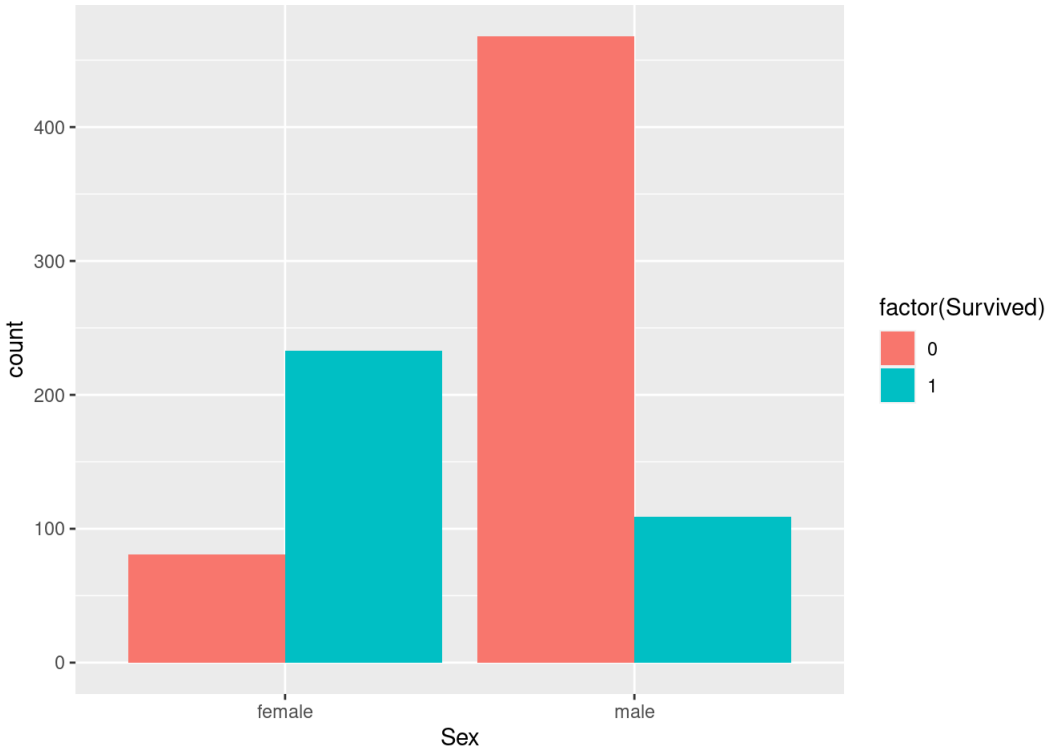
```
train %>%
 ggplot( aes(x = Pclass, fill = factor(Survived))) +
 geom_bar(position = "dodge")
```




```
table(train[,c('Survived', 'Sex')])
```

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

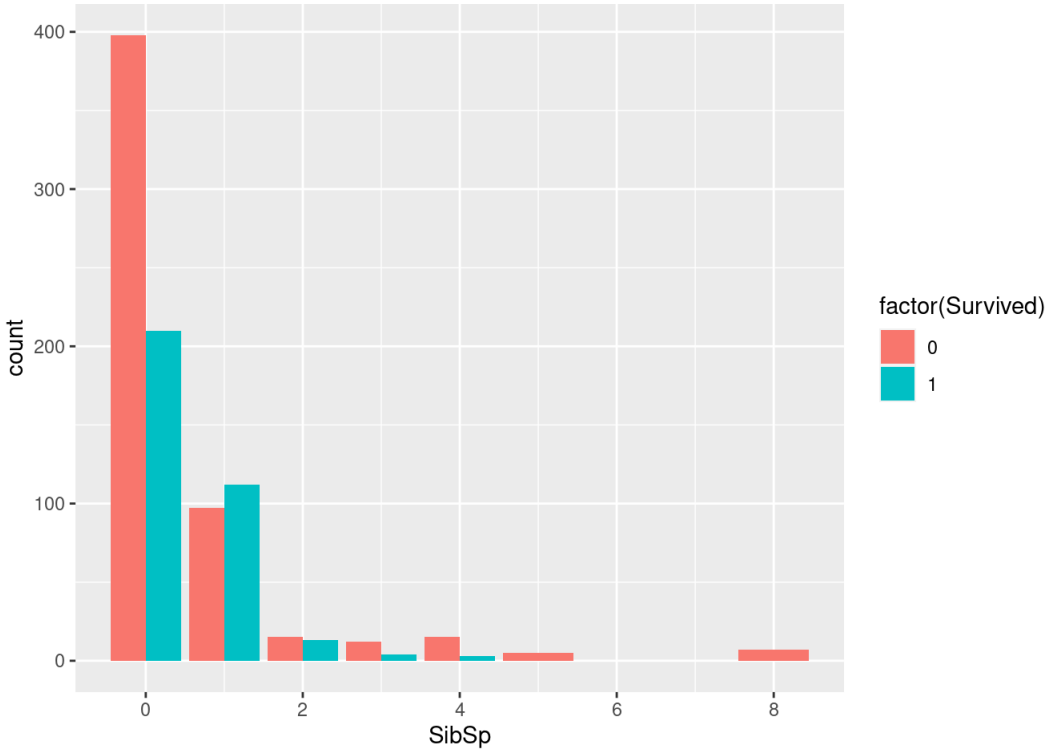
```
train %>%
 ggplot( aes(x = Sex, fill = factor(Survived))) +
 geom_bar(position = "dodge")
```




```
table(train[,c('Survived', 'SibSp')])
```

```
##        SibSp
## Survived   0   1   2   3   4   5   8
##        0 398  97  15  12  15   5   7
##        1 210 112  13   4   3   0   0
```

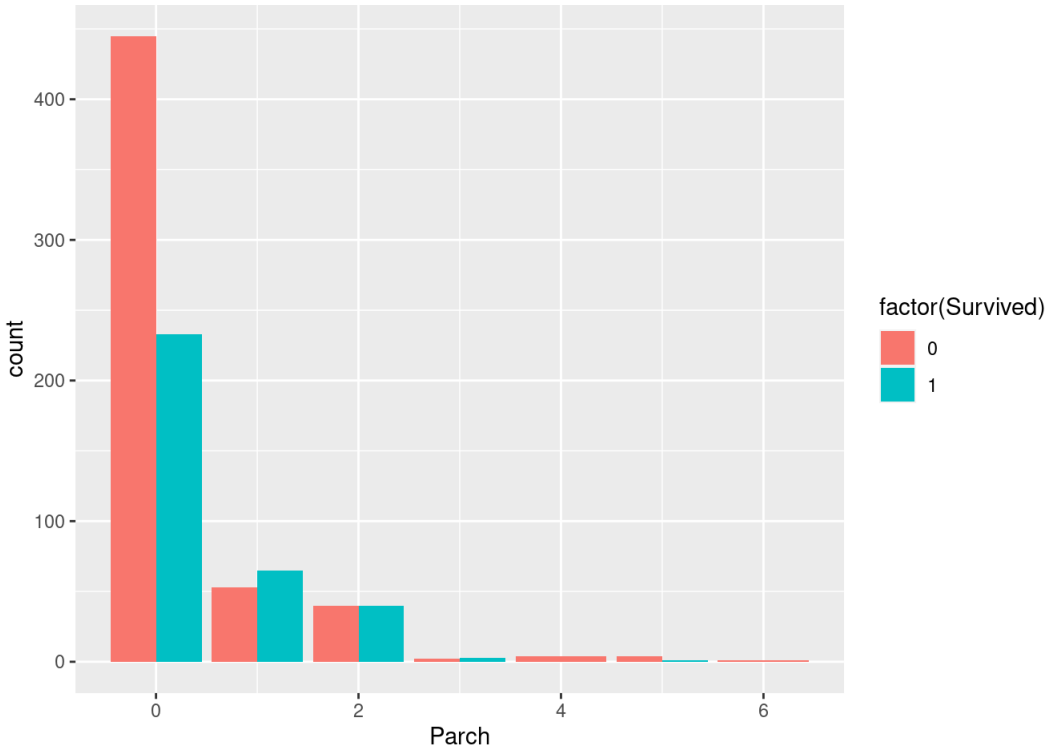
```
train %>%
  ggplot( aes(x = SibSp, fill = factor(Survived))) +
  geom_bar(position = "dodge")
```




```
table(train[,c('Survived', 'Parch')])
```

```
##        Parch
## Survived   0   1   2   3   4   5   6
##        0 445  53  40   2   4   4   1
##        1 233  65  40   3   0   1   0
```

```
train %>%
  ggplot( aes(x = Parch, fill = factor(Survived))) +
  geom_bar(position = "dodge")
```



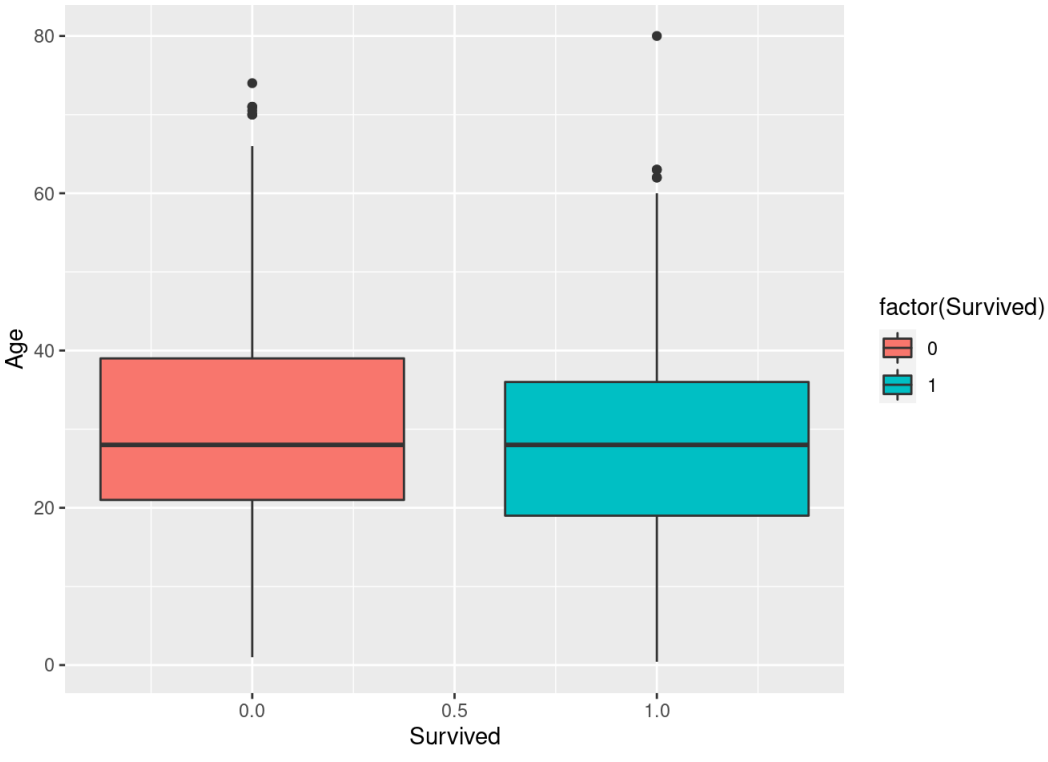

```
table(train[,c('Survived', 'Embarked')])
```

```
##        Embarked
## Survived   C  Q  S
##        0  75 47 427
##        1  93 30 217
```

We can see that all variables above can be useful predictors of *Survived*. This is because as it can be seen, the number of surviving passengers changes a lot for each value of every variable.

```
train %>%
  ggplot(aes(x = Survived, y = Age, fill = factor(Survived))) +
  geom_boxplot()
```

```
## Warning: Removed 177 rows containing non-finite values (stat_boxplot).
```

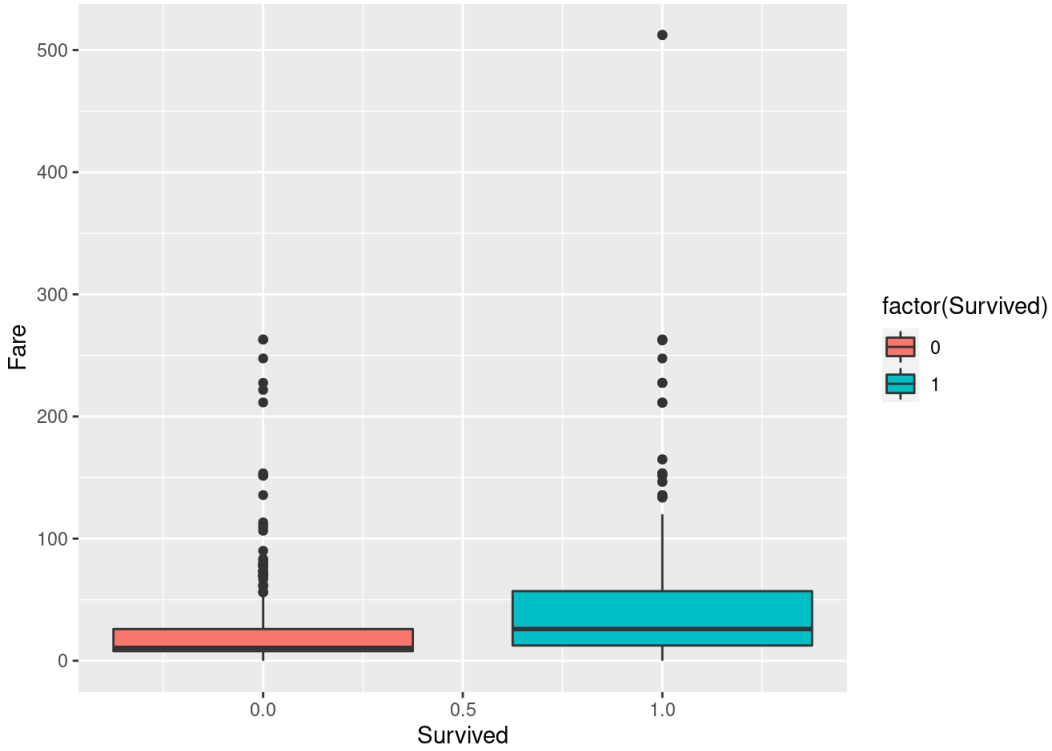


```
summary(train$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.42   20.12   28.00   29.70   38.00   80.00     177
```

As you can see, the box plot of age for people who survived and who didn't is nearly the same. This means that *Age* of a person did not have the biggest effect on whether one survived or not. Also, if you summarize it, there are lots of NA's. So for now, I´m going to exclude the variable *Age* from the model.

```
train %>%
 ggplot(aes(x = Survived, y = Fare, fill = factor(Survived))) +
 geom_boxplot()
```




```
summary(train$Fare)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    7.91   14.45   32.20   31.00  512.33
```

Here, we can see a difference between those who survived, and those who didn't as far as *fare* value. So, I'm including it for the model.

Creating the predictive model

```
# Converting "Survived" to a factor
train$Survived <- factor(train$Survived)

# Set a random seed
set.seed(5)

# Training using "random forest" algorithm
model_1 <- train(Survived ~ Pclass + Sex + SibSp + Embarked + Parch + Fare,
        data = train,
        method = 'rf',
        trControl = trainControl(method = 'cv', number = 5))
model_1
```

```
## Random Forest
##
## 891 samples
##   6 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 712, 713, 713, 713, 713
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.8080472  0.5717891
##   5     0.8103132  0.5874140
##   8     0.8159312  0.6037906
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 8.
```

```
summary(test)
```

```
##   PassengerId         Pclass          Name               Sex
##  Min.   : 892.0   Min.   :1.000   Length:418         Length:418
##  1st Qu.: 996.2   1st Qu.:1.000   Class :character   Class :character
##  Median :1100.5   Median :3.000   Mode  :character   Mode  :character
##  Mean   :1100.5   Mean   :2.266
##  3rd Qu.:1204.8   3rd Qu.:3.000
##  Max.   :1309.0   Max.   :3.000
##
##       Age            SibSp            Parch           Ticket
##  Min.   : 0.17   Min.   :0.0000   Min.   :0.0000   Length:418
##  1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
##  Median :27.00   Median :0.0000   Median :0.0000   Mode  :character
##  Mean   :30.27   Mean   :0.4474   Mean   :0.3923
##  3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :76.00   Max.   :8.0000   Max.   :9.0000
##  NA's   :86
##       Fare            Cabin             Embarked
##  Min.   :  0.000   Length:418         Length:418
##  1st Qu.:  7.896   Class :character   Class :character
##  Median : 14.454   Mode  :character   Mode  :character
##  Mean   : 35.627
##  3rd Qu.: 31.500
##  Max.   :512.329
##  NA's   :1
```

We can see that the variable *Fare* has one NA value. so, let's replace it with the mean of *Fare* column.

```
test$Fare <- ifelse(is.na(test$Fare), mean(test$Fare, na.rm = TRUE), test$Fare)
```

Now, we are ready to make predictions on the *test* set.

```
test$Survived <- predict(model_1, newdata = test)
 test$Survived
```

```
##   [1] 0 1 0 0 1 0 0 0 1 0 0 0 1 0 1 1 0 0 0 1 1 0 1 0 1 0 1 0 1 0 0 0 1 0 1 0 0
##  [38] 0 0 1 1 0 0 1 1 0 0 0 1 1 0 0 1 1 0 0 0 0 0 1 0 0 0 1 0 1 1 0 0 1 1 0 1 0
##  [75] 1 0 0 1 0 1 1 0 0 0 0 0 1 0 1 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0
## [112] 1 1 1 1 0 0 1 1 1 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
## [149] 0 1 1 0 0 1 0 0 1 0 0 1 1 1 1 0 0 1 0 0 1 0 0 0 0 0 0 1 1 1 1 1 0 0 1 0 1
## [186] 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 1 0 0 0 0 1 1 0 1 0 1 0 1 0
## [223] 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 1 1 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 1
## [260] 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0
## [297] 1 0 0 0 0 0 1 0 1 1 1 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0
## [334] 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 1 0 0 0 1 0 1 0 0 1 0 1 1 1 0 1 0 0 0 1 1
## [371] 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 1 0 0 1 0 1 0 0 1 0 1 0 0 1 0
## [408] 0 1 0 1 1 0 0 1 0 0 0
## Levels: 0 1
```

- As a result, it could be said, that we create *Survived column* for the *test data frame* based on the data we had previously, with an accuracy of up to 82%.
- Now, I would like to go further including a new variable that caught my attention, to see if this improves our model's accuracy.

The new variable is going to be called *Title*, but to be able to use it in our analysis I have to extract it from *Name* column.

```
train <- separate(train, Name, into = c("Last_name", "Title_first_name"), sep = ", ")
train <- separate(train, Title_first_name, into = c("Title", "First_name"), sep = "[.] ")
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 1 rows [514].
```

```
head(train)
```

```
##   PassengerId Survived Pclass Last_name Title
## 1           1        0      3    Braund    Mr
## 2           2        1      1   Cumings   Mrs
## 3           3        1      3 Heikkinen  Miss
## 4           4        1      1  Futrelle   Mrs
## 5           5        0      3     Allen    Mr
## 6           6        0      3     Moran    Mr
##                          First_name    Sex Age SibSp Parch           Ticket
## 1                        Owen Harris   male  22     1     0        A/5 21171
## 2 John Bradley (Florence Briggs Thayer) female  38     1     0         PC 17599
## 3                            Laina female  26     0     0 STON/O2. 3101282
## 4        Jacques Heath (Lily May Peel) female  35     1     0           113803
## 5                      William Henry   male  35     0     0           373450
## 6                            James   male  NA     0     0           330877
##      Fare Cabin Embarked
## 1  7.2500              S
## 2 71.2833   C85        C
## 3  7.9250              S
## 4 53.1000  C123        S
## 5  8.0500              S
## 6  8.4583              Q
```

I have already split *Name* column into three new ones: *Last name*, *Title* and *First name*.

```
table(train$Sex, train$Title)
```

```
##
##          Capt Col Don  Dr Jonkheer Lady Major Master Miss Mlle Mme  Mr Mrs  Ms
##   female    0   0   0   1        0    1     0      0  182    2   1   0 125   1
##   male      1   2   1   6        1    0     2     40    0    0   0 517   0   0
##
##          Rev Sir the Countess
##   female   0   0            1
##   male     6   1            0
```

I am going to join those values that appear just a few times, into a new value called *Others*

```
Others <- c('Capt', 'Col', 'Don', 'Dr', 'Jonkheer', 'Lady', 'Major', 'Master', 'Rev', 'Sir', 'the Countess')
Others
```

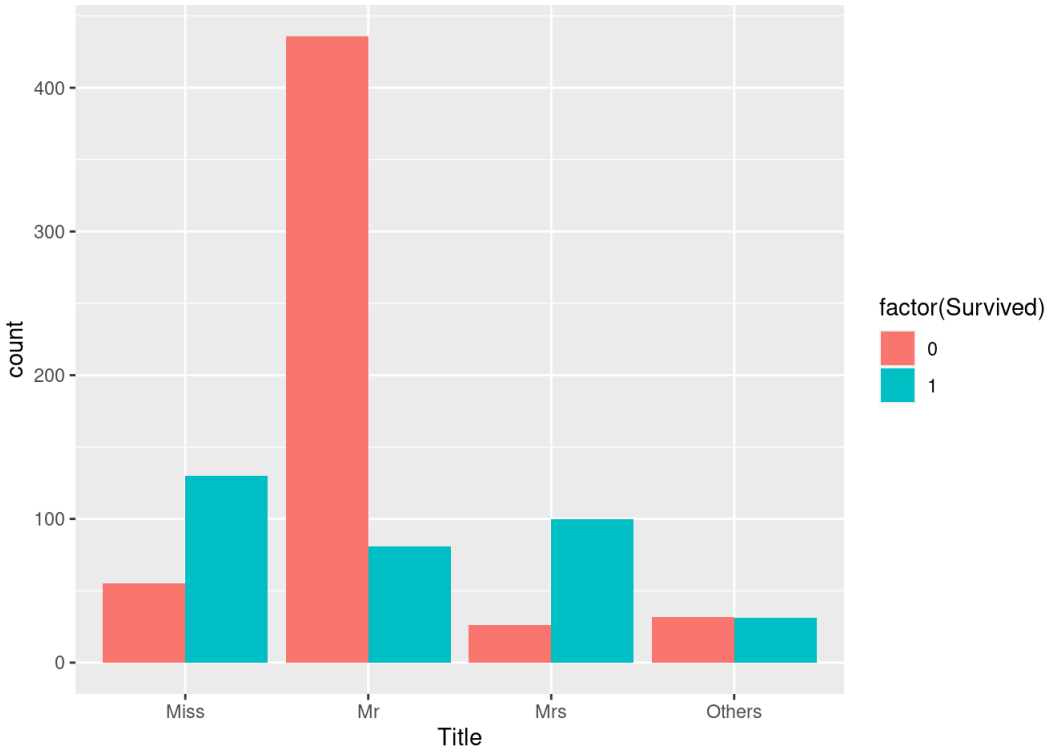
```
##  [1] "Capt"         "Col"          "Don"          "Dr"           "Jonkheer"
##  [6] "Lady"         "Major"        "Master"       "Rev"          "Sir"
## [11] "the Countess"
```

Also, I have to fix some typos. So, in the end, I am going to keep just the most representative values. And, as could be seen in the plot below, surviving seems to be quite different for each value of *Title* variable.

```
train$Title[train$Title == 'Mlle'] <- 'Miss'
train$Title[train$Title == 'Mme'] <- 'Mrs'
train$Title[train$Title == 'Ms'] <- 'Miss'
train$Title[train$Title %in% Others] <- 'Others'
table(train$Sex, train$Title)
```

```
##
##          Miss  Mr Mrs Others
##   female  185   0 126      3
##   male      0 517   0     60
```

```
train %>%
  ggplot( aes(x = Title, fill = factor(Survived))) +
  geom_bar(position = "dodge")
```



Before running the new model, I need to create the *Title* column from the *test* data set as well.

```
test <- separate(test, Name, into = c("Last_name", "Title_first_name"), sep = ", ")
test <- separate(test, Title_first_name, into = c("Title", "First_name"), sep = "[.] ")
head(test)
```

```
##   PassengerId Pclass Last_name Title                 First_name    Sex  Age
## 1         892      3     Kelly    Mr                      James   male 34.5
## 2         893      3    Wilkes   Mrs          James (Ellen Needs) female 47.0
## 3         894      2     Myles    Mr              Thomas Francis   male 62.0
## 4         895      3      Wirz    Mr                     Albert   male 27.0
## 5         896      3  Hirvonen   Mrs Alexander (Helga E Lindqvist) female 22.0
## 6         897      3  Svensson    Mr               Johan Cervin   male 14.0
##   SibSp Parch  Ticket    Fare Cabin Embarked Survived
## 1     0     0  330911  7.8292             Q        0
## 2     1     0  363272  7.0000             S        1
## 3     0     0  240276  9.6875             Q        0
## 4     0     0  315154  8.6625             S        0
## 5     1     1 3101298 12.2875             S        1
## 6     0     0    7538  9.2250             S        0
```

```
table(test$Sex, test$Title)
```

```
##
##          Col Dona  Dr Master Miss  Mr Mrs  Ms Rev
##   female   0    1   0      0   78   0  72   1   0
##   male     2    0   1     21    0 240   0   0   2
```

```
Others1 <- c('Col', 'Dona', 'Dr', 'Master', 'Rev')
Others1
```

```
## [1] "Col"    "Dona"   "Dr"     "Master" "Rev"
```

```
test$Title[test$Title == 'Ms'] <- 'Miss'
test$Title[test$Title %in% Others1] <- 'Others'
table(test$Sex, test$Title)
```

```
##
##          Miss  Mr Mrs Others
##   female   79   0  72      1
##   male      0 240   0     26
```

Creating a new predictive model

```
# Converting "Survived" to a factor
train$Survived <- factor(train$Survived)

# Set a random seed
set.seed(5)

# Training using "random forest" algorithm
model_2 <- train(Survived ~ Pclass + Sex + SibSp + Embarked + Parch + Fare + Title,
        data = train,
        method = 'rf',
        trControl = trainControl(method = 'cv', number = 5))
model_2
```

```
## Random Forest
##
## 891 samples
##   7 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 712, 713, 713, 713, 713
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.8215617  0.6127758
##    6    0.8361371  0.6464882
##   11    0.8305254  0.6373961
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 6.
```

```
summary(test)
```

```
##   PassengerId         Pclass       Last_name           Title
##  Min.   : 892.0   Min.   :1.000   Length:418         Length:418
##  1st Qu.: 996.2   1st Qu.:1.000   Class :character   Class :character
##  Median :1100.5   Median :3.000   Mode  :character   Mode  :character
##  Mean   :1100.5   Mean   :2.266
##  3rd Qu.:1204.8   3rd Qu.:3.000
##  Max.   :1309.0   Max.   :3.000
##
##   First_name           Sex                 Age            SibSp
##  Length:418         Length:418         Min.   : 0.17   Min.   :0.0000
##  Class :character   Class :character   1st Qu.:21.00   1st Qu.:0.0000
##  Mode  :character   Mode  :character   Median :27.00   Median :0.0000
##                                        Mean   :30.27   Mean   :0.4474
##                                        3rd Qu.:39.00   3rd Qu.:1.0000
##                                        Max.   :76.00   Max.   :8.0000
##                                        NA's   :86
##      Parch           Ticket               Fare            Cabin
##  Min.   :0.0000   Length:418         Min.   :  0.000   Length:418
##  1st Qu.:0.0000   Class :character   1st Qu.:  7.896   Class :character
##  Median :0.0000   Mode  :character   Median : 14.454   Mode  :character
##  Mean   :0.3923                      Mean   : 35.627
##  3rd Qu.:0.0000                      3rd Qu.: 31.500
##  Max.   :9.0000                      Max.   :512.329
##
##    Embarked         Survived
##  Length:418         0:276
##  Class :character   1:142
##  Mode  :character
##
##
##
##
```

We can see that the variable *Fare* has one NA value. so, let's replace it with the mean of *Fare* column.

```
test$Fare <- ifelse(is.na(test$Fare), mean(test$Fare, na.rm = TRUE), test$Fare)
```

Now, we are ready to make predictions on the *test* set.

```
test$Survived <- predict(model_2, newdata = test)
 test$Survived
```

```
##   [1] 0 1 0 0 1 0 1 0 1 0 0 0 1 0 1 1 0 0 0 1 1 1 1 0 1 0 1 0 0 0 0 0 1 0 1 0 0
##  [38] 0 0 1 0 1 0 1 1 0 0 0 1 1 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 1 0 0 1 1 0 0 0
##  [75] 1 0 0 1 0 1 1 0 0 0 0 0 1 0 1 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0
## [112] 1 1 1 1 0 0 1 0 1 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
## [149] 1 1 1 0 0 1 0 0 1 0 1 1 1 1 1 0 0 1 0 0 1 0 0 0 0 0 0 1 1 1 1 1 0 1 1 0 1
## [186] 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 1 1 1 1 0 0 1 0 1 0 0 0 0 1 1 0 1 0 1 0 1 0
## [223] 1 0 1 1 0 1 0 0 0 1 0 0 1 0 1 0 1 1 1 1 0 0 1 0 1 0 1 1 1 0 0 0 0 0 0 0 1
## [260] 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0
## [297] 1 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0
## [334] 1 0 0 0 0 0 0 1 0 0 0 1 1 1 0 1 0 1 1 0 0 0 1 0 1 0 0 1 0 1 1 0 1 0 0 0 1
## [371] 0 1 0 0 1 1 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 1 1 0 0 1 0 1 0 0 1 0 1 0 0 0
## [408] 0 1 1 1 1 0 0 1 0 0 1
## Levels: 0 1
```

Finally, we can see how including this new variable improved our model accuracy, therefore our predictions as well (Accuracy went up to almost 84%). I am happy with this new result and I will save the solution.

```
solution <- data.frame(PassengerId = test$PassengerId, Survived = test$Survived)
head(solution)
```

```
##   PassengerId Survived
## 1         892        0
## 2         893        1
## 3         894        0
## 4         895        0
## 5         896        1
## 6         897        0
```

```
write.csv(solution, file = 'Titanic_Solution.csv', row.names = F)
```

Thanks!!