Titanic App.

BEYZA BAKIRTAS

4/9/2020

Contents

Data-Set Preparition

```
train=read.csv("train.csv")
test=read.csv("test.csv")
gender=read.csv("gender_submission.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
```

Train and test data set can be merged. Test is added to final of train data set but they have to have same columns.Also we can see that test data set does not have "Survived" column.Therefore if we want to merge them , we have to create new column for test data-set.

```
test$Survived=NA
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 Survived
## 1     0     0  330911  7.8292              Q       NA
## 2     1     0  363272  7.0000              S       NA
## 3     0     0  240276  9.6875              Q       NA
## 4     0     0  315154  8.6625              S       NA
## 5     1     1 3101298 12.2875              S       NA
## 6     0     0    7538  9.2250              S       NA
```

```
total.data=rbind(train,test)
```

```
str(total.data)
```

```
## '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       : Factor w/ 1307 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 417 58
##  $ 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     : Factor w/ 929 levels "110152","110413",..: 524 597 670 50 473 276 86 396 345 133 ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : Factor w/ 187 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ...
##  $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

To explain each column one by one (What does each column mean,which variable type are they and interpretation of data)

PassengerId: This data is not important because this is just numbers which is just a rank own each passengers.

Survived: This means that the passenger survives(1) or does not(0). It is integer currently but it can be Factor class.

Pclass: It is class of passengers and it will be converted to Factor type. 1st = Upper 2nd = Middle 3rd = Lower

Name: It can changed with character type to interpret according to the names of the passengers next process.

Sibsp: The dataset defines family relations in this way... Sibling = brother, sister, stepbrother, stepsister Spouse = husband, wife (mistresses and fiancés were ignored) *(from kaggle)*

Parch: The dataset defines family relations in this way... Parent = mother, father Child = daughter, son, stepdaughter, stepson Some children travelled only with a nanny, therefore parch=0 for them. *(from kaggle)*

Embarked: C = Cherbourg, Q = Queenstown, S = Southampton

Ticket and Cabin: They will be converted to character class.

Rest of variables do not have special notes and will remain same.

In this step, the above mentioned conversions will be done with using "dplyr" package and mutate function in this package.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
 total.data=total.data%>%mutate(Pclass=factor(Pclass),Survived=factor(Survived),Ticket=as.character(Tic
```

Data is checked for the type of indeks for the last time.

```
str(total.data)
```

```
## '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)"
##  $ 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      : Factor w/ 187 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ...
##  $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

Missing Values

```
summary(total.data)
```

```
##   PassengerId   Survived   Pclass      Name              Sex
##  Min.   :   1   0   :549   1:323   Length:1309        female:466
##  1st Qu.: 328   1   :342   2:277   Class :character   male  :843
##  Median : 655   NA's:418   3:709   Mode  :character
##  Mean   : 655
##  3rd Qu.: 982
##  Max.   :1309
##
##       Age            SibSp            Parch          Ticket
##  Min.   : 0.17   Min.   :0.0000   Min.   :0.000   Length:1309
##  1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.000   Class :character
##  Median :28.00   Median :0.0000   Median :0.000   Mode  :character
##  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
##       Fare                     Cabin      Embarked
##  Min.   :  0.000                  :1014    :  2
##  1st Qu.:  7.896   C23 C25 C27    :   6   C:270
##  Median : 14.454   B57 B59 B63 B66:   5   Q:123
##  Mean   : 33.295   G6             :   5   S:914
##  3rd Qu.: 31.275   B96 B98        :   4
##  Max.   :512.329   C22 C26        :   4
##  NA's   :1         (Other)        : 271
```

The NA values in Survived column are due to the test data we add to the end of this data. Therefore data will be used between 1 and 891 lines. "total.data[1:891,]"

On the other hand, in Age indeks , a significant amount of NA value can be seen obviously. But we will check NA values with other way to be sure.(with "is.na" function)

```
sum(is.na(total.data$Sex))
```

```
## [1] 0
```

If result of this function is true, there is na value. Otherwise, there is not na value.

To apply for all columns.

```
apply(total.data,2,function(x) sum(is.na(x)))
```

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

I will use Mr,Mrs,Miss etc. in Name variable to estimate NA values in Age variable. For example, if it is Mrs in Name variable but there is nothing in Age variable, so it can be written as avarage of rest of people age who have Mrs. title. Regular Expressions will be used to separate people's title from their name.

```
title=sub(".*,.([^.]*)\\..*","\\1",total.data$Name)
total.data$title=title
total.data=total.data%>%mutate(title=factor(title))
head(total.data)
```

```
##   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 title
## 1        A/5 21171  7.2500              S    Mr
## 2         PC 17599 71.2833   C85        C   Mrs
## 3 STON/O2. 3101282  7.9250              S  Miss
## 4           113803 53.1000  C123        S   Mrs
## 5           373450  8.0500              S    Mr
## 6           330877  8.4583              Q    Mr
```

```
levels(total.data$title)
```

```
##  [1] "Capt"         "Col"          "Don"          "Dona"         "Dr"
##  [6] "Jonkheer"     "Lady"         "Major"        "Master"       "Miss"
## [11] "Mlle"         "Mme"          "Mr"           "Mrs"          "Ms"
## [16] "Rev"          "Sir"          "the Countess"
```

Some titles have the same meaning as each others. Therefore they will be merged.

```
library(forcats)

total.data=total.data%>%mutate(title=fct_collapse(title,"Miss"=c("Mlle","Ms"),"Mrs"=c("Mme"),"Ranked"=c

levels(total.data$title)
```

```
## [1] "Ranked"  "Royalty" "Master"  "Miss"    "Mrs"     "Mr"
```

In next step, missing values will be completed according to median values of rest of members in their group. These groups will be created by title.

```
total.data=total.data%>%group_by(title)%>%mutate(Age=ifelse(is.na(Age),round(median(Age,na.rm=T),1),Age)
```

To handle the missing value in FARE column.

```
total.data%>%filter(is.na(Fare))
```

```
## # A tibble: 1 x 13
## # Groups:   title [1]
##   PassengerId Survived Pclass Name  Sex     Age SibSp Parch Ticket  Fare Cabin
##         <int> <fct>    <fct>  <chr> <fct> <dbl> <int> <int> <chr>  <dbl> <fct>
## 1        1044 <NA>     3      Stor~ male   60.5     0     0 3701      NA ""
## # ... with 2 more variables: Embarked <fct>, title <fct>
```

```
Fare=ifelse(is.na(total.data$Fare), round(median(total.data$Fare, na.rm=T),1),
            total.data$Fare)

total.data$Fare=Fare
```

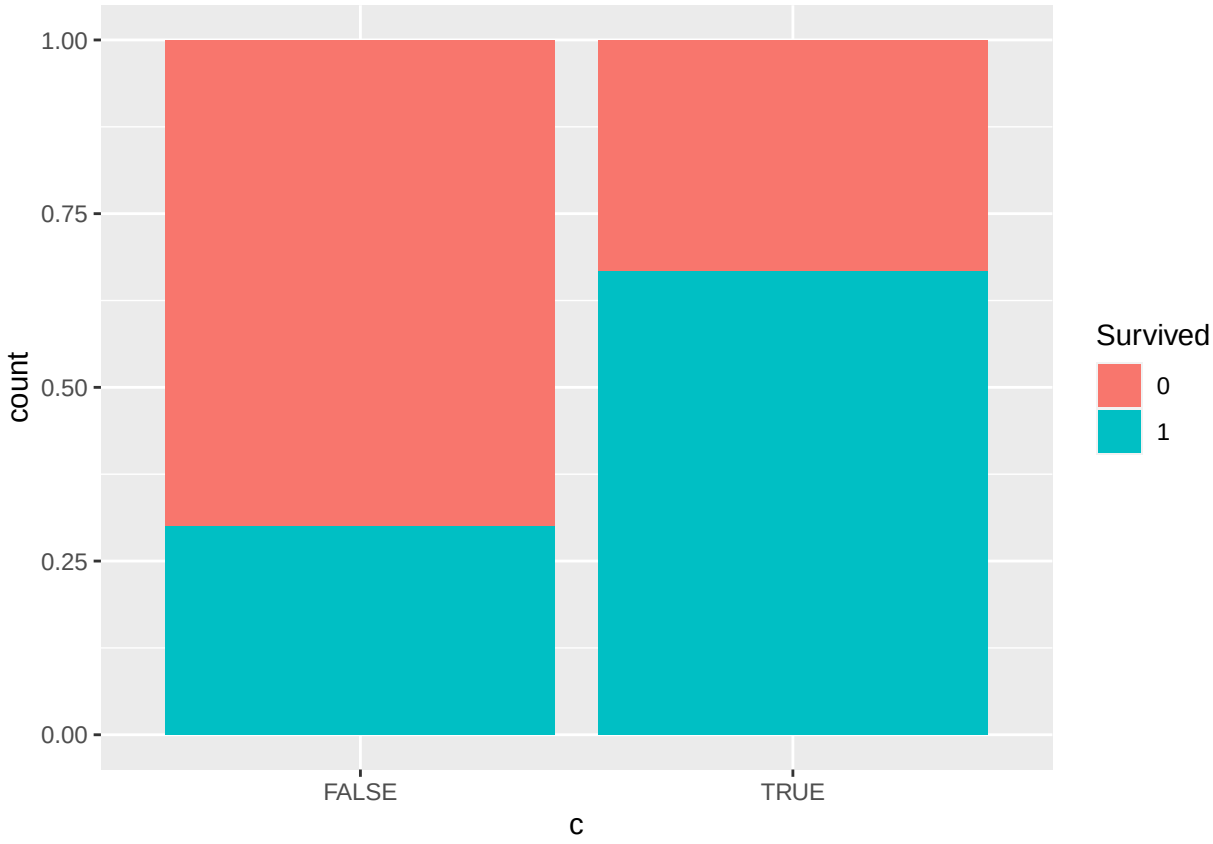
Besides, there are not cabin numbers of some passengers. My opinion, passengers whose cabin number is unknown are the less important passengers and most of them are 3rd class.So, may be the information can be useful.

```
c=ifelse(total.data$Cabin == "",FALSE,TRUE)
total.data$c=c

library(ggplot2)
library(devtools)
```

```
## Loading required package: usethis
```

```
ggplot(total.data[1:891,])+ geom_bar(mapping=aes (x=c, fill= Survived),position = "fill")
```



Prediction

Train and Test Sets

```
train=total.data[1:891,]

test=total.data[892:1309,]
##this test set can not check by us. Because this set of kaggle and answers are not shared.
```

Train set will be used in this period and so a test set is needed to check result which will be obtanined.

```
test.1=train[711:891,]
train.1=train[1:710,]
```

Prediction with RAndom Forest

```
library(randomForest)
```

```
## randomForest 4.6-14
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:ggplot2':
##
##     margin
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
rf=randomForest(Survived~Pclass+Sex+Age+SibSp+Fare+c,data=train,mtry=3,ntree=1000 )
```

```
set.seed(1234)
predict.=predict(rf, test.1[,c(3,5,6,7,8,10,14)])

predict.
```

```
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   1   0   1   0   0   0   1   1   0   0   1   0   0   0   1   0   1   1   0   0
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   1   0   0   0   0   0   0   1   0   0   1   0   1   0   0   0   0   1   0   0
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   1   1   0   0   1   1   0   0   0   1   0   0   0   1   0   1   0   0   0   0
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   0   0   0   0   1   0   0   1   0   1   1   1   1   0   0   0   1   0   1   0
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   0   0   0   0   0   0   1   1   0   0   0   1   1   1   1   0   0   1   0   1
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
##   0   0   0   0   0   0   0   0   0   0   1   0   0   1   0   0   1   1   0   1
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140
##   1   1   0   0   0   1   0   0   1   1   0   0   1   0   0   0   0   0   0   1
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160
##   0   0   0   1   1   1   1   1   1   0   0   0   1   0   0   1   1   0   0   1
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
##   0   1   0   0   1   1   0   0   0   1   1   0   0   0   0   0   0   1   0   1
## 181
##   0
## Levels: 0 1
```

```
table(predict., test.1$Survived)
```

```
##
## predict.   0   1
##        0 112   4
##        1   4  61
```

Percentage of correctly estimated:

```
(112+61)/181
```

```
## [1] 0.9558011
```