PRA2 TCVD

MU Ciència de Dades - 2017-18

Jorge Marsal Poy

Contingut

PRAC1

Descripció de la Pràctica a realitzar

L'objectiu d'aquesta activitat serà el tractament d'un dataset, que pot ser el creat a la pràctica 1 o bé qualsevol dataset lliure disponible a Kaggle (https://www.kaggle.com). Alguns exemples de dataset amb els que podeu treballar són:

- Red Wine Quality (https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009).
- Titanic: Machine Learning from Disaster (https://www.kaggle.com/c/titanic).
- Predict Future Sales (https://www.kaggle.com/c/competitive-data-sciencepredict-future-sales/).

Els últims dos exemples corresponen a competicions actives a Kaggle de manera que, opcionalment, podrieu aprofitar el treball realitzat durant la pràctica per entrar en alguna d'aquestes competicions.

Seguint les principals etapes d'un projecte analític, les diferents tasques a realitzar (i justificar) són les següents:

1. Descripció del dataset. Perquè és important i quina pregunta/problema pretén respondre?
2. Integració i selecció de les dades d'interès a analitzar.
3. Neteja de les dades.
    - 3.1. Les dades contenen zeros o elements buits? Com gestionaries aquests casos?
    - 3.2. Identificació i tractament de valors extrems.
4. Anàlisi de les dades.
    - 4.1. Selecció dels grups de dades que es volen analitzar/comparar (planificació dels anàlisis a aplicar).
    - 4.2. Comprovació de la normalitat i homogeneïtat de la variància.
    - 4.3. Aplicació de proves estadístiques per comparar els grups de dades. En funció de les dades i de l'objectiu de l'estudi, aplicar proves de contrast d'hipòtesis, correlacions, regressions, etc.
5. Representació dels resultats a partir de taules i gràfiques.
6. Resolució del problema. A partir dels resultats obtinguts, quines són les conclusions? Els resultats permeten respondre al problema?
7. Codi: Cal adjuntar el codi, preferiblement en R, amb el que s'ha realitzat la neteja, anàlisi i representació de les dades. Si ho preferiu, també podeu treballar en Python.

TCVD-PRA2

Aquesta pràctica s'ha efectuat a títol individual pel Jordi Marsal Poy.

Enllaç a Github:

https://github.com/jordimarsal/TCVD-PRA2

origen csv:

https://www.kaggle.com/c/titanic

1. Descripció del dataset

El conjunt de dades és un dels proposats a l'enunciat de la pràctica. Descarregat de kaggle.com. està compost per 3 arxius, l'arxiu d'entrenament pel model, ***train.csv***, l'arxiu per testejar el model, ***test.csv***. i finalment un arxiu per comprovar la precisió del model, ***gender_submission.csv***, amb els resultats correctes per comparar-los amb l'estimació que predigui el model.

train.csv conté 12 columnes amb 891 files. ***Test.csv*** conté una columna menys, la columna ***Survived***, que és la que el model intentarà predir. Les columnes del dataset són:

PassengerId: número d'identificació únic per cada passatger.

Survived: valor 1, el passatger va sobreviure, valor 0 el passatger va morir.

Pclass: el passatger anava en 1ª, 2ª o 3ª classe.

Name: Nom i cognoms del passatger.

Sex: Masculí o femení.

Age: edat del passatger.

SibSp: número de germans o cònjuges del passatger.

Parch: número de pares o fills del passatger.

Ticket: número del ticket.

Fare: preu del passatge.

Cabin: cabina on estava allotjat.

Embarked: port des del que va embarcar. C = Cherbourg, Q = Queenstown, S = Southampton.

Importància i que pretén respondre:

El dataset conté dades corresponents al passatge que va abordar el vaixell Titànic en el seu viatge inaugural. Com és per tothom sabut, el vaixell va naufragar i una important part del passatge va morir tràgicament. El dataset és utilitzat per construir models predictius i regles per les quals un subconjunt de les dades poden ser inferides. Concretament s'intenta esbrinar la columna de supervivents. Per aconseguir-ho es seleccionaran les columnes que puguin aportar informació mes útil i a partir d'aquestes es generarà un model.

2. Integració i selecció de les dades d'interès a analitzar.

Per seleccionar quines columnes aporten més informació, donem un cop d'ull al dataset:

```
> 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 86 396 345 133 ...
 $ 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 ...
```

També imprimim unes quantes línies de mostra:

```
> idx <- sample(1:nrow( train ),5)
> train [idx,] # filter 5 rows of data
    PassengerId Survived Pclass                                Name    Sex Age SibSp Parch  Ticket  Fare Cabin Embarked
260         260        1      2     Parrish, Mrs. (Lutie Davis) female  50     0     1  230433 26.00              S
505         505        1      1           Maioni, Miss. Roberta female  16     0     0  110152 86.50   B79        S
891         891        0      3             Dooley, Mr. Patrick   male  32     0     0  370376  7.75              Q
387         387        0      3 Goodwin, Master. Sidney Leonard   male   1     5     2 CA 2144 46.90              S
345         345        0      2         Fox, Mr. Stanley Hubert   male  36     0     0  229236 13.00              S
```

Ja des d'un començament observem que hi ha camps descriptius que no aporten informació per la creació del model. Abans de decidir-nos del tot, donem un cop d'ull a la qualitat de les dades (veure punt 3: Neteja de dades)

Llavors ja tenim informació per decidir-nos, les columnes amb valors nuls són Cabin i Age. Però Age podria tenir significat en el model i la reservem per observar si funciona en el model. La PassengerId només descriu una fila, el Name també descriu una persona, el Ticket diu amb quin títol de transport viatjava i potser que anava amb algú més. Però aquesta informació la aporta amb més exactitud SibSp i Parch. Fare ens diu el preu del bitllet, però per agrupar ja tenim un mode natural ben definit que és la Pclass en la que viatjava.

Alterarem el dataset per reduir-lo amb camps més definitoris:

```
> train_r <- train [,c("Survived","Pclass","Sex","Age","SibSp","Parch","Embarked")]
> test_r <- test [,c("Pclass","Sex","Age","SibSp","Parch","Embarked")]

> idx <- sample(1:nrow( train_r ),5)
> train_r [idx,] # filter 5 rows of data
    Survived Pclass    Sex Age SibSp Parch Embarked
405        0      3 female  20     0     0        S
860        0      3   male  NA     0     0        C
128        1      3   male  24     0     0        S
754        0      3   male  23     0     0        S
14         0      3   male  39     1     5        S
```

3. Neteja de les dades:

```
> sapply( train , function(x) (sum(is.na(x))))
PassengerId    Survived      Pclass        Name         Sex         Age       SibSp       Parch      Ticket        Fare       Cabin    Embarked
          0           0           0           0           0         177           0           0           0           0           0           0

> sapply( train , function(x) (sum(x==""))})
PassengerId    Survived      Pclass        Name         Sex         Age       SibSp       Parch      Ticket        Fare       Cabin    Embarked
          0           0           0           0           0          NA           0           0           0           0         687           2

> sapply( test , function(x) (sum(is.na(x))))
PassengerId      Pclass        Name         Sex         Age       SibSp       Parch      Ticket        Fare       Cabin    Embarked
          0           0           0           0          86           0           0           0           0           0           0

> sapply( test , function(x) (sum(x==""))})
PassengerId      Pclass        Name         Sex         Age       SibSp       Parch      Ticket        Fare       Cabin    Embarked
          0           0           0           0          NA           0           0           0           0         327           0
```

Valors Nuls:

Les dades contenen valors **NA** a la columna ***Age***. Aquestes dades poden ser emplenades amb el valor mitjà o la mediana, però llavors un grup d'edat, precisament el d'aquest valor serà inflat artificialment, donant-li més rellevància de la que en realitat hauria de tenir. Una altra opció es obtenir un valor central per a cada grup i posar la freqüència d'aquests valors proporcional a la grandària de cada grup. Finalment es poden ignorar les dades perdudes i aquesta, es l'opció que es decideix prendre, ja que per obtenir un model predictiu s'ha de reduir el número de columnes.

Les dades contenen valors buits a la columna ***Cabin***. A aquesta columna falten molts valors, 327/891, que és un 36,7% i la informació que falta no és substituïble sense conèixer en profunditat que és el que va passar en aquell viatge i també sabem que haurem de descartar columnes per a la creació del model. Per tant s'ignoraran els valors nuls.

Valors **NA** a la columna ***Embarked***. Ja que la gran majoria de passatgers van embarcar a Southampton, podem afegir els dos valors desconeguts a aquest port sense que signifiqui una gran desviació.

```
> table ( train_r$Embarked )
      C   Q   S
  2 168  77 644
```

Els informem amb el valor "S"

```
> t <- train_r
> t$Embarked <- as.character( t$Embarked )
> t$Embarked [t$Embarked==""] <- "S"
> t$Embarked <- as.factor( t$Embarked )
> table ( t$Embarked )
  C   Q   S
168  77 646
```

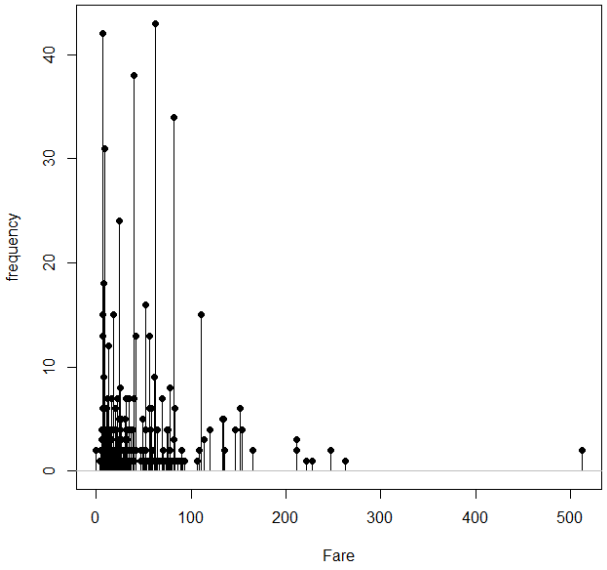
Outliers:

Anem al resum per fixar-nos en els valors límit:

```
> summary(train)
  PassengerId       Survived          Pclass                                      Name         Sex          Age       
 Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Abbing, Mr. Anthony                   :  1   female:314   Min.   : 0.42  
 1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Abbott, Mr. Rossmore Edward           :  1   male  :577   1st Qu.:20.12  
 Median :446.0   Median :0.0000   Median :3.000   Abbott, Mrs. Stanton (Rosa Hunt)      :  1                Median :28.00  
 Mean   :446.0   Mean   :0.3838   Mean   :2.309   Abelson, Mr. Samuel                   :  1                Mean   :29.70  
 3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000   Abelson, Mrs. Samuel (Hannah Wizosky):  1                3rd Qu.:38.00  
 Max.   :891.0   Max.   :1.0000   Max.   :3.000   Adahl, Mr. Mauritz Nils Martin        :  1                Max.   :80.00  
                                                  (Other)                               :885                NA's   :177    
     SibSp           Parch             Ticket         Fare               Cabin      Embarked    
 Min.   :0.000   Min.   :0.0000   1601    :  7   Min.   :  0.00             :687   Length:891  
 1st Qu.:0.000   1st Qu.:0.0000   347082  :  7   1st Qu.:  7.91   B96 B98   :  4   Class :character  
 Median :0.000   Median :0.0000   CA. 2343:  7   Median : 14.45   C23 C25 C27:  4   Mode  :character  
 Mean   :0.523   Mean   :0.3816   3101295 :  6   Mean   : 32.20   G6        :  4                     
 3rd Qu.:1.000   3rd Qu.:0.0000   347088  :  6   3rd Qu.: 31.00   C22 C26   :  3                     
 Max.   :8.000   Max.   :6.0000   CA 2144 :  6   Max.   :512.33   D         :  3                     
                                  (Other) :852                    (Other)   :186                     
```

Alguns valors criden l'atenció, Fare, SibSp i Parch tenen valors màxims alts, veiem si són o no ***outliers***:

Fare:




```
> with(train, stem.leaf(Fare, na.rm=TRUE))
1 | 2: represents 12
 leaf unit: 1
            n: 891
   16     0* | 0000000000000004
  336     0. | 56666666666677777777777777777777777777777777777777777777777777777777777
  457     1* | 00000000000000000000000000011111111122222222222233333333333333
  (58)    1. | 55555555555555555555555556666666666667778888888899999999
  376     2* | 00000000011111111111222233333333444444444
  334     2. | 55555555666666666666666666666666666666666666666666666666666666666666666
  240     3* | 000000000000000111111111111223333444444
  199     3. | 55555667789999999999999
  176     4* | 01112
  171     4. | 6666667999
  161     5* | 00111222222222233333
  141     5. | 5555566666666677779
  122     6* | 1113
  118     6. | 5566999999999
  105     7* | 11133333
HI: 75.25 76.2917 76.7292 76.7292 76.7292 77.2875 77.2875 77.9583 77.9583 77.95
```

Al gràfic de tiges i fulles observem els valors màxims, que passen de 263 a 512. Possiblement es tractés de camarots de luxe, al ser un valor repetit i igual probablement no és un ***outlier***.

```
263 263 263 512.3292 512.3292 512.3292
```

Parch i SibSp:

```
> table( train$Parch )
  0   1   2   3   4   5   6 
678 118  80   5   4   5   1 

> table( train$SibSp )
  0   1   2   3   4   5   8 
608 209  28  16  18   5   7 
```

Observem que els valors no són excessius i sembla que concorden fills i germans o gairebé. No semblen ***outliers***.

4. Anàlisi de les dades

4.1 Selecció dels grups de dades.

Dividirem en dos vessants els grups a seleccionar. Per una part els que seran sotmesos a proves estadístiques i per l'altra el que són necessaris per generar un model predictiu.

Dades per estadística:

```
# agrupacio per estadistiques
# menors d'edat / majors d'edat
train.men <- train[ train[,6] < 18 ,]
train.maj <- train [ train[,6] >= 18


# no supervivents de primera i de tercera classe
train.no1 <- train[ train[,3] == 1 & train[,2]==0 ,]
train.no3 <- train [ train[,3] == 3 & train[,2]==0 ,]
```

Dades per model predictiu:

Observem que hi ha pocs factors al dataset reduit:

```
> str( train_r )
'data.frame': 891 obs. of  7 variables:
 $ Survived: int  0 1 1 1 0 0 0 0 1 1 ...
 $ Pclass  : int  3 1 3 1 3 3 1 3 3 2 ...
 $ 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 ...
 $ Embarked: Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

Que ens queda confirmat en quant intentem generar el model:

```
> mod <- C5.0(Survived~.,data = train_r , rules=TRUE)
<
Missatges
[12] NOTA: Fitxer d'instruccions salvat a logfilename
[13] ERROR: C5.0 models require a factor outcome
```

Així doncs, convertim tots el camps que siguin susceptibles de ser factors, que ja són tots els que hem deixat:

```
> data <- train_r
> data$Survived <- as.factor(data$Survived)
> data$Pclass <- as.factor(data$Pclass)
> data$SibSp <- as.factor(data$SibSp)
> data$Parch <- as.factor(data$Parch)
> edat <- discretize(data$Age,categories=5)
> edat <- factor(edat, labels=c("nen","jove","adult","madur","vell"))
> data$Age <- edat
```

```
> train_f <- data [,c("Survived","Pclass","Sex","Age","SibSp","Parch","Embarked")]
```

```
> mod <- C5.0(Survived~.,data = train_f , rules=TRUE)
c50 code called exit with value 1
> summary(mod)
Call:
C5.0.formula(formula = Survived ~ ., data = train_f, rules = TRUE)
C5.0 [Release 2.07 GPL Edition]    Sat Jun 02 17:13:04 2018
-------------------------------
*** line 11 of `undefined.names': missing name or value before `,'
Error limit exceeded
```

El model no funciona amb tantes columnes, llavors s'ha de decidir que treure, per decidir-ho mirem les freqüències dels valors de les columnes:

```
> surv = subset(data,Survived=="1")
> notsurv = subset(data,Survived=="0")
```

```
> table( surv$Age )
  nen  jove adult madur  vell
   55   128    76    30     1

> table( notsurv$Age )
  nen  jove adult madur  vell
   45   218   112    39    10
```

```
> table( surv$Sex )
female   male
   233    109

> table( notsurv$Sex )
female   male
    81    468
```

```
> table( surv$Pclass )
  1   2   3
136  87 119

> table( notsurv$Pclass )
  1   2   3
 80  97 372
```

```
> table( surv$Embarked )
      C   Q   S
  2  93  30 217

> table( notsurv$Embarked )
      C   Q   S
  0  75  47 427
```

```
> table( surv$SibSp )
  0   1   2   3   4   5   8
210 112  13   4   3   0   0

> table( notsurv$SibSp )
  0   1   2   3   4   5   8
398  97  15  12  15   5   7
```

```
> table( surv$Parch )
  0   1   2   3   4   5   6
233  65  40   3   0   1   0

> table( notsurv$Parch )
  0   1   2   3   4   5   6
445  53  40   2   4   4   1
```

No totes les combinacions de columnes funcionen correctament, traient només una de les que relacionen familiars no és suficient. Traure la classe i el sexe tampoc funciona. Així doncs aquesta és una de les possibilitats:

```
# factorized columns
train_f <- data [,c("Survived","Pclass","Sex","Age","SibSp")]
test_f <- data [,c("Pclass","Sex","Age","SibSp")]
```

4.2.1 Comprovació de la normalitat

Per fer aquesta comprovació farem servir el test Shapiro-Wilks que ja ve integrada en R. Només es pot comprovar en les variables numèriques o enteres:

```
> p_val <- shapiro.test( train.no1$Age )$p.value
> p_val
[1] 0.3882723

> p_val <- shapiro.test( train.no3$Age )$p.value
> p_val
[1] 0.00003562277
```

la variable segueix una distribució normal en el cas de les víctimes de primera classe, en el cas de les de tercera classe no és així ja que el p-valor de la prova és inferior a 0,05

4.2.2 Comprovació de la homogeneïtat de la variància:

Per fer aquesta comprovació farem servir el test de Levene que ja ve integrada en R. En R-Commander només caldrà anar al desplegable Estadistics > Variancies > Test de Test de Levene.

```
> with(train, tapply(Survived, Sex, var, na.rm=TRUE))
   female      male
0.1920291 0.1534879

> leveneTest(Survived ~ Sex, data=train, center="median")
Levene's Test for Homogeneity of Variance (center = "median")
       Df F value  Pr(>F)
group   1  5.8041 0.01619 *
      889
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Com el p-valor es < 0,05 [0,01619] es rebutja la hipòtesi nul·la i es pot afirmar que presenta variàncies diferents. Nota: els subconjunts per estadístiques no donen un valor correcte pel p-valor.

4.3 Aplicació de proves estadístiques:

Contrast d'hipòtesi:

Aquest contrast d'hipòtesi unilateral sobre les dues mostres i la diferència de mitjanes es de grandària major de 30, així que no cal que segueixi una distribució normal per aquest test.

```
# no supervivents de primera i de tercera classe
train.no1 <- train[ train[,3] == 1 & train[,2]==0 ,]
train.no3 <- train [ train[,3] == 3 & train[,2]==0 ,]
```

Suposem que els passatgers de 3a que han mort eren més joves que els que van morir sent de 1a classe, així doncs:

Hipòtesi nul·la H_0: $\mu_1 - \mu_2 = 0$

Hipòtesi alternativa H_1: $\mu_1 - \mu_2 > 0$

μ_1 és la mitjana d'edat de les víctimes de 1a classe i μ_2 és la mitjana d'edat de les víctimes de 3a classe.

$\alpha = 0{,}05$.

```
> t.test( train.no1$Age , train.no3$Age ,alternative = "greater")

        Welch Two Sample t-test

data:  train.no1$Age and train.no3$Age
t = 8.3498, df = 83.488, p-value = 6.279e-13
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 13.72545      Inf
sample estimates:
mean of x mean of y
 43.69531  26.55556
```

obté un p-valor clarament inferior a 0,05, així doncs podem rebutjar la hipòtesi nul·la i afirmar que les víctimes de 3a classe eren més joves que les víctimes de 1a classe

Correlació:

Com les distribucions de les columnes no segueixen la normal, cal fer la correlació amb el mètode de Spearman. Provarem la correlació entre la supervivència i altres variables.

Primer passem els valors del *dataframe* a vectors, ja que la funció opera amb vectors, i a continuació s'efectua el test:

```
# correlation
vsur <- as.vector( train$Survived )
vpclass <- as.vector( train$Pclass )
vage <- as.vector( train$Age )
vsib <- as.vector( train$SibSp )
vfare <- as.vector( train$Fare )
```

```
> st = cor.test( vsur , vpclass , method = "spearman", exact=FALSE)
        Spearman's rank correlation rho

data:  vsur and vpclass
S = 157940000, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
-0.3396679
```

correlació inversa dèbil (-0,33)

```
> st = cor.test( vsur , vage , method = "spearman", exact=FALSE)
        Spearman's rank correlation rho

data:  vsur and vage
S = 63855000, p-value = 0.1606
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
-0.0525653
```

correlació inversa quasi nul·la (-0,05)

```
> st = cor.test( vsur , vsib , method = "spearman", exact=FALSE)
        Spearman's rank correlation rho

data:  vsur and vsib
S = 107410000, p-value = 0.007941
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
0.08887948
```

correlació quasi nul·la (0,08)

```
> st = cor.test( vsur , vfare , method = "spearman", exact=FALSE)
        Spearman's rank correlation rho

data:  vsur and vfare
S = 79726000, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.3237361
```

correlació dèbil (0,32)

Model predictiu:

Per efectuar regressió lineal és necessari que la homoscedasticitat, és a dir que els valors de les variàncies de les dades siguin homogènies. Així doncs serà obviat, en canvi es mostra el model predictiu basat en l'algorisme **C5.0**:

```
> train_f <- data [,c("Survived", "Pclass", "Sex",  "SibSp" )]
> mod <- C5.0(Survived~.,data = train_f , rules=TRUE)
> summary(mod)

Call:
C5.0.formula(formula = Survived ~ ., data = train_f, rules = TRUE)
C5.0 [Release 2.07 GPL Edition]    Sat Jun 02 18:45:30 2018
-------------------------------
Class specified by attribute `outcome'
Read 891 cases (4 attributes) from undefined.data
Rules:

Rule 1: (577/109, lift 1.3)
        Sex = male
        ->  class 0  [0.810]

Rule 2: (491/119, lift 1.2)
        Pclass = 3
        ->  class 0  [0.757]

Rule 3: (170/9, lift 2.5)
        Pclass in (1, 2)
        Sex = female
        ->  class 1  [0.942]

Rule 4: (187/40, lift 2.0)
        Sex = female
        SibSp in (0, 2)
        ->  class 1  [0.783]
```

```
Evaluation on training data (891 cases):

             Rules
       ----------------
         No      Errors

          4  174(19.5%)   <<


        (a)   (b)    <-classified as
       ----  ----
        504    45    (a): class 0
        129   213    (b): class 1


      Attribute usage:

       93.71% Sex
       74.19% Pclass
       20.99% SibSp
```

D'aquesta manera obtenim un model que es composa de regles amb una proporció d'us elevada (Sex i Pclass) i que poden formar un bon candidat per a la predicció

Passem a la predicció:

```
> pred <- predict( mod, test)
> taula <- table(pred,gender_submission$Survived)
> taula
pred   0  1
   0 266  24
   1   0 128
> percent <- 100*sum(diag(taula)) / sum(taula)
> percent
[1] 94.25837
```

El que ens mostra que el model és molt precís al comparar-lo amb els resultats reals que extraiem de ***gender_submission***.

5. Representació dels resultats

Correlacions:

Per tal de fer les representacions mes detallades alguna columna, com la d'edat, es modifica per omplir tots el resultats nuls, per esbrinar quin és el millor mètode observem la freqüència a cada grup que havíem factoritzat:

```
table(train_f$Age)
nen  jove adult madur  vell
139   139   150   138   148
```

s'observa que gairebé no hi ha desviació si utilitzem la mitjana, així llavors:

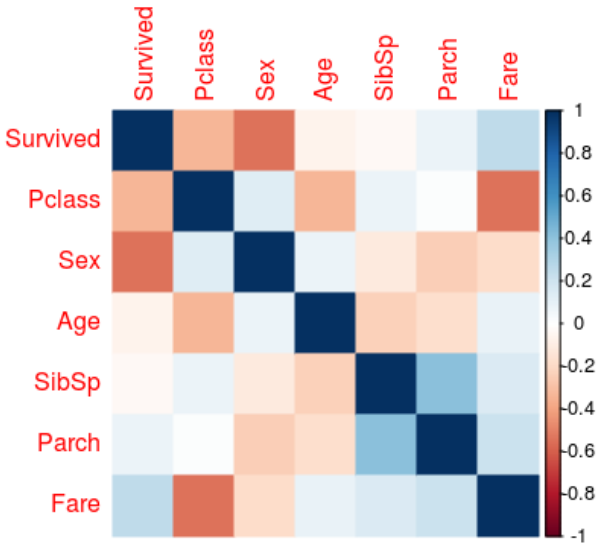
```
train_num <- train [,c("Survived","Pclass","Sex","Age","SibSp","Parch","Fare")]
train_num$Sex <- as.integer(train_num$Sex)
```

```
mean(train$Age, na.rm=TRUE)
[1] 29.69912
train_num$Age [is.na(train_num$Age)] <- 29.7
mean(train_num$Age)
[1] 29.69929
```

omplim els valor amb una aproximació de 29,7

Ara ja podem mostrar una taula de correlacions pel conjunt de dades ***train_num***:

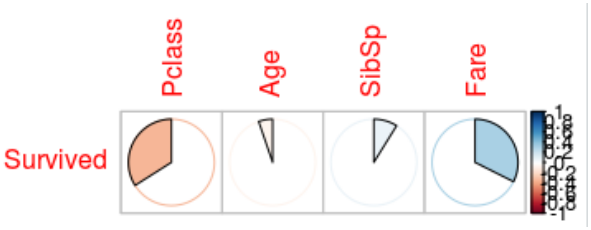
```
M <- cor(train_num)
corrplot(M, method="color")
```



També es pot crear una taula de correlacions contra la columna de supervivents pel mètode de Spearman:

```
st1 = cor.test( vsur , vpclass , method = "spearman", exact=FALSE)
st2 = cor.test( vsur , vage , method = "spearman", exact=FALSE)
st3 = cor.test( vsur , vsib , method = "spearman", exact=FALSE)
st4 = cor.test( vsur , vfare , method = "spearman", exact=FALSE)
```

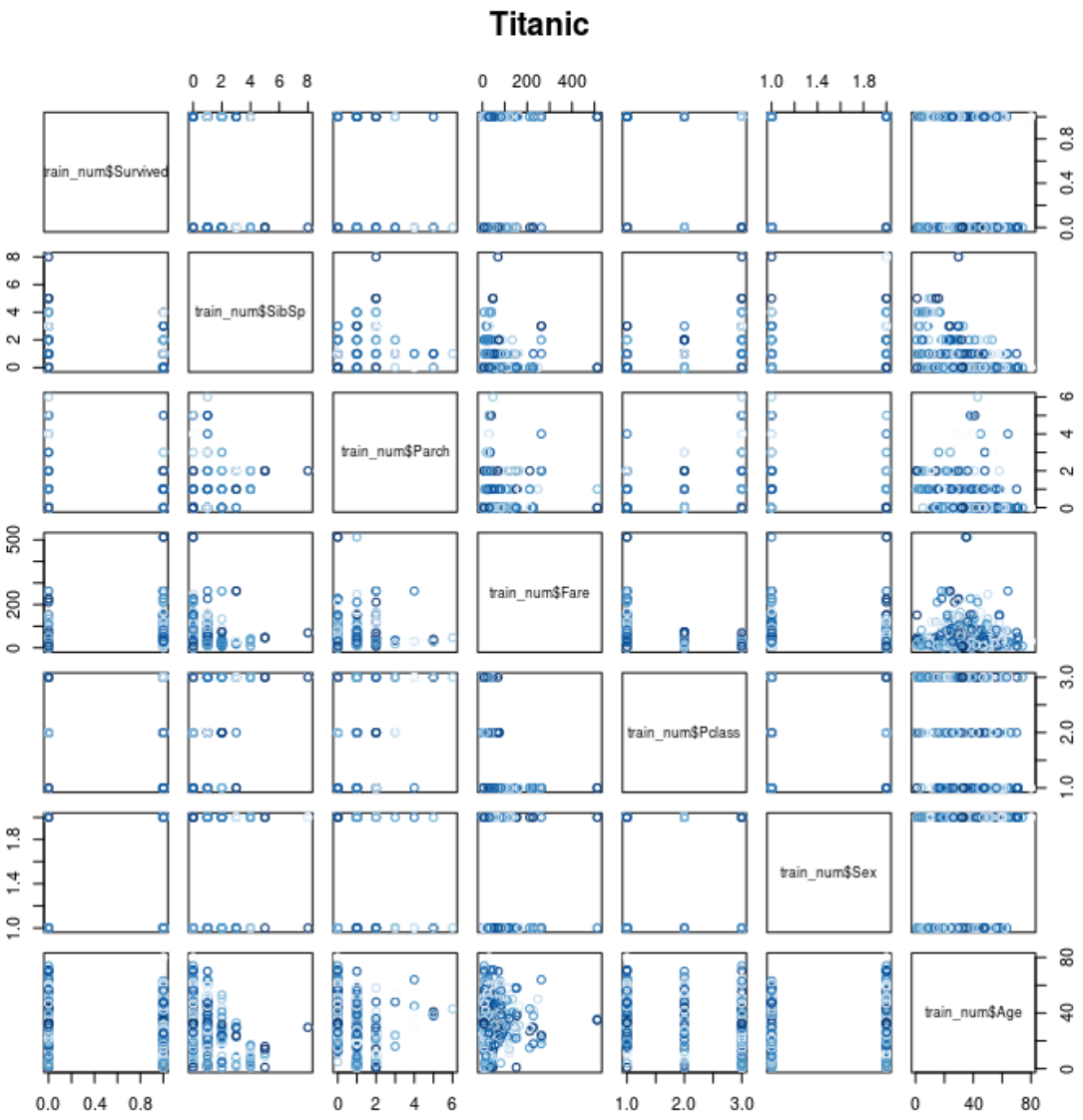
```
tcor <- rbind(c(st1$estimate,st2$estimate,st3$estimate,st4$estimate))
colnames(tcor) <- c("Pclass","Age","SibSp","Fare")
rownames(tcor) <- c("Survived")
tcor
          Pclass        Age       SibSp       Fare
Survived -0.3396679 -0.0525653 0.08887948 0.3237361
corrplot(tcor, method="pie")
```



ScatterPlot:

Un interessant gràfic és el que mostra aparellades les dades del conjunt:

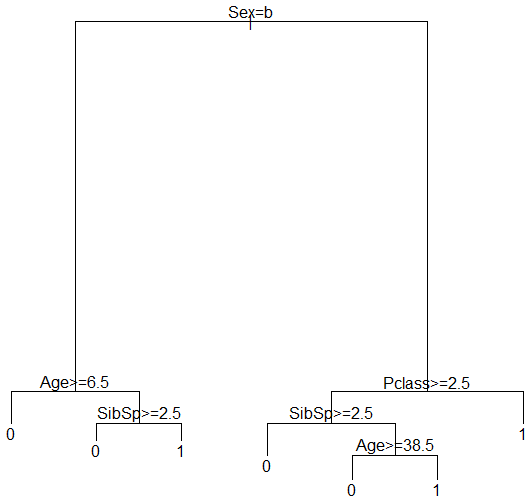
```
pairs(~train_num$Survived+train_num$SibSp+train_num$Parch+train_num$Fare+train_num$Pclass+train_num$Sex+
train_num$Age,data=train_num,      main="Titanic",col=blues9)
```

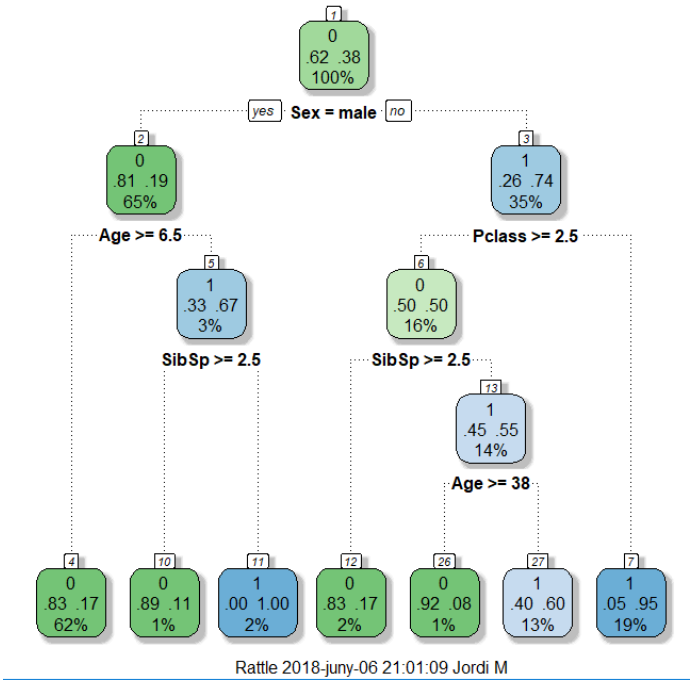


Models:

Finalment, els models predictius també poden aportar gràfiques per entendre les dades, en aquest cas donat el model obtingut amb la llibreria C50, encara que dóna resultats molt bons, no he estat capaç de fer un plot correcte. En el seu lloc he hagut que modelitzar de nou amb la llibreria ***rpart*** per obtenir altres models però si dibuixables. S'adjunta el resultat a ***rpart_prediction.csv***

```
library(rpart)
fit <- rpart(Survived ~ Pclass + Sex + Age + SibSp,
             data=train,
     method="class")
plot(fit)
text(fit)
```





El mateix model però molt més detallat:

```
library(rpart.plot)
library(rattle)
library(RColorBrewer)
fancyRpartPlot(fit)
```

El model també té una capacitat de predicció elevada

```
Prediction <- predict(fit, test, type = "class")
submit <- data.frame(PassengerId = test$PassengerId, Survived = Prediction)
taula <- table(submit$Survived, gender_submission$Survived)
taula
percent <- 100*sum(diag(taula)) / sum(taula)
percent
[1] 97.12919
```

6. Resolució del problema:

Conclusions:

El dataset venia amb poques dades nul·les i en la gran majoria totes les dades eren de qualitat. Això ha ajudat per generar versions apropiades segons les funcions a les que s'havien de sotmetre les dades. Els models c5.0 precisaven factors i les correlacions necessitaven dades numèriques. Les transformacions s'han efectuat amb R, tant en l'entorn de R-Studio com a R-Commander.

Degut a que aquest conjunt de dades és utilitzat en models predictius, s'ha pogut portar a terme tant la vessant estadística de les dades com la vessant de modelització i predicció. Per a aprofundir en la primera em preparat el dataset per poder fer un contrast d'hipòtesi i correlacions, però com les dades no seguien una distribució normal ni eren homogenis els valors de les diferents variàncies per poder efectuar regressió lineal.

Contrast d'hipòtesi:

Hem pogut determinar que els passatgers de 3a que han mort eren més joves que els que van morir sent de 1a classe ja que les mitjanes han estat comparables al ser de grandària major a 30.

Correlacions:

Trobem que hi ha una correlació elevada entre la supervivència i el sexe, i en menor mesura entre la supervivència i la classe. Evidentment el preu del bitllet depèn de la classe en la que s'embarca el viatger, per tant hi ha correlació. També s'ha pogut esbrinar que la edat no ha tingut influència en la probabilitat de sobreviure, ni tampoc la quantitat de parets que varen embarcar.

Paritat:

El gràfic de paritat del dataset no ens mostra amb claredat cap desviació o tret característic, cosa que normalment pot fer-nos descobrir, però no en aquest cas. Bàsicament la distribució en classes, sexe i edats, però cap informació nova.

Models:

El primer model, el ***c5.0***, ja mostra que es poden obtenir regles que ajusten amb bastant precisió la probabilitat de supervivència amb 94,3 % pel sexe i 74,2 % per la classe.

El model ***rpart*** (*Recursive Partitioning and Regression Trees*) detalla encara més els resultats, donant regles més granulars i que confirmen el que es sospita en les correlacions i es mostra en l'anterior model. Les dones van morir menys, i les de classe alta van morir menys encara. Inclús els homes de classe alta varen morir en una proporció molt més baixa, que van quedar els mateixos que els de tercera classe, però de tercera classe hi anaven el triple d'homes originalment.

Resultat per aquest model a ***rpart_prediction.csv***

7. Codi:

Totes les operacions s'han efectuat en R.

```
## ----load_libraries, include=FALSE-------------------------------------
library(arules)
library(arulesViz)
library(plyr)
## --- model libraries
library(C50)
library(rpart)
## -- plot libraries
library(corrplot)
library(rpart.plot)
library(rattle)
library(RColorBrewer)
library(tcltk, pos=19)
library(aplpack, pos=19)

## ---- echo=TRUE-------------------------------------------------------
# read data
gender_submission <- read.csv("../dataset/gender_submission.csv")
test <- read.csv("../dataset/test.csv")
train <- read.csv("../dataset/train.csv")

## ---- echo=TRUE-------------------------------------------------------
# cleaning columns
t <- train_r
t$Embarked <- as.character( t$Embarked )
t$Embarked [t$Embarked==""] <- "S"
t$Embarked <- as.factor( t$Embarked )
table ( t$Embarked )
```

```
train_r <- t
str(train_r)

## ---- echo=TRUE-----------------------------------------------------------
# Select columns
train_r <- train [,c("Survived","Pclass","Sex","Age","SibSp","Parch","Embarked")]
test_r <- test [,c("Pclass","Sex","Age","SibSp","Parch","Embarked")]

## ---- echo=TRUE-----------------------------------------------------------
# factorize train data
data <- train_r

data$Survived <- as.factor(data$Survived)
data$Pclass <- as.factor(data$Pclass)
data$SibSp <- as.factor(data$SibSp)
data$Parch <- as.factor(data$Parch)
data$Name <- as.character(data$Name)
data$Ticket <- as.character( data$Ticket )
data$Cabin <- as.character( data$Cabin )

edat <- discretize(data$Age,categories=5)
edat <- factor(edat, labels=c("nen","jove","adult","madur","vell"))
data$Age <- edat

train_f <- data

# factorize test data for to compare with train
data <- test

data$Survived <- as.factor(data$Survived)
data$Pclass <- as.factor(data$Pclass)
data$SibSp <- as.factor(data$SibSp)
data$Parch <- as.factor(data$Parch)
data$Name <- as.character(data$Name)
data$Ticket <- as.character( data$Ticket )
data$Cabin <- as.character( data$Cabin )

edat <- discretize(data$Age,categories=5)
edat <- factor(edat, labels=c("nen","jove","adult","madur","vell"))
# mean by group:
(0.42+16.34)/2  = 8.38
(16.34+32.25)/2 = 24.3
(32.25+48.17)/2 = 40.21
(48.17+64.08)/2 = 56.13
(64.08+80)/2    = 72.04
```

```
data$Age <- edat

test_f <- data

## ---- echo=TRUE-----------------------------------------------------------
# Quality of data
sapply( train , function(x) (sum(is.na(x))))
sapply( train , function(x) {sum(x=="")})
# sample filtering
idx <- sample(1:nrow( train ),5)
train [idx,] # filter 5 rows of data



## ---- echo=TRUE-----------------------------------------------------------

# factorized columns
train_f <- data [,c("Survived","Pclass","Sex","Age","SibSp")]
test_f <- data [,c("Pclass","Sex","Age","SibSp")]

# frecuencies
surv = subset(data,Survived=="1")
notsurv = subset(data,Survived=="0")
table( surv$Age )
table( notsurv$Age )
table( surv$Sex )
table( notsurv$Sex )
table( surv$Pclass )
table( notsurv$Pclass )
table( surv$Embarked )
table( notsurv$Embarked )
table( surv$SibSp )
table( notsurv$SibSp )
table( surv$Parch )
table( notsurv$Parch )

# Predictive Modeling
mod <- C5.0(Survived~.,data = train_f , rules=TRUE)
summary(mod)
pred <- predict( mod, test_f)
taula <- table(pred,gender_submission$Survived)
taula
percent <- 100*sum(diag(taula)) / sum(taula)
percent

## ---- echo=TRUE-----------------------------------------------------------
```

```
# plots
# tiges i fulles:
library(tcltk, pos=19)
library(aplpack, pos=19)
with(train, stem.leaf(Fare, na.rm=TRUE))
# punts (Outliers)
with(train, discretePlot(Fare, scale="frequency"))

## ---- echo=TRUE-----------------------------------------------------------
# save csv
write.csv(train_f, "D:\\Dropbox\\UOC-MA1\\TCVD_PRAC2\\dataset\\train_f.csv", row.names = FALSE)
write.csv(test_f, "D:\\Dropbox\\UOC-MA1\\TCVD_PRAC2\\dataset\\test_f.csv", row.names = FALSE)

# agrupacio per estadistiques
# menors d'edat / majors d'edat
train.men <- train[ train[,6] < 18 ,]
train.maj <- train [ train[,6] >= 18

# no supervivents de primera i de tercera classe
train.no1 <- train[ train[,3] == 1 & train[,2]==0 ,]
train.no3 <- train [ train[,3] == 3 & train[,2]==0 ,]



# variance analysys - Estadistics - Variancies - Test de Levene
with(train, tapply(Survived, Sex, var, na.rm=TRUE))
leveneTest(Survived ~ Sex, data=train, center="median")
# normality
p_val <- shapiro.test( train$Survived )$p.value
p_val <- shapiro.test( train$PassengerId )$p.value
p_val <- shapiro.test( train$Age )$p.value
p_val <- shapiro.test( train$SibSp )$p.value
p_val <- shapiro.test( train$Parch )$p.value
p_val <- shapiro.test( train$Fare )$p.value

# contraste de hipotesis
with(train, (t.test(Age, alternative='greater', mu=17.0, conf.level=.95)))

# correlation
vsur <- as.vector( train$Survived )
vpclass <- as.vector( train$Pclass )
vage <- as.vector( train$Age )
vsib <- as.vector( train$SibSp )
vfare <- as.vector( train$Fare )

st1 = cor.test( vsur , vpclass , method = "spearman", exact=FALSE)
```

```
st2 = cor.test( vsur , vage , method = "spearman", exact=FALSE)
st3 = cor.test( vsur , vsib , method = "spearman", exact=FALSE)
st4 = cor.test( vsur , vfare , method = "spearman", exact=FALSE)

# plot statistics

train_num <- train [,c("Survived","Pclass","Sex","Age","SibSp","Parch","Fare")]
train_num$Sex <- as.integer(train_num$Sex)
train_num$Age [is.na(train_num$Age)] <- 29.7
write.csv(train_num, "D:\\Dropbox\\UOC-MA1\\TCVD_PRAC2\\dataset\\train_num.csv", row.names = FALSE)

install.packages("corrplot")
library(corrplot)
M <- cor(train_num)
corrplot(M, method="circle")
corrplot(M, method="color")

table(train_f$Age)
nen  jove adult madur  vell
139   139   150   138   148

mean(train$Age, na.rm=TRUE)
[1] 29.69912
train_num$Age [is.na(train_num$Age)] <- 29.7
mean(train_num$Age)
[1] 29.69929

M <- cor(train_num)
corrplot(M, method="color")

tcor <- rbind(c(st1$estimate,st2$estimate,st3$estimate,st4$estimate))
colnames(tcor) <- c("Pclass","Age","SibSp","Fare")
rownames(tcor) <- c("Survived")
tcor
               Pclass        Age      SibSp      Fare
Survived -0.3396679 -0.0525653 0.08887948 0.3237361
corrplot(tcor, method="pie")

# altres plots
plot(train$Pclass, train$Fare, main="Fare x Pclass",  xlab="Classe", ylab="Preu bitllet", pch=19)

pairs(~train_num$Survived+train_num$SibSp+train_num$Parch+train_num$Fare+train_num$Pclass+train_num
$Sex+train_num$Age,data=train_num,      main="Titanic",col=blues9)

# model 2 i plots de model
```

```
library(rpart)
fit <- rpart(Survived ~ Pclass + Sex + Age + SibSp,
               data=train,
  method="class")
plot(fit)
text(fit)

library(rpart.plot)
library(rattle)
library(RColorBrewer)
fancyRpartPlot(fit)

Prediction <- predict(fit, test, type = "class")
submit <- data.frame(PassengerId = test$PassengerId, Survived = Prediction)
taula <- table(submit$Survived, gender_submission$Survived)
taula
percent <- 100*sum(diag(taula)) / sum(taula)
percent
[1] 97.12919
```

Recursos

Andrie de Vries, Joris Meys, (2012) How to Test Data Normality in a Formal Way in R [http://www.dummies.com/programming/r/how-to-test-data-normality-in-a-formal-way-in-r/] Dummies.com

http://vivaelsoftwarelibre.com/test-de-levene-homocedasticidad/

Robert I. Kabacoff, (2017) How to Test Data Normality in a Formal Way in R [https://www.statmethods.net/stats/ttest.html/] statmethods.net