Tipologia i cicle de vida de les dades: PRA2

Autor: Adem Ait; Dani Ponce

Juny 2022

Contents

Elecció del dataset **1**
Descripció de les dades . . . 2

Neteja de les dades **3**
Dades perdudes . . . 3
Discretització . . . 4
Selecció de dades . . . 5
Valors extrem . . . 6
Clean data . . . 7

Visualització de les dades **8**

Anàlisi de les dades **12**
Selecció dels grups de dades . . . 12

Classificació **14**
GLM . . . 14
Arbres de decisió . . . 16
Random Forest . . . 19

Conclusions **20**

Contribucions **20**

Elecció del dataset

Nosaltres hem escollit el dataset anomenat Titanic. Aquest dataset (com a mínim des del repositori de Kaggle del quan l'hem obtingut) es conforma per dos fitxers CSV, un destinat a l'entrenament i l'altre a l'avaluació d'un model de *machine learning*. En aquest dataset trobem els passatgers del naviu Titanic, amb diversos atributs per a cada passatger, incloent el camp Survived, que indica si va sobreviure o no. La finalitat de l'entrenament d'aquest dataset serà predir si un passatger sobreviurà o no. Per tal de netejar totes les dades de la mateixa manera, juntarem els dos arxius en una sola estructura i treballarem sobre ella. Notem que el fitxer de les dades d'avaluació conté un atribut menys (l'objectiu; *target*), per tant a la columna del *target* (`Survived`) ficarem NA's.

```
set.seed(123)

train <- read.csv("data/train.csv", stringsAsFactors = F)
test <- read.csv("data/test.csv", stringsAsFactors = F)

train$data <- "train"
```

```
test$data <- "test"

test$Survived <- NA
titanic <- rbind(train,test)
```

Descripció de les dades

Un cop carregades les dades anem a entendre que es vol aconseguir amb aquest dataset i com està compost.

Aquest dataset conté les dades sobre els passatgers del icònic transatlàntic que va enfonsar per un iceberg l'any 1912. L'objectiu del conjunt de dades és saber si donada la informació d'un passatger (edat, classe, sexe, etc.) es pot predir si sobreviurà (va sobreviure, per ser estrictament correctes) o no a l'enfonsament del Titànic.

Anem a fer una inspecció ràpida d'aquest dataset:

```
dim(titanic)
```

```
## [1] 1309   13
```

```
head(titanic, 3)
```

```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      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
##             Ticket    Fare Cabin Embarked  data
## 1        A/5 21171  7.2500              S train
## 2         PC 17599 71.2833   C85        C train
## 3 STON/O2. 3101282  7.9250              S train
```

```
str(titanic)
```

```
## 'data.frame':    1309 obs. of  13 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" ...
##  $ data       : chr  "train" "train" "train" "train" ...
```

Primer veiem que el nostre conjunt conté 1309 observacions amb 13 variables. Després podem veure una petita selecció de mostra de com és el dataset. Per acabar veiem els tipus de les variables del nostre conjunt de dades. Per tant, abans de començar amb el preprocessament de les dades, anem a crear un diccionari del conjunt de dades:

- **PassengerId** (enter): identificador del passatger

- **Survived** (enter)(*target*): si el passatger va sobreviure o no
- **Pclass** (enter): classe en la que el passatger viatjava
- **Name** (caràcter): Nom del passatger
- **Sex** (caràcter): Sexe del passatger
- **Age** (decimal): Edat del passatger
- **Sibsp** (enter): Nombre de germans/esposes a bord del Titànic
- **parch** (enter): Nombre de pares/fills a bord del Titànic
- **Ticket** (caràcter): número del ticket
- **Fare** (decimal): Preu del viatge
- **Cabin** (caràcter): Número de camarot
- **Embarked** (caràcter): Port d'embarcació: C = Cherbourg, Q = Queenstown, S = Southampton
- **data** (caràcter): dades d'entrenament o d'avaluació

Neteja de les dades

Un cop entenem que significa cada variable, anem a analitzar-les. Respecte els tipus de dades veiem que hi ha alguns tipus inusuals, com per exemple l'edat sigui decimal enlloc de tipus enter. Anem a analitzar aquesta variable:

```
sort(unique(titanic$Age))
```

```
##  [1]  0.17  0.33  0.42  0.67  0.75  0.83  0.92  1.00  2.00  3.00  4.00  5.00
## [13]  6.00  7.00  8.00  9.00 10.00 11.00 11.50 12.00 13.00 14.00 14.50 15.00
## [25] 16.00 17.00 18.00 18.50 19.00 20.00 20.50 21.00 22.00 22.50 23.00 23.50
## [37] 24.00 24.50 25.00 26.00 26.50 27.00 28.00 28.50 29.00 30.00 30.50 31.00
## [49] 32.00 32.50 33.00 34.00 34.50 35.00 36.00 36.50 37.00 38.00 38.50 39.00
## [61] 40.00 40.50 41.00 42.00 43.00 44.00 45.00 45.50 46.00 47.00 48.00 49.00
## [73] 50.00 51.00 52.00 53.00 54.00 55.00 55.50 56.00 57.00 58.00 59.00 60.00
## [85] 60.50 61.00 62.00 63.00 64.00 65.00 66.00 67.00 70.00 70.50 71.00 74.00
## [97] 76.00 80.00
```

Veiem que trobem decimals pel cas que el passatger sigui un nadó menor d'un any, o mig any per alguns adults, per tant tot sembla correcte.

Dades perdudes

Anem a veure si hi ha valors nuls o buits a les nostres dades:

```
# NAs
colSums(is.na(titanic))
```

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

Veiem que hi ha valors nuls a les columnes `Survived`, `Age` i `Fare`. Respecte la columna `Survived` els valors nuls corresponen a les files que pertanyen a les dades d'avaluació, per tant no hem de fer res. En canvi en les altres dos columnes els valors nuls no són legítims i, per tant, aplicarem una imputació per aquestes instàncies utilitzant la mediana (és més robusta als outliers que no pas la mitjana).

```
# Treat NAs with central approach (median)
titanic$Age[is.na(titanic$Age)] <- median(titanic$Age, na.rm = T)
titanic$Fare[is.na(titanic$Fare)] <- median(titanic$Fare, na.rm = T)
```

Ara anem a veure si hi ha instàncies amb valors buits.

```
colSums(titanic=="", na.rm = T)
```

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

Veiem que les columnes `Cabin` i `Embarked` tenen valors buits. En quant a la cabina, els valors són legítims i indiquen que el passatger no disposa de cabina. Per tant, crearem una nova variable per detectar els passatgers que tenen cabina i els que no.

```
titanic$hasCabin <- ifelse(titanic$Cabin != "", 1, 0)
titanic$hasCabin <- as.factor(titanic$hasCabin)
summary(titanic$hasCabin)
```

```
##    0    1
## 1014  295
```

Veiem com les 1014 instàncies que tenien valors buits representes els passatgers sense cabina i, els 295 restants els passatger amb cabina.

Respecte a la columna `Embarked`, transformarem els valors buits a `NA` per posteriorment aplicar una imputació basada en l'algorisme KNN.

```
index <- which(titanic$Embarked=="")
titanic[index,]$Embarked <- NA
titanic$Embarked <- kNN(titanic)$Embarked
titanic[index,]$Embarked
```

```
## [1] "S" "S"
```

Veiem com a les dues files se'ls hi ha assignat el port de Southampton. La imputació per KNN s'ha fet per mostrar una altra manera d'imputar dades perdudes. Hi ha altres mètodes per tractar valors nuls, com eliminar les files que continguin valors nuls, eliminar columnes que continguin un alt índex de valors nuls, imputar els valors perduts aplicant una substitució estadística (mitjana, mediana, etc), o mètodes més complexes com inferències basades en la regressió, models bayesians o arbres de decisió.

Discretització

Seguim amb el tractament de les dades factoritzant tres columnes d'especial interés (ens semblen columnes bastant relevants, o en el cas d'`Embarked` té pocs valors únics per tant és bona idea factoritzar).

```
col_factors <- c("Survived", "Sex", "Embarked")
titanic[,col_factors] <- lapply(titanic[,col_factors], as.factor)
summary(titanic[,col_factors])
```

```
##  Survived       Sex      Embarked
##  0   :549   female:466   C:270
##  1   :342   male  :843   Q:123
##  NA's:418                S:916
```

Veiem com s'han factoritzat correctament.

Addicionalment, podem discretitzar variables continues que ens puguin ser d'utilitat, com per exemple la variable `Age`.

Per discretitzar aquesta variable farem servir la funció `discretize` de la llibreria `arules`. Aquesta funció permet fer la discretització amb quatre mètodes: intervals de mateixa amplitud, intervals amb la mateixa freqüència (nombre de instàncies), fent *clustering* amb k-means i amb intervals fixats prèviament. Nosaltres hem optat pel mètode de *clustering* amb una $k = 5$, ja que tenim edats de 2 mesos fins a 80 anys. Anem a veure si amb $k = 5$ es creen uns intervals representatius (respecte a grups d'edat: adolescent, jove, adult, etc.).

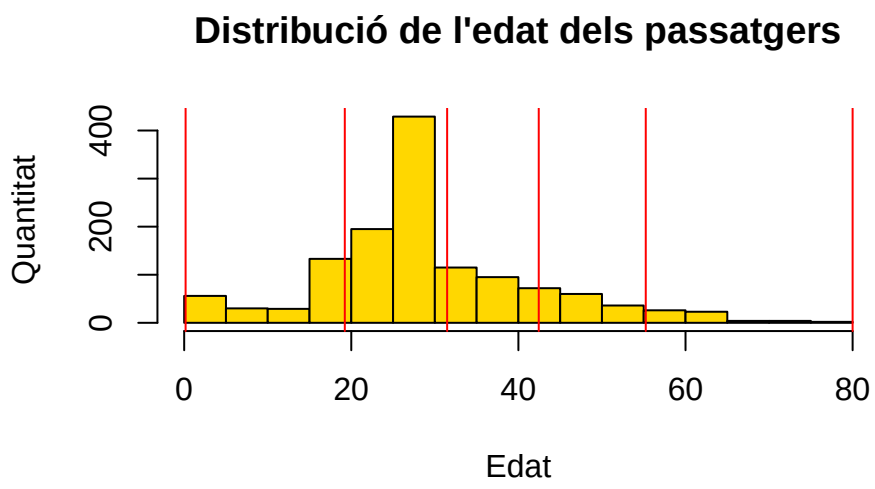
```
table(discretize(titanic$Age, "cluster", breaks = 5 ))
```

```
##
## [0.17,12.2) [12.2,23.7) [23.7,33.9) [33.9,47.8)   [47.8,80]
##          94         267         585         230         133
```

Els intervals semblen ser bastant representatius, els intervals per ordre es podrien identificar com a: nen, jove, adult, adult-gran, gran.

Visualitzem com es veuria aquesta discretització a la distribució de la variable `Age`.

```
hist(titanic$Age, main="Distribució de l'edat dels passatgers",xlab="Edat",
     ylab="Quantitat",col = "gold")
abline(v=discretize(titanic$Age, method="cluster", onlycuts=TRUE, breaks = 5),col="red")
```




```
titanic$segment_age <- discretize(titanic$Age, "cluster", breaks = 5,
                                  labels = c("nen","jove","adult","adult-gran","gran") )
```

Selecció de dades

Pensem que tots els camps poden ser necessaris per això els valorarem tots. Però som conscients que hi ha camps que té pinta que siguin més significants que altres. Per exemple, el camp `Name`, no té pinta que tingui relevància, però podem extreure el títol del passatger. Per tant, substituïrem la columna `Name`, per un nou camp calculat.

Podem extreure noves variables de columnes existents (com hem fet prèviament amb `hasCabin`). Com acabem de mencionar se'ns ofereix informació sobre el nom de cada passatger, la qual cosa no sembla rellevant, però d'aquí podem extreure el títol de cada individu (Mr, Mrs, etc.). Procedim a extreure'l i esborrar el camp `Name`.

```
titanic$title <- gsub('(.*, )|(\\..*)', '', titanic$Name)
table(titanic$title)
```

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

```
titanic$Name <- NULL
```

Veiem que els títols més repetits són *Miss*, *Mrs*, *Mr* i *Master*. Però addicionalment tenim altres valors com: *Ms* que és el mateix que *Miss*, *Dona* i *Lady* que són sinònims de *Mrs*, i *Don* i *Sir* que són sinònims de *Mr*. Aquests sinònims són títols nobiliaris, entre altres, i com només ens interessa mostrar la creació de nous atributs a partir d'existents i no fer un anàlisi exhaustiu sobre aquest aspecte ens quedarem amb els títols de *Mrs*, *Miss*, *Mr* i *Master*. La resta de títols els catalogarem com `Special`.

```
titanic$title[titanic$title == "Dona"] <- "Mrs"
titanic$title[titanic$title == "Lady"] <- "Mrs"
titanic$title[titanic$title == "Ms"] <- "Miss"
titanic$title[titanic$title == "Don"] <- "Mr"
titanic$title[titanic$title == "Sir"] <- "Mr"

residual <- c("Capt", "Col", "Dr", "Jonkheer", "Major", "Mile", "Mme", "Rev",
              "the Countess", "Mlle")
titanic$title[titanic$title %in% residual] <- "Special"
titanic$title <- as.factor(titanic$title)
summary(titanic$title)
```

```
##  Master    Miss      Mr     Mrs Special
##      61     262     759     199      28
```

Un altre atribut que podem extreure és la mida de la família (contant-se el propi passatger) que viatja conjuntament al Titanic

```
titanic$FamilySize <- titanic$SibSp + titanic$Parch +1
summary(titanic$FamilySize)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   1.000   1.884   2.000  11.000
```

Veiem com la majora viatgen sols (la mida de la família és 1),

Valors extrem

Ara anem a tractar els valors extrem (*outliers*).

```
# Agafem les variables numèriques
numCols <- c("Age", "SibSp", "Parch","Fare")
sapply(colnames(titanic[,numCols]), function(x) boxplot.stats(titanic[,x])$out)
```

```
## $Age
##   [1]  2.00 58.00 55.00  2.00 66.00 65.00  0.83 59.00 71.00 70.50  2.00 55.50
##  [13]  1.00 61.00  1.00 56.00  1.00 58.00  2.00 59.00 62.00 58.00 63.00 65.00
##  [25]  2.00  0.92 61.00  2.00 60.00  1.00  1.00 64.00 65.00 56.00  0.75  2.00
##  [37] 63.00 58.00 55.00 71.00  2.00 64.00 62.00 62.00 60.00 61.00 57.00 80.00
##  [49]  2.00  0.75 56.00 58.00 70.00 60.00 60.00 70.00  0.67 57.00  1.00  0.42
##  [61]  2.00  1.00 62.00  0.83 74.00 56.00 62.00 63.00 55.00 60.00 60.00 55.00
```

```
##  [73] 67.00  2.00 76.00 63.00  1.00 61.00 60.50 64.00 61.00  0.33 60.00 57.00
##  [85] 64.00 55.00  0.92  1.00  0.75  2.00  1.00 64.00  0.83 55.00 55.00 57.00
##  [97] 58.00  0.17 59.00 55.00 57.00
##
## $SibSp
##  [1] 3 4 3 3 4 5 3 4 5 3 3 4 8 4 4 3 8 4 8 3 4 4 4 4 8 3 3 5 3 5 3 4 4 3 3 5 4 3
## [39] 4 8 4 3 4 8 4 8 3 4 5 3 4 8 4 8 4 3 3
##
## $Parch
##   [1] 1 2 1 5 1 1 5 2 2 1 1 2 2 2 1 2 2 2 3 2 2 1 1 1 1 2 1 1 2 2 1 2 2 2 1 2 1
##  [38] 1 2 1 4 1 1 1 1 2 2 1 2 1 1 1 2 1 1 2 2 2 1 1 2 2 1 2 1 1 1 1 1 1 1 2 1 2
##  [75] 2 1 1 2 1 1 2 1 1 1 1 2 1 1 1 4 1 1 2 2 2 2 2 1 1 1 2 2 1 1 2 2 3 4 1 2 1
## [112] 1 2 1 2 1 2 1 1 2 2 1 1 1 1 2 2 2 2 2 2 1 1 2 1 4 1 1 2 1 2 1 1 2 5 2 1 1
## [149] 1 2 1 5 2 1 1 1 2 1 6 1 2 1 2 1 1 1 1 1 1 1 3 2 1 1 1 1 2 1 2 3 1 2 1 2 2
## [186] 1 1 2 1 2 1 2 1 1 1 2 1 1 2 1 2 1 1 1 1 3 2 1 1 1 1 5 2 1 1 1 1 3 1 2 2 1
## [223] 2 1 2 1 2 4 1 1 2 1 1 1 4 6 2 3 1 1 2 2 2 1 1 2 5 2 3 2 1 1 1 2 1 2 2 2 1
## [260] 2 1 1 2 1 2 1 2 1 2 2 1 1 1 1 1 2 1 1 2 1 1 1 2 1 2 9 1 1 1 2 2 2 1 9 1 1
## [297] 2 2 1 1 2 1 1 1 1 1 1
##
## $Fare
##   [1]  71.2833 263.0000 146.5208  82.1708  76.7292  80.0000  83.4750  73.5000
##   [9] 263.0000  77.2875 247.5208  73.5000  77.2875  79.2000  66.6000  69.5500
##  [17]  69.5500 146.5208  69.5500 113.2750  76.2917  90.0000  83.4750  90.0000
##  [25]  79.2000  86.5000 512.3292  79.6500 153.4625 135.6333  77.9583  78.8500
##  [33]  91.0792 151.5500 247.5208 151.5500 110.8833 108.9000  83.1583 262.3750
##  [41] 164.8667 134.5000  69.5500 135.6333 153.4625 133.6500  66.6000 134.5000
##  [49] 263.0000  75.2500  69.3000 135.6333  82.1708 211.5000 227.5250  73.5000
##  [57] 120.0000 113.2750  90.0000 120.0000 263.0000  81.8583  89.1042  91.0792
##  [65]  90.0000  78.2667 151.5500  86.5000 108.9000  93.5000 221.7792 106.4250
##  [73]  71.0000 106.4250 110.8833 227.5250  79.6500 110.8833  79.6500  79.2000
##  [81]  78.2667 153.4625  77.9583  69.3000  76.7292  73.5000 113.2750 133.6500
##  [89]  73.5000 512.3292  76.7292 211.3375 110.8833 227.5250 151.5500 227.5250
##  [97] 211.3375 512.3292  78.8500 262.3750  71.0000  86.5000 120.0000  77.9583
## [105] 211.3375  79.2000  69.5500 120.0000  93.5000  80.0000  83.1583  69.5500
## [113]  89.1042 164.8667  69.5500  83.1583  82.2667 262.3750  76.2917 263.0000
## [121] 262.3750 262.3750 263.0000 211.5000 211.5000 221.7792  78.8500 221.7792
## [129]  75.2417 151.5500 262.3750  83.1583 221.7792  83.1583  83.1583 247.5208
## [137]  69.5500 134.5000 227.5250  73.5000 164.8667 211.5000  71.2833  75.2500
## [145] 106.4250 134.5000 136.7792  75.2417 136.7792  82.2667  81.8583 151.5500
## [153]  93.5000 135.6333 146.5208 211.3375  79.2000  69.5500 512.3292  73.5000
## [161]  69.5500  69.5500 134.5000  81.8583 262.3750  93.5000  79.2000 164.8667
## [169] 211.5000  90.0000 108.9000
```

Com podem veure, cap valor extrem sembla ser erroni ni molt allunyat del conjunt de valors. Per exemple, la columna `Fare`, té algun valor allunyat (512.3292) que es pot correspondre per un bitllet de primera classe [Ref]. Per tant, no cal fer cap tractament d'aquests *outliers*.

Clean data

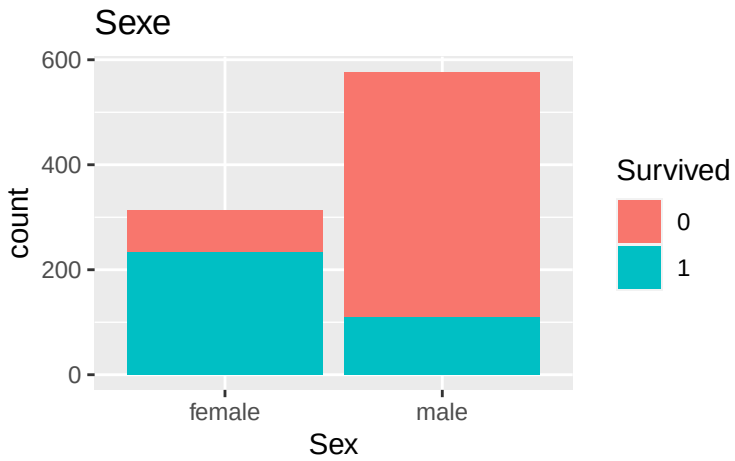
Passem a guardar les dades netejades:

```
write.csv(titanic, "data/titanic_clean.csv", row.names = F)
```

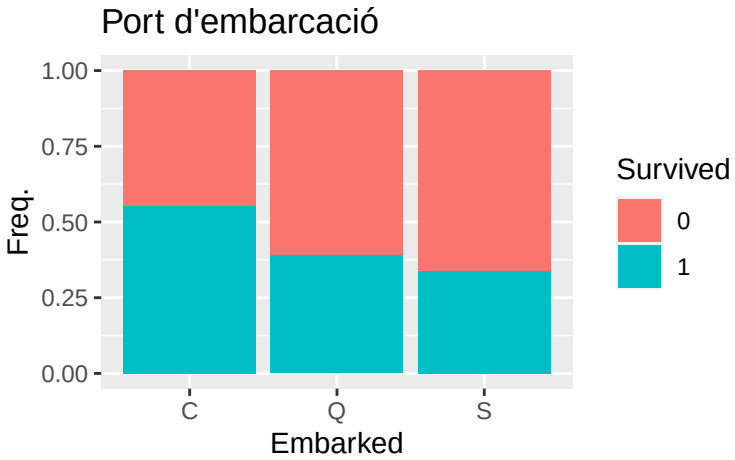
Visualització de les dades

Procedim doncs a visualitzar les distribucions d'algunes de les columnes que ens poden explicar millor el factor clau per la supervivència en l'enfonsament del Titànic. Aquestes són: `Pclass`, `Sex`, `segment_age`, `Sibsp`, `Embarked` i `FamilySize`. Això no significa que sigui les úniques variables que farem servir.

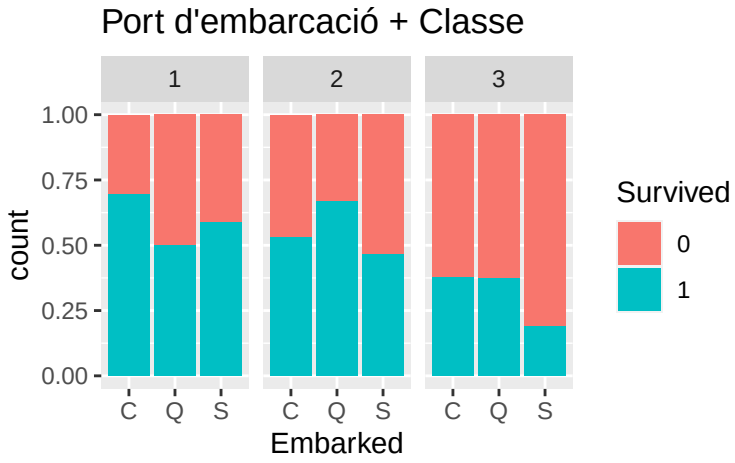
```
train_data <- titanic[titanic$data == "train",]
ggplot(data= train_data,aes(x=Sex,fill=Survived)) + geom_bar() +  ggtitle("Sexe")
```




```
ggplot(data = train_data,aes(x=Embarked,fill=Survived)) +
  geom_bar(position="fill") +  ylab("Freq.") + ggtitle("Port d'embarcació")
```

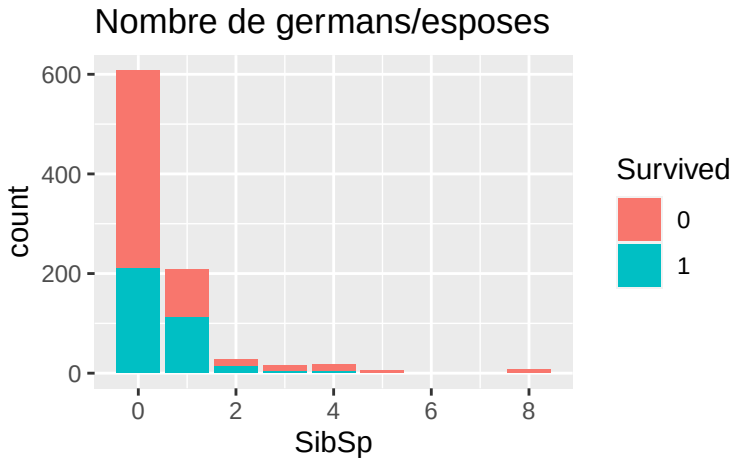



```
ggplot(data = train_data,aes(x=Embarked,fill=Survived))+
  geom_bar(position="fill") +  facet_wrap(~Pclass) +  ggtitle("Port d'embarcació + Classe")
```

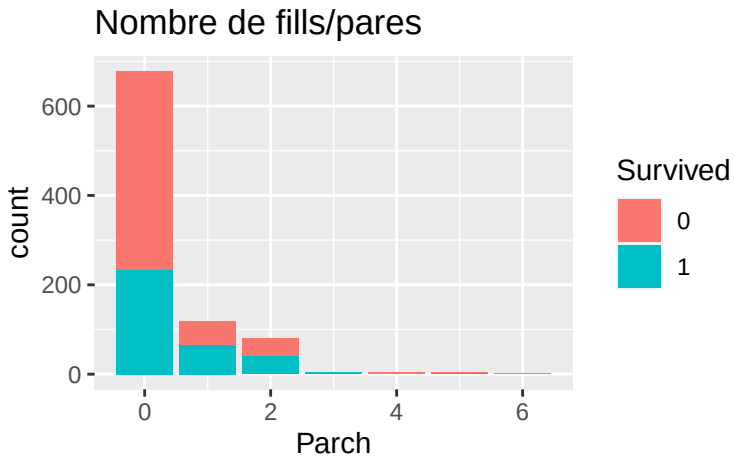



```
ggplot(data = train_data,aes(x=Pclass,fill=Survived))+  geom_bar(position="fill") +
  facet_wrap(~Sex) +  ggtitle("Classe + Sexe")
```


```
ggplot(data = train_data,aes(x=SibSp,fill=Survived)) +
  geom_bar() +  ggtitle("Nombre de germans/esposes")
```

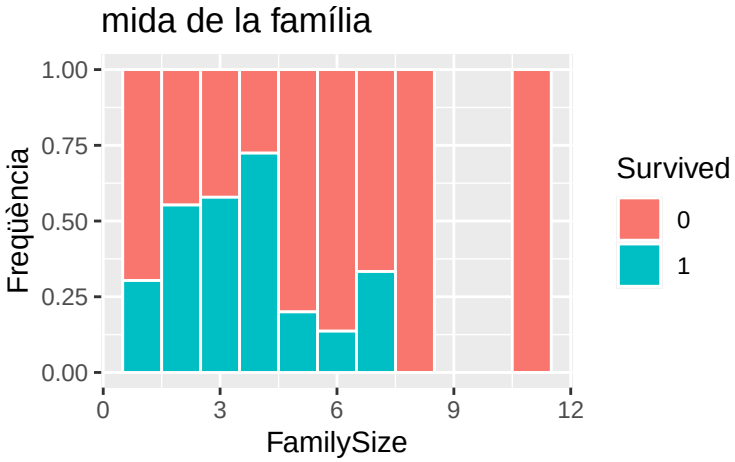


```
ggplot(data = train_data,aes(x=Parch,fill=Survived)) +
  geom_bar() +  ggtitle("Nombre de fills/pares")
```

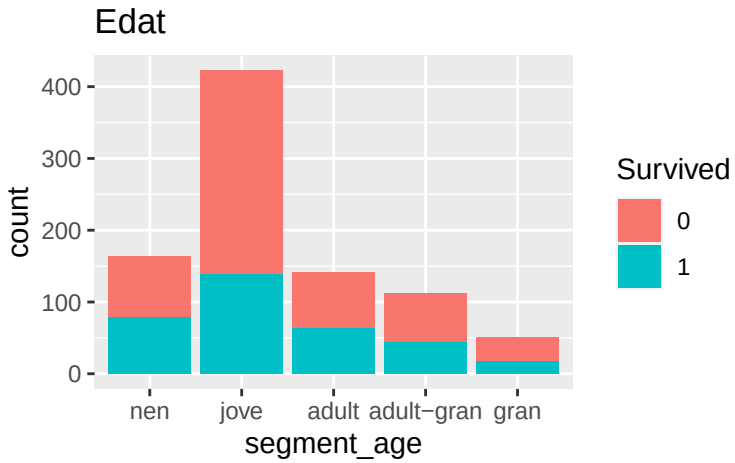



```
ggplot(data = train_data, aes(x=FamilySize,fill=Survived)) +
  geom_histogram(binwidth =1,position="fill", color = "white") +
  ylab("Freqüència")+  ggtitle("mida de la família")
```

```
## Warning: Removed 4 rows containing missing values (geom_bar).
```




```
ggplot(data= train_data,aes(x=segment_age,fill=Survived)) +  geom_bar() +  ggtitle("Edat")
```



A simple vista podem deduir que aquestes variables seran bones candidates a ser variables explicatives per al nostre model, ja que s'observen diferències significatives entre les categories de la variable respecte si van sobreviure o no.

Anem a veure un exemple per a avaluar la relació existent entre una variable explicativa i la nostra variable target. Prenem la variable `Sex`. Utilitzarem el test de Fisher per a avaluar si existeix associació entre la variable sexe i el fet de sobreviure. Plantegem les hipòtesis següents:

H_0: Les variables són independents, ie, els valors de sexe no influeixen en els valors que pren la variable `Survived`

H_1: Les variables són dependents, ie, els valors que pren la variable sexe tenen relació amb els valors que pren la variable `Survived`

```
tt <- table(titanic$Survived[titanic$data=="train"], titanic$Sex[titanic$data=="train"],
            dnn = c("Sobreviu", "Sexe"))
fisher.test(x = tt, alternative = "two.sided")
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  tt
## p-value < 2.2e-16
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.0575310 0.1138011
## sample estimates:
## odds ratio
## 0.08128333
```

Observem que el p-valor inferior a 0.05 ens permet rebutjar la hipòtesi nul · la i dir que tenim relació entre la variable sexe i el fet de sobreviure. Anem a avaluar la contundència d'aquesta relació analitzant la força d'associació:

```
assocstats(x = tt)
```

```
##                     X^2 df P(> X^2)
## Likelihood Ratio 268.85  1        0
## Pearson          263.05  1        0
##
## Phi-Coefficient   : 0.543
## Contingency Coeff.: 0.477
```

```
## Cramer's V         : 0.543
```

Un coeficient V de Cramer a partir de 0.5 es considera (per convenció) una força d'associació gran (tamany de l'efecte).

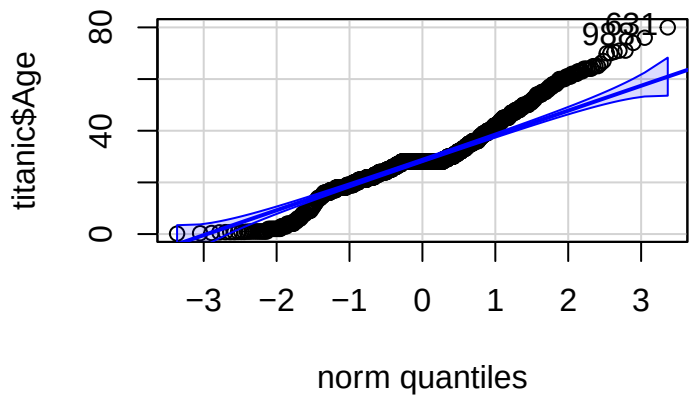
Anàlisi de les dades

Selecció dels grups de dades

Anem a comprovar la normalitat i homogeneïtat de la variància.

Normalitat

```
qqPlot(titanic$Age)
```

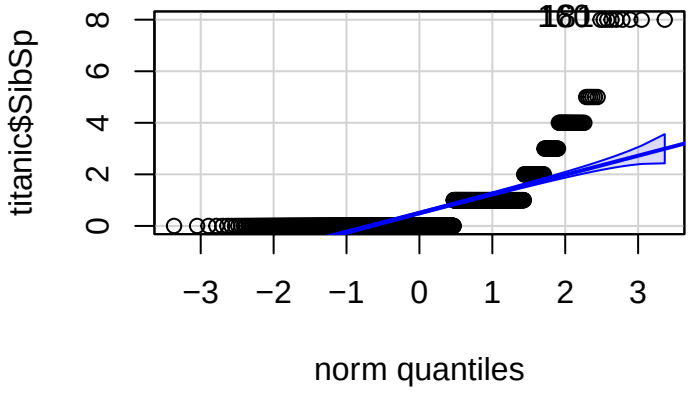



```
## [1] 631 988
```

```
shapiro.test(titanic$Age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  titanic$Age
## W = 0.95107, p-value < 2.2e-16
```

```
qqPlot(titanic$SibSp)
```

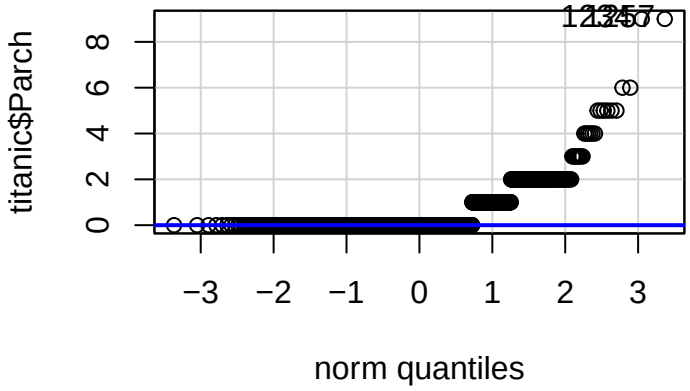



```
## [1] 160 181
```

```
shapiro.test(titanic$SibSp)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  titanic$SibSp
## W = 0.51108, p-value < 2.2e-16
```

```
qqPlot(titanic$Parch)
```



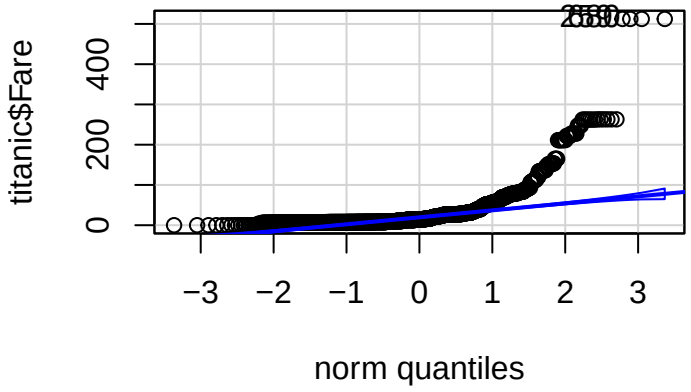

```
## [1] 1234 1257
```

```
shapiro.test(titanic$Parch)
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  titanic$Parch
## W = 0.49797, p-value < 2.2e-16
```

```
qqPlot(titanic$Fare)
```




```
## [1] 259 680
```

```
shapiro.test(titanic$Fare)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  titanic$Fare
## W = 0.52766, p-value < 2.2e-16
```

Com veiem cap nivell de significança és major que 0.05, ni les dades estan dins de l'interval de confiança. Per tant, cap d'aquestes variables presenta normalitat.

Homogeneitat de la variància

Com sabem, molts testos estadístics assumeixen la homogeneitat de la variància per a dur a terme contrastos d'hipòtesis. Això és, que en incrementar el valor una variable explicativa, la nostra variable dependent manté la variància constant. En el nostre cas però, estem davant d'una variable target categòrica amb dues classes (sobreviu, no sobreviu) i doncs, no té massa rellevància aquest estudi.

Classificació

GLM

Anem a construir un model de regressió logística per predir si un passatger sobreviu o no. Construirem un model lineal generalitzat (GLM) amb la familia binomial.

```
model.glm <- glm(Survived ~ Sex + Pclass + Age + Fare +
                   Embarked + segment_age + title + hasCabin + FamilySize,
                 data = titanic[titanic$data == "train",],
                 family = binomial)
summary(model.glm)
```

```
##
## Call:
## glm(formula = Survived ~ Sex + Pclass + Age + Fare + Embarked +
##     segment_age + title + hasCabin + FamilySize, family = binomial,
##     data = titanic[titanic$data == "train", ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4053  -0.5604  -0.3785   0.5639   2.5351
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            20.706497 636.735157   0.033   0.9741
## Sexmale               -15.407574 636.734630  -0.024   0.9807
## Pclass                 -0.848570   0.183283  -4.630 3.66e-06 ***
## Age                    -0.036722   0.025597  -1.435   0.1514
## Fare                    0.002569   0.002687   0.956   0.3391
## EmbarkedQ              -0.150064   0.400352  -0.375   0.7078
## EmbarkedS              -0.482204   0.248811  -1.938   0.0526 .
## segment_agejove         0.093459   0.438541   0.213   0.8312
## segment_ageadult        0.771794   0.639294   1.207   0.2273
## segment_ageadult-gran   0.433993   0.896136   0.484   0.6282
## segment_agegran         0.325871   1.273105   0.256   0.7980
## titleMiss             -15.861150 636.734818  -0.025   0.9801
## titleMr                -3.484985   0.557987  -6.246 4.22e-10 ***
## titleMrs              -15.096286 636.734884  -0.024   0.9811
## titleSpecial           -3.504830   0.840156  -4.172 3.02e-05 ***
## hasCabin1               0.638994   0.315904   2.023   0.0431 *
## FamilySize             -0.471819   0.087808  -5.373 7.73e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  717.55  on 874  degrees of freedom
## AIC: 751.55
##
## Number of Fisher Scoring iterations: 14
```

Veiem que les variables que expliquen la major part del factor de supervivència en el nostre model són: `Pclass`, `title`, `FamilySize` i `hasCabin`.

Anem a predir la supervivència dels passatgers

```
titanic$pred <- predict(model.glm, titanic)
```

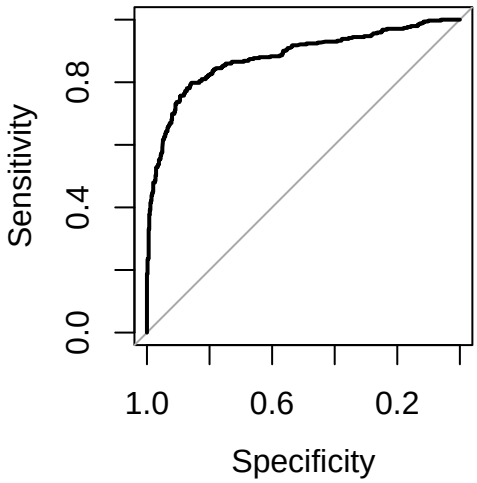
Per tal de veure com de bé s'ajusta aquest model es pot utilitzar la corba ROC, que representa la relació entre la ratio de vertaders positius (TPR) i de falsos positius (FPR):

```
roc <- roc(titanic$Survived, titanic$pred)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
plot(roc)
```



Veiem que es la corba es troba molt per sobre de la línia diagonal i doncs deduim que tenim un bon classificador. El classificador òptim és aquell on la corba creix en perpendicular a l'eix d'abcisses fins a y=1 i després es manté en l'ordenada 1 fins a x=1.

Amb les probabilitats trobades, assignem el valor corresponent a la variable target.

```
titanic$predSurvived <- ifelse(titanic$pred < 0.5, 0, 1)
```

Obtenim així la predicció del conjunt de test:

```
output <- titanic[titanic$data == "test", c("PassengerId", "predSurvived")]
colnames(output) <- c("PassengerId", "Survived")
```

Arbres de decisió

Els arbres de decisió són un dels models supervisats de classificació que s'usen més en problemes de mineria de dades. La raó principal és perquè tenen una alta capacitat explicativa i perquè és molt fàcil interpretar el model que se n'obté.

Així doncs és ideal per aplicar-lo en el nostre cas. Utilitzarem la funció `c5.0` de la llibreria `c50`.

Selecciones les columnes d'interés.

```
interest_cols = c("Pclass", "Sex", "Age", "SibSp", "Parch", "Fare", "Embarked", "title",
                  "segment_age", "FamilySize", "hasCabin")
```

Amb aquestes columnes crearem el model i l'entrenarem. Amb la funció `summary` i l'atribut `rules=TRUE` podrem analitzar a fons l'arbre de decisió creat.

```
model <- C50::C5.0(titanic[titanic$data == "train",interest_cols],
                   titanic$Survived[titanic$data == "train"],
                   rules=TRUE )
summary(model)
```

```
##
## Call:
## C5.0.default(x = titanic[titanic$data == "train", interest_cols], y
```

```
##  = titanic$Survived[titanic$data == "train"], rules = TRUE)
##
##
## C5.0 [Release 2.07 GPL Edition]      Sun Jun 05 11:45:08 2022
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 891 cases (12 attributes) from undefined.data
##
## Rules:
##
## Rule 1: (8, lift 1.5)
##  Parch > 0
##  Embarked = Q
##  ->  class 0  [0.900]
##
## Rule 2: (537/86, lift 1.4)
##  Sex = male
##  title in {Mr, Special}
##  ->  class 0  [0.839]
##
## Rule 3: (491/119, lift 1.2)
##  Pclass > 2
##  ->  class 0  [0.757]
##
## Rule 4: (44, lift 2.5)
##  Sex = female
##  Fare > 15.2458
##  Embarked = C
##  segment_age in {nen, jove, adult, gran}
##  ->  class 1  [0.978]
##
## Rule 5: (22, lift 2.5)
##  SibSp <= 2
##  title = Master
##  ->  class 1  [0.958]
##
## Rule 6: (170/9, lift 2.5)
##  Pclass <= 2
##  Sex = female
##  ->  class 1  [0.942]
##
## Rule 7: (9, lift 2.4)
##  Sex = female
##  Fare <= 13.8625
##  Embarked = C
##  ->  class 1  [0.909]
##
## Rule 8: (93/14, lift 2.2)
##  Sex = female
##  title in {Mrs, Special}
##  segment_age in {nen, jove, adult, gran}
##  FamilySize <= 4
```

```
##  ->  class 1  [0.842]
##
## Rule 9: (33/6, lift 2.1)
##  Embarked = S
##  title = Miss
##  segment_age = nen
##  FamilySize <= 4
##  ->  class 1  [0.800]
##
## Rule 10: (33/6, lift 2.1)
##  Sex = female
##  Parch <= 0
##  Embarked = Q
##  ->  class 1  [0.800]
##
## Default class: 0
##
##
## Evaluation on training data (891 cases):
##
##          Rules
##    ----------------
##      No      Errors
##
##      10  128(14.4%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     520    29    (a): class 0
##      99   243    (b): class 1
##
##
##  Attribute usage:
##
##   86.98% Sex
##   76.88% title
##   74.19% Pclass
##   16.84% segment_age
##   14.25% Embarked
##   14.14% FamilySize
##    5.95% Fare
##    4.60% Parch
##    2.47% SibSp
##
##
## Time: 0.0 secs
```

Primer de tot analitzem la taxa d'error: `Errors` mostra el número i percentatge de casos mal classificats en el subconjunt d'entrenament. L'arbre obtingut classifica erròniament 110 dels 891 casos donats, una taxa d'error del 12,3%.

En total tenim 19 regles. Les regles estan numerades i estan acompanyades de dos valors `(n, lift x)` on n a vegades està de la forma n/m. n és el nombre de casos d'entrenament tractats per la regla i m, si hi és, indica quants no pertanyen a la classe que prediu la regla. La precisió de la regla és estimada pel ratio de

Laplace $(n-m+1)/(n+2)$. I `lift x` és el resultat de dividir la precisió estimada per la freqüència relativa de la classe predita en el conjunt d'entrenament. També podem veure les condicions que s'han de satisfer, la classe que prediu la regla i la confidència amb la que la regla prediu la classe.

Com a exemple analitzarem la primera regla a fons. Aquesta regla tracta 41 observacions i prediu amb una confidència del 9,7% que 38 dels 41 passatgers que eren homes i van embarcar al port de Queenston no van sobreviure (`class 0`).

Si ens hi fixem en varies regles veiem com un dels trets més distintius és el sexe. Les regles que classifiquen passatgers com a no supervivents, com per exemple la regla 1, 3, 4 o 6, tenen un distintiu masculí, ja sigui pel títol o pel sexe mateix. I les que classifiquen passatgers com a supervivents solen valorar el fet que el passatger sigui una dona, per exemple les regles 7, 10, 11, 12 o 15.

Anem a veure la precisió de l'arbre a partir de la matriu de contingència.

```
titanic$dtSurvived <- NA
predicted_model <- predict( model,
                            titanic[titanic$data=='test', interest_cols],
                            type="class" )
titanic[titanic$data=="test",]$dtSurvived <- predicted_model
head(titanic[titanic$data=="test",]$dtSurvived, 10)
```

```
##  [1] 1 1 1 1 2 1 2 1 2 1
```

Hem predit la supervivència dels passatgers de les dades d'avaluació. Podem extreure el percentatge de passatgers que van sobreviure amb:

```
surv <- titanic[titanic$data=="test",]$dtSurvived

(length(surv[surv==2]))/(length(surv))
```

```
## [1] 0.3516746
```

Un 34,69% dels passatgers no van sobreviure.

Random Forest

Un altre algorisme que funciona en base als arbres de decisió és el *random forest*. Aquest genera una multitud d'arbres de decisió i, en el cas dels projectes de classificació, selecciona la classe que més arbres de decisió han seleccionat.

```
interest_cols = c("Pclass", "Sex", "Age", "SibSp", "Parch", "Fare", "Embarked", "title",
                  "segment_age", "FamilySize", "hasCabin")

classifier <- randomForest(x = titanic[titanic$data == "train",interest_cols],
                           y = titanic$Survived[titanic$data == "train"],
                           ntree = 500, random_state = 0)

titanic$rfPred <- NA
pred <- predict(classifier,
                newdata = titanic[titanic$data == "test", interest_cols])
titanic[titanic$data == "test",]$rfPred <- pred
```

Amb la predicció usant el random forest tenim una taxa de supervivència en els passatgers del conjunt de dades d'avaluació del:

```
surv <- titanic[titanic$data == "test",]$rfPred
(length(surv[surv==1]))/(length(surv))
```

```
## [1] 0.6483254
```

La taxa de supervivència és del 35,41%, hi ha una diferència de 0,72 punts amb la predicció de l'arbre de decisió.

Hem vist diferents mètodes per fer la classificació, i si haguéssim d'escollir un mètode per respondre a la pregunta de quins passatgers sobreviuen en el conjunt de dades d'avaluació, escolliríem el mètode random forest, degut al seu potencial i al fet de que utilitza 500 arbres de decisió per fer la classificació, essent així més fiable que no pas un sol arbre de decisió.

Per tant adjudiquem aquests valors al camp `Survived`:

```
titanic[titanic$data == "test",]$Survived <- as.factor(
  titanic[titanic$data == "test",]$rfPred-1) # els valors estan en 1 i 2, enlloc de 0 i 1
```

Conclusions

Hem vist com atacar el problema dels valors mancants amb les dades amb dues tècniques diferents (utitzant la mediana i amb el mètode KNN). A més, hem analitzat els valors extrems i dut a terme un anàlisi de la distribució de les dades i la relació que tenen amb la nostra variable target (`Survived`). Aquest anàlisi l'hem realitzat tant visualment com amb una proba estadística.

Per altra banda, hem pogut veure com poder extreure informació de dades que inicialment no era possible tractar. Per exemple, el cas del nom del passatger podria ser fàcilment ignorat, però hem constatat que contenia informació rellevant (títol o forma de tractament) i significativa per al nostre model.

Hem creat 3 models classificadors per a predir si un passatger sobreviu o no. En primer lloc hem realitzat un model de regressió logística, seguit d'un arbre de decisió i acabant amb un model random forest.

Contribucions

Aquest estudi ha estat realitzat conjuntament per l'Adem Ait (AA) i el Dani Ponce (DP):

- **Investigació prèvia**: AA, DP
- **Redacció de les respostes**: AA, DP
- **Desenvolupament codi**: AA, DP