CODI:

library(readr) # per llegir els fitxers csv

train <- read.csv("~/UOC/cicle/p3/dataset/train.csv")

View(train)

test <- read.csv("~/UOC/cicle/p3/dataset/test.csv")

View(test)

#mirem destructura de les dades

str(train)

str(test)

summary(train)

summary(test)

#Elimina els NAs dels Datasets

library(caTools) # llibreria per dividir el dataset

set.seed(123) # creacio d'un set

split = sample.split(train$Survived, SplitRatio = 0.8)

training\_set = subset(train, split == TRUE)

validation\_set = subset(train, split == FALSE)

final\_training\_set = train

colSums(is.na(training\_set)) # compta el nomber de NA per columna

colSums(is.na(validation\_set))

colSums(is.na(test))

#1.Ara conformem subsets amb les variables que volem utilitzar per entrenar el model, les variables que considerarem seran:Pclass, Sex, Age, Sibsp, Parch, Fare, Embarked, survived.

training\_set\_clean <- subset(training\_set, select = -c(PassengerId,Ticket,Cabin,Name))

validation\_set\_clean <- subset(validation\_set, select = -c(PassengerId,Ticket,Cabin,Name))

test\_clean <- subset(test, select = -c(Ticket,Cabin,Name))

final\_training\_set\_clean <- subset(final\_training\_set, select = -c(PassengerId,Ticket,Cabin,Name))

#2.Tambe tenim 263 varialbes d'edat perdudes. Representa un 20% es quantios. Substituirem els NA per el median dels datasets.

training\_set\_clean$Age[is.na(training\_set\_clean$Age)]<-median(na.omit(training\_set\_clean$Age)) #substitueix els perduts amb el median

validation\_set\_clean$Age[is.na(validation\_set\_clean$Age)]<-median(na.omit(validation\_set\_clean$Age)) # idem

test\_clean$Age[is.na(test\_clean$Age)]<-median(na.omit(test\_clean$Age)) # idem

final\_training\_set\_clean$Age[is.na(final\_training\_set\_clean$Age)]<-median(na.omit(final\_training\_set\_clean$Age))

#El punt negatiu d'utilitzar el median es que tindrem un 20% de dades mes sobre el median.

plot(test\_clean$Age)

#2.Per la variable fare hi ha un valor perdut. Utilitzarem el median coma substitut.

test\_clean$Fare[is.na(test\_clean$Fare)]<-median(na.omit(test\_clean$Fare))# sustitueix pel median.

test\_clean$Fare[is.na(test\_clean$Fare)]<-median(na.omit(test\_clean$Fare))

#3.Hi ha dos valors perduts a "embarked”.En aquest cas substituirem el NA per la localitzacio mes comuna.

table(training\_set\_clean$Embarked) # compta el nombre de tiquets per embarcament.

table(validation\_set\_clean$Embarked)

table(test\_clean$Embarked)

training\_set\_clean$Embarked[is.na(training\_set\_clean$Embarked)]<-"S" # substitueix els NA per la localitzacio mes comuna.

final\_training\_set\_clean$Embarked[is.na(final\_training\_set\_clean$Embarked)]<-"S" # idem

#ara verifiquem que no hi hagi NAS.

colSums(is.na(training\_set\_clean))

colSums(is.na(validation\_set\_clean))

colSums(is.na(test\_clean))

colSums(is.na(final\_training\_set\_clean))

#valors extrems, tots els valors semblen possibles, per tant no seliminen

boxplot(train$Age, main="Age")

boxplot(train$Pclass, main="PClass")

boxplot(train$SibSp, main="SibSp")

boxplot(train$Parch, main="Parch")

boxplot(train$Fare, main="fare")

#Ara passem a analitzar els valors buits

table(train$Cabin)

table(train$Age)

table(train$Survived)

table(train$Pclass)

table(train$Sex)

table(train$SibSp)

table(train$Parch)

table(train$Fare)

table(train$Embarked)

#Escalarem les variales que no son factors, l'edat i el fare.

training\_set\_clean[4] = scale(training\_set\_clean[4])#age

training\_set\_clean[7] = scale(training\_set\_clean[7])#fare

validation\_set\_clean[4] = scale(validation\_set\_clean[4])#age

validation\_set\_clean[7] = scale(validation\_set\_clean[7])#fare

test\_clean[4] = scale(test\_clean[4])#fare

test\_clean[7] = scale(test\_clean[7])#age

final\_training\_set\_clean[4] = scale(final\_training\_set\_clean[4])#age

final\_training\_set\_clean[7] = scale(final\_training\_set\_clean[7])

#escriu un arxiu csv

#enregistrem els arxius csv

write.csv(training\_set\_clean, file = "training\_set\_clean.csv", row.names = FALSE)

write.csv(validation\_set\_clean, file = "validation\_set\_clean.csv", row.names = FALSE)

write.csv(test\_clean, file = "test\_clean.csv", row.names = FALSE)

write.csv(final\_training\_set\_clean, file = "final\_training\_set\_clean.csv", row.names = FALSE)

library(readr) #per llegir els csv

training\_set\_clean <- read\_csv("training\_set\_clean.csv")

validation\_set\_clean <- read\_csv("validation\_set\_clean.csv")

final\_training\_set\_clean <- read\_csv("final\_training\_set\_clean.csv")

test\_clean <- read\_csv("test\_clean.csv")

head(training\_set\_clean,1)

#Com s'especifica anteriorment, s'utilitzen 8 variables per l'analisis:

#Survived: Sortida Binaria, 1 vol dir sobreviscut, 0 vol dir no ha sobreviscut.

#Pclass: Tipus de classe. 1 = Primera, 2 = Segona, 3 = Tercera

#Sex: Sortida binaria, home o dona.

#Age: Edat escalada per normalitzar la variable.

#SibSp: Nombre de germans / esposes a bord

#Parch: Nombre de pares / fills a bord

#Fare: Tarifa del tiquet escalada per normalitza la variable.

#Embarked: lloc d'embarcament

#Pel dataset “test\_clean”, tota la informacio menys la columna survived.

#algunes variales es codifiquen com a factors: “Survived”,“Pclass”,“Sex”, “Embarked”.

str(training\_set\_clean,give.attr = FALSE)

str(validation\_set\_clean,give.attr = FALSE)

str(final\_training\_set\_clean,give.attr = FALSE)

str(test\_clean,give.attr = FALSE)

#Encodifcar com a factor

training\_set\_clean$Survived <- as.factor(training\_set\_clean$Survived)

training\_set\_clean$Pclass <- as.factor(training\_set\_clean$Pclass)

training\_set\_clean$Sex <- as.factor(training\_set\_clean$Sex)

training\_set\_clean$Embarked <- as.factor(training\_set\_clean$Embarked)

validation\_set\_clean$Survived <- as.factor(validation\_set\_clean$Survived)

validation\_set\_clean$Pclass <- as.factor(validation\_set\_clean$Pclass)

validation\_set\_clean$Sex <- as.factor(validation\_set\_clean$Sex)

validation\_set\_clean$Embarked <- as.factor(validation\_set\_clean$Embarked)

final\_training\_set\_clean$Survived <- as.factor(final\_training\_set\_clean$Survived)

final\_training\_set\_clean$Pclass <- as.factor(final\_training\_set\_clean$Pclass)

final\_training\_set\_clean$Sex <- as.factor(final\_training\_set\_clean$Sex)

final\_training\_set\_clean$Embarked <- as.factor(final\_training\_set\_clean$Embarked)

test\_clean$Pclass <- as.factor(test\_clean$Pclass)

test\_clean$Sex <- as.factor(test\_clean$Sex)

test\_clean$Embarked <- as.factor(test\_clean$Embarked)

#1. L'arbre de decisio

#En la part que segueix es crea l'arbre de decisio i es dibuixen els resultats. La decisio a l'esquerra es menys atractiva que la solucio a la dreta pero produeix informacio sensitiva. La variable “Sex” denota ser molt mes important en la prediccio que qualsevol altra variable.

#La creacio de l'abre de decisio

library(rpart)

start.time <- Sys.time()

Dtree <- rpart(Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked,

data=training\_set\_clean,

method="class")

Dtree\_pred = predict(Dtree, newdata = validation\_set\_clean[-1], type = 'class')# eliminem amb -1 allo que volem predir, la variable survived.

end.time <- Sys.time()

time.takenDT <- end.time - start.time

time.takenDT

par(mfrow=c(1,2)) # imprimeix dos caracter horitzontalment

plot(Dtree)

text(Dtree)

library(rattle)

library(rpart.plot)

library(RColorBrewer)

fancyRpartPlot(Dtree)

#ara es testeja el model amb el set de validacio i creem, la matriu de confusio per definir, la validesa del model. La matriu de confusio en possibilita un analisi de l'exit de la prediccio . En l'eix horitzontal es tenen les classes i en leix vertical es tenen les classes predites.

#La matriu de confusio per l'arbre

#creracio de la matriu de confusio

Dtree\_cm = table(t(validation\_set\_clean[, 1]), Dtree\_pred)

# calculem el rati dexit total

Dtree\_success = sum(diag(Dtree\_cm))/sum(Dtree\_cm)\*100

Dtree\_success

# calculem el rati dexit per predir una supervivencia

Dtree\_success\_survived = Dtree\_cm[2,2]/(Dtree\_cm[2,1]+Dtree\_cm[2,2])\*100

Dtree\_success\_survived

# calcul del rati d'exit per predir la casualitat.

Dtree\_success\_notsurvived = Dtree\_cm[1,1]/(Dtree\_cm[1,2]+Dtree\_cm[1,1])\*100

Dtree\_success\_notsurvived

# prdit en vertical i actual en horitzontal ( 68 sobrevivents en el set de validacio )

Dtree\_cm

#El rati dexit de prediccio es de 77.53% , la prediccio exitosa de la supervivencia es del 63.24% mentres l'exit de prediccio per la no supervivencia es del 86.36%. El model ha tardat 0.06 segons en fer la prediccio.

#Ara provarem una aproximacio de prediccio des dun model de regresio

#Grupos de hombres y mujeres

training\_set\_clean.men <- training\_set\_clean[training\_set\_clean$Sex == "male",]

training\_set\_clean.women <- training\_set\_clean[training\_set\_clean$Sex == "female",]

train1<-training\_set\_clean # Utilitzem el mateix set per entrenar el model que amb el model darbre

train2<-validation\_set\_clean # Utilitzem el mateix set per validar el mdoel que amb el model darbre

#testeig de la normalitat

library(nortest)

alpha = 0.05

col.names = colnames(training\_set\_clean)

for (i in 1:ncol(training\_set\_clean)) {

if (i == 1) cat("Variables que no siguen una distribución normal:\n")

if (is.integer(training\_set\_clean[,i]) | is.numeric(training\_set\_clean[,i])) {

p\_val = ad.test(training\_set\_clean[,i])$p.value

if (p\_val < alpha) {

cat(col.names[i])

# Formateig de la sortida

if (i < ncol(training\_set\_clean) - 1) cat(", ")

if (i %% 3 == 0) cat("\n")

}

}

}

#Seguidament sestudia la homegeneitat dels sets train i validacio

library(car)

fligner.test(Survived ~ Age, data = training\_set\_clean)

#Edat mitja dels que van sobreviure i dels que van morir

aggregate(Age~Survived, data=training\_set\_clean, mean)

t.test(Age ~ Survived, data=training\_set\_clean)

# Generación de un modelo de regresion logit

model <- glm(Survived ~.,family=binomial(link='logit'),data=train1)

summary(model)

#Parch, Fare, EmbarkedQ no tenen prou rellevancia >0.05 les eliminem del model

stepmodel = step(model, direction="both")

formula(stepmodel)

summary(stepmodel)

#validacion del model amb el set de validacio

pred.train <- predict(model,train2)

pred.train <- ifelse(pred.train > 0.5,1,0)

# Mitjana de prediccio encertada

mean(pred.train==train2$Survived)

t1<-table(pred.train,train2$Survived)

# Presicio i sensibilidad del model

presicion<- t1[1,1]/(sum(t1[1,]))

recall<- t1[1,1]/(sum(t1[,1]))

presicion

recall

RESULTAT

> library(readr) # per llegir els fitxers csv

> train <- read.csv("~/UOC/cicle/p3/dataset/train.csv")

> View(train)

> test <- read.csv("~/UOC/cicle/p3/dataset/test.csv")

> View(test)

>

> #mirem destructura de les dades

> 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 ...

> str(test)

'data.frame': 418 obs. of 11 variables:

$ PassengerId: int 892 893 894 895 896 897 898 899 900 901 ...

$ Pclass : int 3 3 2 3 3 3 3 2 3 3 ...

$ Name : Factor w/ 418 levels "Abbott, Master. Eugene Joseph",..: 210 409 273 414 182 370 85 58 5 104 ...

$ Sex : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...

$ Age : num 34.5 47 62 27 22 14 30 26 18 21 ...

$ SibSp : int 0 1 0 0 1 0 0 1 0 2 ...

$ Parch : int 0 0 0 0 1 0 0 1 0 0 ...

$ Ticket : Factor w/ 363 levels "110469","110489",..: 153 222 74 148 139 262 159 85 101 270 ...

$ Fare : num 7.83 7 9.69 8.66 12.29 ...

$ Cabin : Factor w/ 77 levels "","A11","A18",..: 1 1 1 1 1 1 1 1 1 1 ...

$ Embarked : Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...

> summary(train)

PassengerId Survived Pclass Name Sex

Min. : 1.0 Min. :0.0000 Min. :1.000 Abbing, Mr. Anthony : 1 female:314

1st Qu.:223.5 1st Qu.:0.0000 1st Qu.:2.000 Abbott, Mr. Rossmore Edward : 1 male :577

Median :446.0 Median :0.0000 Median :3.000 Abbott, Mrs. Stanton (Rosa Hunt) : 1

Mean :446.0 Mean :0.3838 Mean :2.309 Abelson, Mr. Samuel : 1

3rd Qu.:668.5 3rd Qu.:1.0000 3rd Qu.:3.000 Abelson, Mrs. Samuel (Hannah Wizosky): 1

Max. :891.0 Max. :1.0000 Max. :3.000 Adahl, Mr. Mauritz Nils Martin : 1

(Other) :885

Age SibSp Parch Ticket Fare Cabin Embarked

Min. : 0.42 Min. :0.000 Min. :0.0000 1601 : 7 Min. : 0.00 :687 : 2

1st Qu.:20.12 1st Qu.:0.000 1st Qu.:0.0000 347082 : 7 1st Qu.: 7.91 B96 B98 : 4 C:168

Median :28.00 Median :0.000 Median :0.0000 CA. 2343: 7 Median : 14.45 C23 C25 C27: 4 Q: 77

Mean :29.70 Mean :0.523 Mean :0.3816 3101295 : 6 Mean : 32.20 G6 : 4 S:644

3rd Qu.:38.00 3rd Qu.:1.000 3rd Qu.:0.0000 347088 : 6 3rd Qu.: 31.00 C22 C26 : 3

Max. :80.00 Max. :8.000 Max. :6.0000 CA 2144 : 6 Max. :512.33 D : 3

NA's :177 (Other) :852 (Other) :186

> summary(test)

PassengerId Pclass Name Sex Age

Min. : 892.0 Min. :1.000 Abbott, Master. Eugene Joseph : 1 female:152 Min. : 0.17

1st Qu.: 996.2 1st Qu.:1.000 Abelseth, Miss. Karen Marie : 1 male :266 1st Qu.:21.00

Median :1100.5 Median :3.000 Abelseth, Mr. Olaus Jorgensen : 1 Median :27.00

Mean :1100.5 Mean :2.266 Abrahamsson, Mr. Abraham August Johannes : 1 Mean :30.27

3rd Qu.:1204.8 3rd Qu.:3.000 Abrahim, Mrs. Joseph (Sophie Halaut Easu): 1 3rd Qu.:39.00

Max. :1309.0 Max. :3.000 Aks, Master. Philip Frank : 1 Max. :76.00

(Other) :412 NA's :86

SibSp Parch Ticket Fare Cabin Embarked

Min. :0.0000 Min. :0.0000 PC 17608: 5 Min. : 0.000 :327 C:102

1st Qu.:0.0000 1st Qu.:0.0000 113503 : 4 1st Qu.: 7.896 B57 B59 B63 B66: 3 Q: 46

Median :0.0000 Median :0.0000 CA. 2343: 4 Median : 14.454 A34 : 2 S:270

Mean :0.4474 Mean :0.3923 16966 : 3 Mean : 35.627 B45 : 2

3rd Qu.:1.0000 3rd Qu.:0.0000 220845 : 3 3rd Qu.: 31.500 C101 : 2

Max. :8.0000 Max. :9.0000 347077 : 3 Max. :512.329 C116 : 2

(Other) :396 NA's :1 (Other) : 80

>

> #Elimina els NAs dels Datasets

>

> library(caTools) # llibreria per dividir el dataset

> set.seed(123) # creacio d'un set

> split = sample.split(train$Survived, SplitRatio = 0.8)

> training\_set = subset(train, split == TRUE)

> validation\_set = subset(train, split == FALSE)

> final\_training\_set = train

>

> colSums(is.na(training\_set)) # compta el nomber de NA per columna

PassengerId Survived Pclass Name Sex Age SibSp Parch Ticket

0 0 0 0 0 141 0 0 0

Fare Cabin Embarked

0 0 0

>

> colSums(is.na(validation\_set))

PassengerId Survived Pclass Name Sex Age SibSp Parch Ticket

0 0 0 0 0 36 0 0 0

Fare Cabin Embarked

0 0 0

> colSums(is.na(test))

PassengerId Pclass Name Sex Age SibSp Parch Ticket Fare

0 0 0 0 86 0 0 0 1

Cabin Embarked

0 0

>

>

>

> #1.Ara conformem subsets amb les variables que volem utilitzar per entrenar el model, les variables que considerarem seran:Pclass, Sex, Age, Sibsp, Parch, Fare, Embarked, survived.

>

> training\_set\_clean <- subset(training\_set, select = -c(PassengerId,Ticket,Cabin,Name))

> validation\_set\_clean <- subset(validation\_set, select = -c(PassengerId,Ticket,Cabin,Name))

> test\_clean <- subset(test, select = -c(Ticket,Cabin,Name))

> final\_training\_set\_clean <- subset(final\_training\_set, select = -c(PassengerId,Ticket,Cabin,Name))

>

> #2.Tambe tenim 263 varialbes d'edat perdudes. Representa un 20% es quantios. Substituirem els NA per el median dels datasets.

>

> training\_set\_clean$Age[is.na(training\_set\_clean$Age)]<-median(na.omit(training\_set\_clean$Age)) #substitueix els perduts amb el median

> validation\_set\_clean$Age[is.na(validation\_set\_clean$Age)]<-median(na.omit(validation\_set\_clean$Age)) # idem

> test\_clean$Age[is.na(test\_clean$Age)]<-median(na.omit(test\_clean$Age)) # idem

> final\_training\_set\_clean$Age[is.na(final\_training\_set\_clean$Age)]<-median(na.omit(final\_training\_set\_clean$Age))

>

> #El punt negatiu d'utilitzar el median es que tindrem un 20% de dades mes sobre el median.

>

> plot(test\_clean$Age)

>

> #2.Per la variable fare hi ha un valor perdut. Utilitzarem el median coma substitut.

> test\_clean$Fare[is.na(test\_clean$Fare)]<-median(na.omit(test\_clean$Fare))# sustitueix pel median.

> test\_clean$Fare[is.na(test\_clean$Fare)]<-median(na.omit(test\_clean$Fare))

>

>

> #3.Hi ha dos valors perduts a "embarked”.En aquest cas substituirem el NA per la localitzacio mes comuna.

>

> table(training\_set\_clean$Embarked) # compta el nombre de tiquets per embarcament.

C Q S

2 127 55 529

>

> table(validation\_set\_clean$Embarked)

C Q S

0 41 22 115

>

> table(test\_clean$Embarked)

C Q S

102 46 270

>

> training\_set\_clean$Embarked[is.na(training\_set\_clean$Embarked)]<-"S" # substitueix els NA per la localitzacio mes comuna.

> final\_training\_set\_clean$Embarked[is.na(final\_training\_set\_clean$Embarked)]<-"S" # idem

>

> #ara verifiquem que no hi hagi NAS.

>

> colSums(is.na(training\_set\_clean))

Survived Pclass Sex Age SibSp Parch Fare Embarked

0 0 0 0 0 0 0 0

> colSums(is.na(validation\_set\_clean))

Survived Pclass Sex Age SibSp Parch Fare Embarked

0 0 0 0 0 0 0 0

> colSums(is.na(test\_clean))

PassengerId Pclass Sex Age SibSp Parch Fare Embarked

0 0 0 0 0 0 0 0

> colSums(is.na(final\_training\_set\_clean))

Survived Pclass Sex Age SibSp Parch Fare Embarked

0 0 0 0 0 0 0 0

> #valors extrems, tots els valors semblen possibles, per tant no seliminen

> boxplot(train$Age, main="Age")

> boxplot(train$Pclass, main="PClass")

> boxplot(train$SibSp, main="SibSp")

> boxplot(train$Parch, main="Parch")

> boxplot(train$Fare, main="fare")

>

> #Ara passem a analitzar els valors buits

> table(train$Cabin)

A10 A14 A16 A19 A20 A23

687 1 1 1 1 1 1

A24 A26 A31 A32 A34 A36 A5

1 1 1 1 1 1 1

A6 A7 B101 B102 B18 B19 B20

1 1 1 1 2 1 2

B22 B28 B3 B30 B35 B37 B38

2 2 1 1 2 1 1

B39 B4 B41 B42 B49 B5 B50

1 1 1 1 2 2 1

B51 B53 B55 B57 B59 B63 B66 B58 B60 B69 B71 B73 B77

2 2 2 1 1 1 2

B78 B79 B80 B82 B84 B86 B94 B96 B98

1 1 1 1 1 1 4

C101 C103 C104 C106 C110 C111 C118

1 1 1 1 1 1 1

C123 C124 C125 C126 C128 C148 C2

2 2 2 2 1 1 2

C22 C26 C23 C25 C27 C30 C32 C45 C46 C47

3 4 1 1 1 1 1

C49 C50 C52 C54 C62 C64 C65 C68

1 1 2 1 1 2 2

C7 C70 C78 C82 C83 C85 C86

1 1 2 1 2 1 1

C87 C90 C91 C92 C93 C95 C99

1 1 1 2 2 1 1

D D10 D12 D11 D15 D17 D19 D20

3 1 1 1 2 1 2

D21 D26 D28 D30 D33 D35 D36

1 2 1 1 2 2 2

D37 D45 D46 D47 D48 D49 D50

1 1 1 1 1 1 1

D56 D6 D7 D9 E10 E101 E12

1 1 1 1 1 3 1

E121 E17 E24 E25 E31 E33 E34

2 1 2 2 1 2 1

E36 E38 E40 E44 E46 E49 E50

1 1 1 2 1 1 1

E58 E63 E67 E68 E77 E8 F E69

1 1 2 1 1 2 1

F G63 F G73 F2 F33 F38 F4 G6

1 2 3 3 1 2 4

T

1

> table(train$Age)

0.42 0.67 0.75 0.83 0.92 1 2 3 4 5 6 7 8 9 10 11 12 13 14 14.5 15 16 17

1 1 2 2 1 7 10 6 10 4 3 3 4 8 2 4 1 2 6 1 5 17 13

18 19 20 20.5 21 22 23 23.5 24 24.5 25 26 27 28 28.5 29 30 30.5 31 32 32.5 33 34

26 25 15 1 24 27 15 1 30 1 23 18 18 25 2 20 25 2 17 18 2 15 15

34.5 35 36 36.5 37 38 39 40 40.5 41 42 43 44 45 45.5 46 47 48 49 50 51 52 53

1 18 22 1 6 11 14 13 2 6 13 5 9 12 2 3 9 9 6 10 7 6 1

54 55 55.5 56 57 58 59 60 61 62 63 64 65 66 70 70.5 71 74 80

8 2 1 4 2 5 2 4 3 4 2 2 3 1 2 1 2 1 1

> table(train$Survived)

0 1

549 342

> table(train$Pclass)

1 2 3

216 184 491

> table(train$Sex)

female male

314 577

> table(train$SibSp)

0 1 2 3 4 5 8

608 209 28 16 18 5 7

> table(train$Parch)

0 1 2 3 4 5 6

678 118 80 5 4 5 1

> table(train$Fare)

0 4.0125 5 6.2375 6.4375 6.45 6.4958 6.75 6.8583 6.95 6.975 7.0458 7.05

15 1 1 1 1 1 2 2 1 1 2 1 7

7.0542 7.125 7.1417 7.225 7.2292 7.25 7.3125 7.4958 7.5208 7.55 7.6292 7.65 7.725

2 4 1 12 15 13 1 3 1 4 1 4 1

7.7292 7.7333 7.7375 7.7417 7.75 7.775 7.7875 7.7958 7.8 7.8292 7.8542 7.875 7.8792

1 4 2 1 34 16 1 6 1 2 13 1 4

7.8875 7.8958 7.925 8.0292 8.05 8.1125 8.1375 8.1583 8.3 8.3625 8.4042 8.4333 8.4583

1 38 18 1 43 1 1 1 1 1 1 1 1

8.5167 8.6542 8.6625 8.6833 8.7125 8.85 9 9.2167 9.225 9.35 9.475 9.4833 9.5

1 1 13 1 1 1 2 1 2 2 1 1 9

9.5875 9.825 9.8375 9.8417 9.8458 10.1708 10.4625 10.5 10.5167 11.1333 11.2417 11.5 12

2 2 1 1 1 1 2 24 1 3 2 4 1

12.275 12.2875 12.35 12.475 12.525 12.65 12.875 13 13.4167 13.5 13.7917 13.8583 13.8625

1 1 3 4 1 1 1 42 1 4 1 1 1

14 14.1083 14.4 14.4542 14.4583 14.5 15 15.0458 15.05 15.1 15.2458 15.5 15.55

1 1 2 7 3 7 1 1 1 1 5 8 1

15.7417 15.75 15.85 15.9 16 16.1 16.7 17.4 17.8 18 18.75 18.7875 19.2583

2 1 4 2 1 9 2 1 2 3 3 2 4

19.5 19.9667 20.2125 20.25 20.525 20.575 21 21.075 21.6792 22.025 22.3583 22.525 23

2 2 2 2 3 2 6 4 1 1 2 1 4

23.25 23.45 24 24.15 25.4667 25.5875 25.925 25.9292 26 26.25 26.2833 26.2875 26.3875

2 2 2 8 4 1 1 2 31 6 1 3 1

26.55 27 27.7208 27.75 27.9 28.5 28.7125 29 29.125 29.7 30 30.0708 30.5

15 2 5 4 6 1 1 2 5 3 6 2 5

30.6958 31 31.275 31.3875 32.3208 32.5 33 33.5 34.0208 34.375 34.6542 35 35.5

2 3 7 4 1 1 3 1 1 4 1 1 4

36.75 37.0042 38.5 39 39.4 39.6 39.6875 40.125 41.5792 42.4 46.9 47.1 49.5

2 2 1 4 1 2 6 1 3 1 6 1 1

49.5042 50 50.4958 51.4792 51.8625 52 52.5542 53.1 55 55.4417 55.9 56.4958 56.9292

2 1 1 1 2 7 3 5 2 1 2 7 2

57 57.9792 59.4 61.175 61.3792 61.9792 63.3583 65 66.6 69.3 69.55 71 71.2833

2 2 1 1 1 1 1 2 2 2 7 2 1

73.5 75.25 76.2917 76.7292 77.2875 77.9583 78.2667 78.85 79.2 79.65 80 81.8583 82.1708

5 1 1 3 2 3 2 2 4 3 2 1 2

83.1583 83.475 86.5 89.1042 90 91.0792 93.5 106.425 108.9 110.8833 113.275 120 133.65

3 2 3 2 4 2 2 2 2 4 3 4 2

134.5 135.6333 146.5208 151.55 153.4625 164.8667 211.3375 211.5 221.7792 227.525 247.5208 262.375 263

2 3 2 4 3 2 3 1 1 4 2 2 4

512.3292

3

> table(train$Embarked)

C Q S

2 168 77 644

>

>

> #Escalarem les variales que no son factors, l'edat i el fare.

>

> training\_set\_clean[4] = scale(training\_set\_clean[4])#age

> training\_set\_clean[7] = scale(training\_set\_clean[7])#fare

> validation\_set\_clean[4] = scale(validation\_set\_clean[4])#age

> validation\_set\_clean[7] = scale(validation\_set\_clean[7])#fare

> test\_clean[4] = scale(test\_clean[4])#fare

> test\_clean[7] = scale(test\_clean[7])#age

> final\_training\_set\_clean[4] = scale(final\_training\_set\_clean[4])#age

> final\_training\_set\_clean[7] = scale(final\_training\_set\_clean[7])

>

>

> #escriu un arxiu csv

>

> #enregistrem els arxius csv

>

> write.csv(training\_set\_clean, file = "training\_set\_clean.csv", row.names = FALSE)

> write.csv(validation\_set\_clean, file = "validation\_set\_clean.csv", row.names = FALSE)

> write.csv(test\_clean, file = "test\_clean.csv", row.names = FALSE)

> write.csv(final\_training\_set\_clean, file = "final\_training\_set\_clean.csv", row.names = FALSE)

>

> library(readr) #per llegir els csv

> training\_set\_clean <- read\_csv("training\_set\_clean.csv")

Parsed with column specification:

cols(

Survived = col\_integer(),

Pclass = col\_integer(),

Sex = col\_character(),

Age = col\_double(),

SibSp = col\_integer(),

Parch = col\_integer(),

Fare = col\_double(),

Embarked = col\_character()

)

> validation\_set\_clean <- read\_csv("validation\_set\_clean.csv")

Parsed with column specification:

cols(

Survived = col\_integer(),

Pclass = col\_integer(),

Sex = col\_character(),

Age = col\_double(),

SibSp = col\_integer(),

Parch = col\_integer(),

Fare = col\_double(),

Embarked = col\_character()

)

> final\_training\_set\_clean <- read\_csv("final\_training\_set\_clean.csv")

Parsed with column specification:

cols(

Survived = col\_integer(),

Pclass = col\_integer(),

Sex = col\_character(),

Age = col\_double(),

SibSp = col\_integer(),

Parch = col\_integer(),

Fare = col\_double(),

Embarked = col\_character()

)

> test\_clean <- read\_csv("test\_clean.csv")

Parsed with column specification:

cols(

PassengerId = col\_integer(),

Pclass = col\_integer(),

Sex = col\_character(),

Age = col\_double(),

SibSp = col\_integer(),

Parch = col\_integer(),

Fare = col\_double(),

Embarked = col\_character()

)

>

> head(training\_set\_clean,1)

# A tibble: 1 × 8

Survived Pclass Sex Age SibSp Parch Fare Embarked

<int> <int> <chr> <dbl> <int> <int> <dbl> <chr>

1 0 3 male -0.5972816 1 0 -0.5368636 S

>

> #Com s'especifica anteriorment, s'utilitzen 8 variables per l'analisis:

>

> #Survived: Sortida Binaria, 1 vol dir sobreviscut, 0 vol dir no ha sobreviscut.

> #Pclass: Tipus de classe. 1 = Primera, 2 = Segona, 3 = Tercera

> #Sex: Sortida binaria, home o dona.

> #Age: Edat escalada per normalitzar la variable.

> #SibSp: Nombre de germans / esposes a bord

> #Parch: Nombre de pares / fills a bord

> #Fare: Tarifa del tiquet escalada per normalitza la variable.

> #Embarked: lloc d'embarcament

>

> #Pel dataset “test\_clean”, tota la informacio menys la columna survived.

>

> #algunes variales es codifiquen com a factors: “Survived”,“Pclass”,“Sex”, “Embarked”.

>

> str(training\_set\_clean,give.attr = FALSE)

Classes ‘tbl\_df’, ‘tbl’ and 'data.frame': 713 obs. of 8 variables:

$ Survived: int 0 1 1 1 0 0 1 1 1 0 ...

$ Pclass : int 3 1 3 1 3 3 3 3 1 3 ...

$ Sex : chr "male" "female" "female" "female" ...

$ Age : num -0.597 0.671 -0.28 0.433 0.433 ...

$ SibSp : int 1 1 0 1 0 0 0 1 0 0 ...

$ Parch : int 0 0 0 0 0 0 2 1 0 0 ...

$ Fare : num -0.537 0.852 -0.522 0.457 -0.52 ...

$ Embarked: chr "S" "C" "S" "S" ...

>

>

> str(validation\_set\_clean,give.attr = FALSE)

Classes ‘tbl\_df’, ‘tbl’ and 'data.frame': 178 obs. of 8 variables:

$ Survived: int 0 0 1 0 0 1 0 0 0 1 ...

$ Pclass : int 1 3 2 3 2 1 1 3 3 3 ...

$ Sex : chr "male" "male" "female" "female" ...

$ Age : num 1.731 -1.849 -1.023 -1.023 0.423 ...

$ SibSp : int 0 3 1 0 0 0 0 0 2 1 ...

$ Parch : int 0 1 0 0 0 0 0 0 0 0 ...

$ Fare : num 0.3034 -0.1919 -0.0472 -0.4046 -0.1126 ...

$ Embarked: chr "S" "S" "C" "S" ...

> str(final\_training\_set\_clean,give.attr = FALSE)

Classes ‘tbl\_df’, ‘tbl’ and 'data.frame': 891 obs. of 8 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 : chr "male" "female" "female" "female" ...

$ Age : num -0.565 0.663 -0.258 0.433 0.433 ...

$ SibSp : int 1 1 0 1 0 0 0 3 0 1 ...

$ Parch : int 0 0 0 0 0 0 0 1 2 0 ...

$ Fare : num -0.502 0.786 -0.489 0.42 -0.486 ...

$ Embarked: chr "S" "C" "S" "S" ...

> str(test\_clean,give.attr = FALSE)

Classes ‘tbl\_df’, ‘tbl’ and 'data.frame': 418 obs. of 8 variables:

$ PassengerId: int 892 893 894 895 896 897 898 899 900 901 ...

$ Pclass : int 3 3 2 3 3 3 3 2 3 3 ...

$ Sex : chr "male" "female" "male" "male" ...

$ Age : num 0.386 1.37 2.55 -0.205 -0.598 ...

$ SibSp : int 0 1 0 0 1 0 0 1 0 2 ...

$ Parch : int 0 0 0 0 1 0 0 1 0 0 ...

$ Fare : num -0.497 -0.512 -0.464 -0.482 -0.417 ...

$ Embarked : chr "Q" "S" "Q" "S" ...

>

> #Encodifcar com a factor

> training\_set\_clean$Survived <- as.factor(training\_set\_clean$Survived)

> training\_set\_clean$Pclass <- as.factor(training\_set\_clean$Pclass)

> training\_set\_clean$Sex <- as.factor(training\_set\_clean$Sex)

> training\_set\_clean$Embarked <- as.factor(training\_set\_clean$Embarked)

>

> validation\_set\_clean$Survived <- as.factor(validation\_set\_clean$Survived)

> validation\_set\_clean$Pclass <- as.factor(validation\_set\_clean$Pclass)

> validation\_set\_clean$Sex <- as.factor(validation\_set\_clean$Sex)

> validation\_set\_clean$Embarked <- as.factor(validation\_set\_clean$Embarked)

>

> final\_training\_set\_clean$Survived <- as.factor(final\_training\_set\_clean$Survived)

> final\_training\_set\_clean$Pclass <- as.factor(final\_training\_set\_clean$Pclass)

> final\_training\_set\_clean$Sex <- as.factor(final\_training\_set\_clean$Sex)

> final\_training\_set\_clean$Embarked <- as.factor(final\_training\_set\_clean$Embarked)

>

> test\_clean$Pclass <- as.factor(test\_clean$Pclass)

> test\_clean$Sex <- as.factor(test\_clean$Sex)

> test\_clean$Embarked <- as.factor(test\_clean$Embarked)

>

> #1. L'arbre de decisio

>

> #En la part que segueix es crea l'arbre de decisio i es dibuixen els resultats. La decisio a l'esquerra es menys atractiva que la solucio a la dreta pero produeix informacio sensitiva. La variable “Sex” denota ser molt mes important en la prediccio que qualsevol altra variable.

> #La creacio de l'abre de decisio

>

> library(rpart)

> start.time <- Sys.time()

> Dtree <- rpart(Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked,

+ data=training\_set\_clean,

+ method="class")

> Dtree\_pred = predict(Dtree, newdata = validation\_set\_clean[-1], type = 'class')# eliminem amb -1 allo que volem predir, la variable survived.

> end.time <- Sys.time()

> time.takenDT <- end.time - start.time

> time.takenDT

Time difference of 0.03125095 secs

> par(mfrow=c(1,2)) # imprimeix dos caracter horitzontalment

> plot(Dtree)

> text(Dtree)

> library(rattle)

Rattle: A free graphical interface for data mining with R.

Version 4.1.0 Copyright (c) 2006-2015 Togaware Pty Ltd.

Type 'rattle()' to shake, rattle, and roll your data.

>

> library(rpart.plot)

> library(RColorBrewer)

> fancyRpartPlot(Dtree)

>

> #ara es testeja el model amb el set de validacio i creem, la matriu de confusio per definir, la validesa del model. La matriu de confusio en possibilita un analisi de l'exit de la prediccio . En l'eix horitzontal es tenen les classes i en leix vertical es tenen les classes predites.

> #La matriu de confusio per l'arbre

>

> #creracio de la matriu de confusio

> Dtree\_cm = table(t(validation\_set\_clean[, 1]), Dtree\_pred)

> # calculem el rati dexit total

> Dtree\_success = sum(diag(Dtree\_cm))/sum(Dtree\_cm)\*100

> Dtree\_success

[1] 77.52809

> # calculem el rati dexit per predir una supervivencia

> Dtree\_success\_survived = Dtree\_cm[2,2]/(Dtree\_cm[2,1]+Dtree\_cm[2,2])\*100

> Dtree\_success\_survived

[1] 63.23529

> # calcul del rati d'exit per predir la casualitat.

> Dtree\_success\_notsurvived = Dtree\_cm[1,1]/(Dtree\_cm[1,2]+Dtree\_cm[1,1])\*100

> Dtree\_success\_notsurvived

[1] 86.36364

> # prdit en vertical i actual en horitzontal ( 68 sobrevivents en el set de validacio )

> Dtree\_cm

Dtree\_pred

0 1

0 95 15

1 25 43

> #El rati dexit de prediccio es de 77.53% , la prediccio exitosa de la supervivencia es del 63.24% mentres l'exit de prediccio per la no supervivencia es del 86.36%. El model ha tardat 0.03 segons en fer la prediccio.

> #Ara provarem una aproximacio de prediccio des dun model de regresio

>

> #Grupos de hombres y mujeres

> training\_set\_clean.men <- training\_set\_clean[training\_set\_clean$Sex == "male",]

> training\_set\_clean.women <- training\_set\_clean[training\_set\_clean$Sex == "female",]

>

> train1<-training\_set\_clean # Utilitzem el mateix set per entrenar el model que amb el model darbre

> train2<-validation\_set\_clean # Utilitzem el mateix set per validar el mdoel que amb el model darbre

>

>

>

> #testeig de la normalitat

> library(nortest)

> alpha = 0.05

> col.names = colnames(training\_set\_clean)

> for (i in 1:ncol(training\_set\_clean)) {

+ if (i == 1) cat("Variables que no siguen una distribución normal:\n")

+ if (is.integer(training\_set\_clean[,i]) | is.numeric(training\_set\_clean[,i])) {

+ p\_val = ad.test(training\_set\_clean[,i])$p.value

+ if (p\_val < alpha) {

+ cat(col.names[i])

+ # Formateig de la sortida

+ if (i < ncol(training\_set\_clean) - 1) cat(", ")

+ if (i %% 3 == 0) cat("\n")

+ }

+ }

+ }

Variables que no siguen una distribución normal:

>

>

> #Seguidament sestudia la homegeneitat dels sets train i validacio

> library(car)

> fligner.test(Survived ~ Sex, data = training\_set\_clean)

Fligner-Killeen test of homogeneity of variances

data: Survived by Age

Fligner-Killeen:med chi-squared = 1.1626, df = 1, p-value = 0.21

> #Edat mitja dels que van sobreviure i dels que van morir

> aggregate(Age~Survived, data=training\_set\_clean, mean)

Survived Age

1 0 0.05244214

2 1 -0.08402226

>

> t.test(Age ~ Survived, data=training\_set\_clean)

Welch Two Sample t-test

data: Age by Survived

t = 1.7499, df = 552.36, p-value = 0.0807

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-0.01672108 0.28964990

sample estimates:

mean in group 0 mean in group 1

0.05244214 -0.08402226

>

>

>

> # Generación de un modelo de regresion logit

>

> model <- glm(Survived ~.,family=binomial(link='logit'),data=train1)

> summary(model)

Call:

glm(formula = Survived ~ ., family = binomial(link = "logit"),

data = train1)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.5315 -0.6218 -0.4181 0.6180 2.4585

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) 3.07402 0.36235 8.484 < 2e-16 \*\*\*

Pclass2 -0.89844 0.33284 -2.699 0.00695 \*\*

Pclass3 -2.10402 0.33340 -6.311 2.77e-10 \*\*\*

Sexmale -2.72914 0.22279 -12.250 < 2e-16 \*\*\*

Age -0.50377 0.11465 -4.394 1.11e-05 \*\*\*

SibSp -0.27578 0.11522 -2.394 0.01669 \*

Parch -0.04967 0.13450 -0.369 0.71190

Fare -0.01282 0.12670 -0.101 0.91942

EmbarkedQ -0.02165 0.44578 -0.049 0.96127

EmbarkedS -0.60229 0.27299 -2.206 0.02736 \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 946.06 on 710 degrees of freedom

Residual deviance: 628.56 on 701 degrees of freedom

(2 observations deleted due to missingness)

AIC: 648.56

Number of Fisher Scoring iterations: 5

> #Parch, Fare, EmbarkedQ no tenen prou rellevancia >0.05 les eliminem del model

> stepmodel = step(model, direction="both")

Start: AIC=648.56

Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked

Df Deviance AIC

- Fare 1 628.57 646.57

- Parch 1 628.70 646.70

<none> 628.56 648.56

- Embarked 2 634.86 650.86

- SibSp 1 635.11 653.11

- Age 1 649.39 667.39

- Pclass 2 675.80 691.80

- Sex 1 816.57 834.57

Step: AIC=646.57

Survived ~ Pclass + Sex + Age + SibSp + Parch + Embarked

Df Deviance AIC

- Parch 1 628.73 644.73

<none> 628.57 646.57

+ Fare 1 628.56 648.56

- Embarked 2 634.92 648.92

- SibSp 1 635.38 651.38

- Age 1 649.47 665.47

- Pclass 2 694.34 708.34

- Sex 1 816.84 832.84

Step: AIC=644.73

Survived ~ Pclass + Sex + Age + SibSp + Embarked

Df Deviance AIC

<none> 628.73 644.73

+ Parch 1 628.57 646.57

+ Fare 1 628.70 646.70

- Embarked 2 635.20 647.20

- SibSp 1 637.52 651.52

- Age 1 649.59 663.59

- Pclass 2 694.99 706.99

- Sex 1 824.72 838.72

> formula(stepmodel)

Survived ~ Pclass + Sex + Age + SibSp + Embarked

>

> summary(stepmodel)

Call:

glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Embarked,

family = binomial(link = "logit"), data = train1)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.5609 -0.6211 -0.4162 0.6258 2.4604

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) 3.032749 0.326824 9.279 < 2e-16 \*\*\*

Pclass2 -0.887719 0.299357 -2.965 0.00302 \*\*

Pclass3 -2.092034 0.280044 -7.470 8.0e-14 \*\*\*

Sexmale -2.708208 0.216401 -12.515 < 2e-16 \*\*\*

Age -0.501988 0.114154 -4.397 1.1e-05 \*\*\*

SibSp -0.293125 0.107614 -2.724 0.00645 \*\*

EmbarkedQ 0.008495 0.439026 0.019 0.98456

EmbarkedS -0.594440 0.269922 -2.202 0.02765 \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 946.06 on 710 degrees of freedom

Residual deviance: 628.73 on 703 degrees of freedom

(2 observations deleted due to missingness)

AIC: 644.73

Number of Fisher Scoring iterations: 5

>

> #validacion del model amb el set de validacio

>

> pred.train <- predict(model,train2)

> pred.train <- ifelse(pred.train > 0.5,1,0)

> # Mitjana de prediccio encertada

> mean(pred.train==train2$Survived)

[1] 0.7752809

>

> t1<-table(pred.train,train2$Survived)

> # Presicio i sensibilidad del model

> presicion<- t1[1,1]/(sum(t1[1,]))

> recall<- t1[1,1]/(sum(t1[,1]))

> presicion

[1] 0.7692308

> recall

[1] 0.9090909