Titanic prediction model

Yoan Bidart

11/16/2017

Introduction

For this analysis about Titanic survival on Kaggle, we will first do an exploratory analysis of the data. Then we will clean the dataset and create new interesting variables for prediction. Finally we will create a prediction model using RandomForest and submit the predicted values.

Preprocessing

Load and visualise the data

```
library(ggplot2)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.2
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.1
```

```
allData <- read.csv("train.csv", stringsAsFactors = FALSE)
str(allData)
```

```
## 'data.frame':    891 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)"
##  $ 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" ...
```

We have 891 observations of 12 variable, and our aim is to predict the Survival state 1 or 0.

Preprocessing

After looking carefully at each variable (not included in this report as it would make it quite long for an easy reading), we find some interesting points to model our data, and created a preProcess function.

Names

```
head(allData$Name)
```

```
## [1] "Braund, Mr. Owen Harris"
## [2] "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
## [3] "Heikkinen, Miss. Laina"
## [4] "Futrelle, Mrs. Jacques Heath (Lily May Peel)"
## [5] "Allen, Mr. William Henry"
## [6] "Moran, Mr. James"
```

For prediction model we can use an interesting thing in this feature : the title , some are in french and need to be translated, and noble titles can be useful for predicting survival. The full names will be removed.

Age

```
table(is.na(allData$Age))
```

```
##
## FALSE  TRUE
##   714   177
```

We choose to replace NA by the median value of the variable

Cabin, Ticket

```
table(allData$Cabin=="")
```

```
##
## FALSE  TRUE
##   204   687
```

As more than the half of Cabin values are empty, we will remove this column for the prediction. As we have also the fare and the class we can assume that these values are highly linked to the cabin. Ticket number will also be removed for the prediction model.

preProcess function

The function aggregate these steps for an easy use.

```
preProcess <- function(x) {
        temp <- strsplit(as.character(x$Name), ", ")
        #Title
        title <- NULL
        for (i in 1:length(temp)) {
                temp2 <- strsplit(temp[[i]][2], " ")
                temp3 <- temp2[[1]][1]
                title <- c(title, temp3)
        }

        wom <- c("Mme.|Ms.")
        title <- gsub(wom, "Mrs.", title)
        title <- gsub("Mlle.","Miss.", title)
```

```r
        highGrade <- c("Master.|Don.|Rev.|Dr.|Major.|Lady.|Sir.|Col.|Capt.|
                       Jonkheer.|the")
        title <- gsub(highGrade, "Noble.", title)

        x <- cbind(x, title)

        #Age
        index <- is.na(x$Age)
        x$Age[index] <- median(x$Age, na.rm=TRUE)

        #Cabin
        x <- select(x, -c(Cabin, Ticket, Name))

        #classes
        x$title <- as.character(x$title)
        x
}

allData <- preProcess(allData)
```

Create training and testing datasets

We will use 70% of the data for the training set and 30% for the testing set.

```r
set.seed(12345)
inTrain <- createDataPartition(allData$PassengerId, p=.7, list=FALSE)
training <- allData[inTrain,]
testing <- allData[-inTrain,]
```

Prediction model

We chose to use Random Forest for predicting the survival, to arrive at a theorical accuracy of roughly 83.3%. We tried a lot of ways including combining different models, but it showed no benefit for the accuracy.

```r
training$Survived <- as.factor(training$Survived)
test1 <- select(testing, -Survived)
#random forest model
control <- trainControl(method="cv", number=3, verboseIter=FALSE)
fit1 <- train(Survived~., data=training, method="rf", trControl=control)
```

```
## randomForest 4.6-12

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

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

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

```
pred1 <- predict(fit1, test1)
accu1 <- confusionMatrix(testing$Survived, pred1)
accu1$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##   8.333333e-01   6.333333e-01   7.827908e-01   8.762117e-01   6.666667e-01
## AccuracyPValue  McnemarPValue
##   9.093940e-10   2.912928e-01
```

Predicting and creating output

Here we are ! Let's predict on the test.csv file to create our output for the Kaggle Competition.

```
input <- read.csv("test.csv", stringsAsFactors = FALSE)
input <- preProcess(input)
#repairing a missing title and a missing fare
input$title[415] <- "Mrs."
input$Fare[153] <- median(input$Fare, na.rm=TRUE)
output <- predict(fit1, input)

#Write the solution to a file
solution <- data.frame(PassengerId=input$PassengerId, Survived=output)
write.csv(solution, file="solution.csv", row.names=FALSE)
```

Thank you for reading !