Titanic -Kaggle

Shinjini

10 December 2017

TITANIC KAGGLE COMPETITION

-Competition Description

The sinking of the RMS Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. This sensational tragedy shocked the international community and led to better safety regulations for ships.

One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew. Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper-class.

In this challenge, we are asked to complete the analysis of what sorts of people were likely to survive. In particular, we ask you to apply the tools of machine learning to predict which passengers survived the tragedy.

- Import The Dataset

```
library(knitr)
train=read.csv("C://Users//Administrator//Desktop//KAGGLE//titanic.train.csv")
str(train)
```

```
## '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       : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520
629 417 581 ...
##  $ 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/ 681 levels "110152","110413",..: 524 597 670 50 473 276 86 396 3
45 133 ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : Factor w/ 148 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 ...
```

```
test=read.csv("C://Users//Administrator//Desktop//KAGGLE//titanic.test.csv")
test$Survived=NA
```

- Combine both the test and train into a single dataset
- This ensures that any changes made to the train dataset reflects on the test dataset
- Next, a new column Titles is formed from the Name Column

```
combi=rbind(train,test)
combi$Name=as.character(combi$Name)
combi$Titles <- sapply(combi$Name, FUN=function(x) {strsplit(x, split='[,.]')[[1]][2]})#extra
cting
combi$Titles <- sub(' ', '', combi$Titles)#removing spaces from the first
head(combi)
```

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

- Similar Titles are Clubbed Together

```
table(combi$Titles)
```

```
##
##         Capt          Col          Don         Dona           Dr
##            1            4            1            1            8
##     Jonkheer         Lady        Major       Master         Miss
##            1            1            2           61          260
##         Mlle          Mme           Mr          Mrs           Ms
##            2            1          757          197            2
##          Rev          Sir the Countess
##            8            1            1
```

```
combi$Titles[combi$Titles %in% c('Mme', 'Mlle')] <- 'Mlle'
combi$Titles[combi$Titles %in% c('Capt', 'Don', 'Major', 'Sir')] <- 'Sir'
combi$Titles[combi$Titles %in% c('Dona', 'Lady', 'the Countess', 'Jonkheer')] <- 'Lady'
combi$Titles <- factor(combi$Titles)
```

```
combi$FamilySize <- combi$SibSp + combi$Parch + 1

#extracting the surnames from the names
combi$Surname <- sapply(combi$Name, FUN=function(x) {strsplit(x, split='[,.]')[[1]][1]})
combi$FamilyID <- paste(as.character(combi$FamilySize), combi$Surname, sep="")
combi$FamilyID[combi$FamilySize <= 2] <- 'Small'

table(combi$FamilyID)
```

```
##
##            11Sage           3Abbott         3Appleton         3Beckwith
##                11                 3                 1                 2
##           3Boulos           3Bourke            3Brown         3Caldwell
##                 3                 3                 4                 3
##          3Christy          3Collyer          3Compton          3Cornell
##                 2                 3                 3                 1
##           3Coutts           3Crosby           3Danbom           3Davies
##                 3                 3                 3                 5
##            3Dodge          3Douglas             3Drew            3Elias
##                 3                 1                 3                 3
##       3Frauenthal        3Frolicher 3Frolicher-Stehli        3Goldsmith
##                 1                 1                 2                 3
##       3Gustafsson       3Hamalainen           3Hansen             3Hart
##                 2                 2                 1                 3
##             3Hays          3Hickman         3Hiltunen         3Hirvonen
##                 2                 3                 1                 1
##         3Jefferys          3Johnson             3Kink    3Kink-Heilmann
##                 2                 3                 2                 2
##           3Klasen         3Lahtinen           3Mallet            3McCoy
##                 3                 2                 3                 3
##          3Minahan         3Moubarek            3Nakid         3Navratil
##                 1                 3                 3                 3
##           3Newell           3Newsom         3Nicholls          3Peacock
##                 1                 1                 1                 3
##            3Peter            3Quick         3Richards          3Rosblom
##                 3                 3                 2                 3
##           3Samaan        3Sandstrom           3Silven          3Spedden
##                 3                 3                 1                 3
##            3Strom          3Taussig           3Thayer           3Thomas
##                 1                 3                 3                 1
##            3Touma     3van Billiard         3Van Impe    3Vander Planke
##                 3                 3                 3                 2
##            3Wells             3Wick          3Widener          4Allison
##                 3                 3                 3                 4
##        4Backstrom          4Baclini           4Becker           4Carter
##                 1                 4                 4                 4
##         4Davidson             4Dean           4Herman          4Hocking
##                 1                 4                 4                 2
##        4Jacobsohn         4Johnston          4Laroche           4Renouf
##                 1                 4                 4                 1
##    4Vander Planke             4West             5Ford          5Hocking
##                 1                 4                 5                 1
##    5Kink-Heilmann          5Lefebre          5Palsson          5Ryerson
##                 1                 5                 5                 5
##          6Fortune           6Panula             6Rice         6Richards
##                 6                 6                 6                 1
##            6Skoog        7Andersson          7Asplund          8Goodwin
##                 6                 9                 7                 8
##             Small
##              1025
```

```
famIDs <- data.frame(table(combi$FamilyID))
#let's subset this dataframe to show only those unexpectedly small FamilyID groups
famIDs <- famIDs[famIDs$Freq <= 2,]

#We then need to overwrite any family IDs in our dataset for groups that were not correctly i
dentified and finally convert it to a factor:
combi$FamilyID[combi$FamilyID %in% famIDs$Var1] <- 'Small'
combi$FamilyID <- factor(combi$FamilyID)
```

Using Random Forest:

- na values
- check null values

```
apply(combi, 2, function(x) sum(is.na(x)*100/length(x)))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##  0.00000000 31.93277311  0.00000000  0.00000000  0.00000000 20.09167303
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##  0.00000000  0.00000000  0.00000000  0.07639419  0.00000000  0.00000000
##      Titles  FamilySize     Surname    FamilyID
##  0.00000000  0.00000000  0.00000000  0.00000000
```

MISSING VALUE TREATMENT

-We now also want to use the method="anova" version of our decision tree, as we are not trying to predict a category any more, but a continuous variable. So let's grow a tree on the subset of the data with the age values available, and then replace those that are missing:

```
library(rpart)
Agefit <- rpart::rpart(Age ~ Pclass + Sex + SibSp + Parch + Fare + Embarked + Titles + Family
Size,
                data=combi[!is.na(combi$Age),],
                method="anova")
combi$Age[is.na(combi$Age)] <- predict(Agefit, combi[is.na(combi$Age),])
```

recheck if missing values where replaced

```
apply(combi, 2, function(x) sum(is.na(x)*100/length(x)))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##  0.00000000 31.93277311  0.00000000  0.00000000  0.00000000  0.00000000
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##  0.00000000  0.00000000  0.00000000  0.07639419  0.00000000  0.00000000
##      Titles  FamilySize     Surname    FamilyID
##  0.00000000  0.00000000  0.00000000  0.00000000
```

```
summary(combi)
```

```
##   PassengerId      Survived          Pclass          Name
##  Min.   :   1   Min.   :0.0000   Min.   :1.000   Length:1309
##  1st Qu.: 328   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median : 655   Median :0.0000   Median :3.000   Mode  :character
##  Mean   : 655   Mean   :0.3838   Mean   :2.295
##  3rd Qu.: 982   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :1309   Max.   :1.0000   Max.   :3.000
##                 NA's   :418
##      Sex           Age            SibSp            Parch
##  female:466   Min.   : 0.17   Min.   :0.0000   Min.   :0.000
##  male  :843   1st Qu.:22.00   1st Qu.:0.0000   1st Qu.:0.000
##               Median :28.86   Median :0.0000   Median :0.000
##               Mean   :29.70   Mean   :0.4989   Mean   :0.385
##               3rd Qu.:36.50   3rd Qu.:1.0000   3rd Qu.:0.000
##               Max.   :80.00   Max.   :8.0000   Max.   :9.000
##
##       Ticket          Fare                     Cabin      Embarked
##  CA. 2343:  11   Min.   :  0.000                  :1014    :  2
##  1601    :   8   1st Qu.:  7.896   C23 C25 C27    :   6   C:270
##  CA 2144 :   8   Median : 14.454   B57 B59 B63 B66:   5   Q:123
##  3101295 :   7   Mean   : 33.295   G6             :   5   S:914
##  347077  :   7   3rd Qu.: 31.275   B96 B98        :   4
##  347082  :   7   Max.   :512.329   C22 C26        :   4
##  (Other) :1261   NA's   :1         (Other)        : 271
##      Titles      FamilySize       Surname                FamilyID
##  Mr     :757   Min.   : 1.000   Length:1309        Small     :1074
##  Miss   :260   1st Qu.: 1.000   Class :character   11Sage    :  11
##  Mrs    :197   Median : 1.000   Mode  :character   7Andersson:   9
##  Master : 61   Mean   : 1.884                      8Goodwin  :   8
##  Dr     :  8   3rd Qu.: 2.000                      7Asplund  :   7
##  Rev    :  8   Max.   :11.000                      6Fortune  :   6
##  (Other): 18                                       (Other)   : 194
```

```
summary(combi$Embarked)
```

```
##       C   Q   S
##   2 270 123 914
```

```
# Because it's so few observations and such a large majority boarded in Southampton, let's ju
st replace those two with "S".
combi$Embarked[which(combi$Embarked == '')] = "S"
combi$Embarked <- factor(combi$Embarked)

#The other naughty variable was Fare, let's take a look:
summary(combi$Fare)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   7.896  14.454  33.295  31.275 512.329       1
```

```
#It's only one passenger with a NA,so let's find out which one it is and replace it with the
 median fare:
combi$Fare[which(is.na(combi$Fare))] <- median(combi$Fare, na.rm=TRUE)
```

- Random Forests in R can only digest factors with up to 32 levels.
- Our FamilyID variable had almost double that.

```
combi$FamilyID2 <- combi$FamilyID
combi$FamilyID2 <- as.character(combi$FamilyID2)
combi$FamilyID2[combi$FamilySize <= 3] <- 'Small'
combi$FamilyID2 <- factor(combi$FamilyID2)
```

So let’s break them apart and do some predictions on our new fancy engineered variables:

```
train <- combi[1:891,]
test <- combi[892:1309,]
```

package:RANDOMFOREST

```
library(randomForest)
```

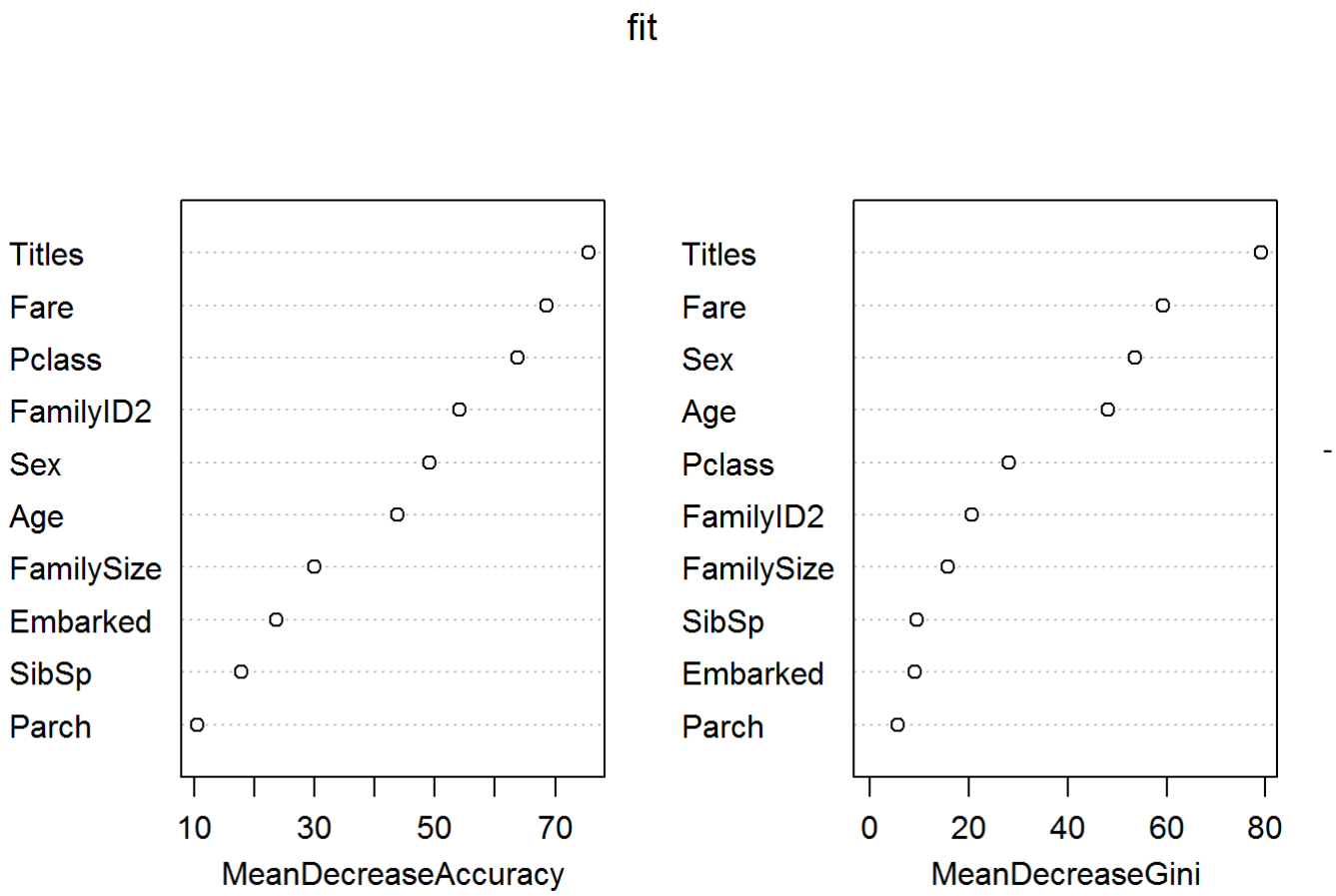
```
## randomForest 4.6-12
```

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

```
set.seed(1233)
fit <-randomForest::randomForest(as.factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch + F
are +
                      Embarked + Titles + FamilySize + FamilyID2,
                    data=train,
                    importance=TRUE,
                    ntree=2000)
```

- The importance=TRUE argument allows us to inspect variable importance as we’ll see,and the ntree argument specifies how many trees we want to grow.
- which variables where important

```
varImpPlot(fit)
```



the Gini one digs into the mathematics behind decision trees,but essentially measures how pure the nodes are at the end of the tree.Again it tests to see the result - if each variable is taken out and a high score means the variable was important.

FINAL

```
Prediction <- predict(fit, test)
submit <- data.frame(PassengerId = test$PassengerId, Survived = Prediction)
write.csv(submit, file = "Titanic_RF.csv", row.names = FALSE)
```

- 77%accuracy

CONDITIONAL INFERENCE

- There's more than one ensemble model. Let's try a forest of conditional inference trees.
- They make their decisions in slightly different ways, using a statistical test rather than a purity measure, but the basic construction of each tree is fairly similar

```
library(party)
```

```
## Warning: package 'party' was built under R version 3.4.3
```

```
## Loading required package: grid
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

```
## Loading required package: strucchange
```

```
## Warning: package 'strucchange' was built under R version 3.4.3
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
## Warning: package 'sandwich' was built under R version 3.4.3
```

```
set.seed(4415)
fit <- cforest(as.factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch + Fare +
                 Embarked + Titles + FamilySize + FamilyID,
               data = train,
               controls=cforest_unbiased(ntree=2000, mtry=3))
```

- Conditional inference trees are able to handle factors with more levels than Random Forests can,so let's go back to out original version of FamilyID

RESUBMISSION

```
Prediction <- predict(fit, test, OOB=TRUE, type = "response")
submit <- data.frame(PassengerId = test$PassengerId, Survived = Prediction)
write.csv(submit, file = "Titanic_CF.csv", row.names = FALSE)
```