MAS: Trabalho de Grupo (indicar ID de grupo)

nome completo de representante do grupo

25 de março, 2022

Nota:

Deve efetuar todos os Save com "Save with encoding UTF-8" de modo a manter palavras acentuadas e caracteres especiais**

```
# Remover tudo!
rm(list=ls(all=TRUE))

# Incluir as libraries de que necessita
library(MASS)        # The MASS library contains the Boston data set
library(Metrics)     # To help calculating metrics
library(ggplot2)     # To provide graphics
library(lsr)         # For ETA and Cramer's V measure of association
library(caret)       # Cross-validation + Metrics for classification
library(e1071)       # For classification with Naïve Bayes
library(FNN)         # Implementing KNN - K-Nearest Neighbour
library(car)         # To verify multicolinearity
library(psych)       # For some descriptives
library(nnet)        # For Multinomial Logistic Regression
library(knitr)       # To pretty outputs
library(tree)        # For Classification Tree
```

1. Leitura dos dados Cellular e análise preliminar dos mesmos

1.1) Leitura os dados usando cellular<-read.csv("Cellular.csv", header=TRUE, dec=".", sep=";").Nota: verifique sep no ficheiro de origem.

```
# Leitura do dataset
cellular<-read.csv("Cellular.csv", header=TRUE, dec=".", sep=";")
```

1.2) [1 valor] Apresentação de head(cellular), definição do fator score_r e apresentação da tabela de frequências absolutas correspondente

```
# Primeiras observações do dataset
head(cellular)
```

```
##   minutes  bill business  los income score score_r
## 1  276.46 48.43    28.11 3.50  68.86 64.98       1
## 2  189.01 61.93    22.57 2.42  77.31 52.65       1
## 3  197.49 47.90    27.48 2.42  56.89 63.72       1
## 4  256.77 66.92    44.84 2.34  75.23 72.11       1
## 5  274.82 72.78    37.56 3.38  87.60 83.45       1
## 6  207.29 55.83    36.89 3.18  72.72 70.41       1
```

```
# Definir a variável score_r como factor
cellular$score_r <- factor(cellular$score_r,levels = c(0,1), labels =
c("No churn", "Churn"))

# Tabela de frequências absolutas
table(cellular$score_r)
```

```
##
## No churn    Churn
##      200       50
```

1.3)[0.5 valores] Realização de uma análise descritiva dos dados apresentando o número de observações, mínimo, máximo, média, desvio padrão, medida de assimetria e de achatamento

```
# Análise descritiva e dimensão do dataset "cellular"
summary(cellular)
```

```
##     minutes            bill           business          los
##  Min.   : 53.64   Min.   :  8.01   Min.   : 5.65   Min.   :1.020
##  1st Qu.:131.80   1st Qu.: 49.68   1st Qu.:27.19   1st Qu.:2.290
##  Median :158.47   Median : 63.70   Median :32.43   Median :2.680
##  Mean   :162.19   Mean   : 63.40   Mean   :32.68   Mean   :2.680
##  3rd Qu.:188.84   3rd Qu.: 77.19   3rd Qu.:38.24   3rd Qu.:3.087
##  Max.   :326.25   Max.   :121.24   Max.   :59.23   Max.   :4.370
##      income          score           score_r
##  Min.   :30.15   Min.   :16.71   No churn:200
##  1st Qu.:55.28   1st Qu.:33.03   Churn   : 50
##  Median :60.93   Median :37.91
##  Mean   :61.59   Mean   :41.54
```

```
##  3rd Qu.:68.81   3rd Qu.:44.86
##  Max.   :95.44   Max.   :83.45

dim(cellular)

## [1] 250   7

# Mínimo, máximo, média, desvio padrão, medida de assimetria (skewness) e
de achatamento (kurtosis)
describe(cellular)[c(8,9,3,4,11,12)]

##            min    max   mean    sd  skew kurtosis
## minutes  53.64 326.25 162.19 46.57  0.54     0.38
## bill      8.01 121.24  63.40 19.80  0.02     0.22
## business  5.65  59.23  32.68  9.07  0.09     0.07
## los       1.02   4.37   2.68  0.60 -0.10    -0.12
## income   30.15  95.44  61.59 11.12 -0.06     0.12
## score    16.71  83.45  41.54 13.32  1.02     0.43
## score_r*  1.00   2.00   1.20  0.40  1.49     0.22
```

1.4) [0.5 valores] Divisão dos dados em amostra de treino (65%) e de teste (35%) usando set.seed(888) e apresentação de tabela de frequências absolutas de score_r em cada amostra

```
# Definir o set.seed para permitir reprodutibilidade dos resultados
set.seed(888)

# Divisão em Conjunto Treino/Teste
ind_train <- sample(nrow(cellular),0.65*nrow(cellular))

# Conjunto Treino (cellular_train)
cellular_train <- cellular[ind_train,]
paste("O Conjunto de Treino tem", nrow(cellular_train),"observações.")

## [1] "O Conjunto de Treino tem 162 observações."

# Tabela de frequências relativas da variável score_r - conjunto treino
table(cellular_train$score_r)

##
## No churn    Churn
##      125       37

# Conjunto Teste (cellular_test)
cellular_test <- cellular[-ind_train,]
paste("O Conjunto de Teste tem", nrow(cellular_test),"observações.")

## [1] "O Conjunto de Teste tem 88 observações."

# Tabela de frequências relativas da variável score_r - conjunto teste
table(cellular_test$score_r)
```

```
##
## No churn    Churn
##       75       13
```

1.5) [0.5 valores] Obtenção dos dados dos preditores normalizados (normalização 0-1), nas amostras de treino e teste, e apresentação das primeiras 6 linhas destas amostras após normalização

```
# Função de normalização (0-1)
normalize <- function(x){
  return ((x -min(x)) / (max(x)-min(x)))
  }

# Conjunto de Treino Normalizado (0-1)
cellular_train_norm <- cellular_train
cellular_train_norm[,1:6] <-sapply(cellular_train[,1:6],normalize)
head(cellular_train_norm)
```

```
##       minutes      bill  business       los    income      score  score_r
## 136 0.3191739 0.6275222 0.8875536 0.4925373 0.5556563 0.27344598 No churn
## 193 0.3808738 0.3902649 0.5763948 0.4388060 0.1774837 0.51340877 No churn
## 175 0.2124280 0.3099359 0.5409871 0.4208955 0.2554387 0.29607813 No churn
## 11  0.5406258 0.4396098 0.2877682 0.3611940 0.4033720 0.95458069    Churn
## 107 0.2825648 0.2973128 0.7733906 0.8298507 0.2902466 0.22229112 No churn
## 167 0.4029566 0.1892512 0.4113734 0.2119403 0.2610587 0.09409394 No churn
```

```
# Conjunto de Teste Normalizado (0-1)
cellular_test_norm <- cellular_test
cellular_test_norm[,1:6] <-sapply(cellular_test[,1:6],normalize)
head(cellular_test_norm)
```

```
##      minutes      bill  business       los    income     score score_r
## 4  0.9133545 0.5350104 0.7314296 0.3591549 0.6904580 0.8168309   Churn
## 5  1.0000000 0.5882300 0.5955580 0.7253521 0.8799204 1.0000000   Churn
## 12 0.5760849 0.7461629 0.3296006 0.2746479 0.4480012 0.6478759   Churn
## 13 0.7873464 0.5707928 0.6767451 0.4401408 0.4650023 0.8387983   Churn
## 21 0.7603687 0.7874852 0.5696155 0.4190141 0.7017920 0.5696979   Churn
## 23 0.9863191 0.4944147 0.7472938 0.4119718 0.3637617 0.5193022   Churn
```

1.6) [1 valor] Completação das frases seguintes em comentário do script (com eventual obtenção de resultados adicionais):

```
#A dimensão de "Cellular.csv" é de ______ número de linhas e ________
número de colunas; na amostra original encontram-se ______ casos com
score_r="No churn" e no conjunto de teste esta categoria corresponde a
_____ % das observações.

# 1 - 250
# 2 - 7
# 3 - 200
# 4 - 75
```

A dimensão de "Cellular.csv" é de **250** número de linhas e **7** número de colunas; na amostra original encontram-se **200** casos com score_r="No churn" e no conjunto de teste esta categoria corresponde a **75** % das observações.

2. Aprendizagem, sobre a amostra de treino, do 3-Nearest Neighbour (baseado em dois preditores) para prever score_r e avaliação do seu desempenho

2.1) [1.5 valores] Escolha dos preditores, justificando

```
# Associação entre o target categórico e os preditores métricos
eta<- matrix(0,6,1)
rownames(eta)<-colnames(cellular[,1:6])

for (i in 1:6) {
  anova_ <- aov (cellular[,i] ~ score_r, cellular) # numbers for levels
(not strings)
  eta[i-2]<-sqrt(etaSquared(anova_ )[,1])
  }
eta
```

```
##              [,1]
## minutes  0.1417491
## bill     0.1073271
## business 0.1926101
## los      0.8695291
## income   0.6734982
## score    0.6734982
```

```
# Correlação entre preditores
cor(cellular[1:6])
```

```
##            minutes      bill  business        los    income      score
## minutes  1.0000000 0.4777417 0.3473440 0.31449095 0.3326367 0.60764629
## bill     0.4777417 1.0000000 0.5049000 0.30303450 0.2129874 0.31171841
## business 0.3473440 0.5049000 1.0000000 0.30911095 0.2316232 0.15209559
## los      0.3144910 0.3030345 0.3091110 1.00000000 0.2408207 0.09872305
## income   0.3326367 0.2129874 0.2316232 0.24082072 1.0000000 0.23757218
## score    0.6076463 0.3117184 0.1520956 0.09872305 0.2375722 1.00000000
```

Escolhi os preditores ***los*** e ***income*** para que sejam fortemente associadas ao target e não entre preditores de modo a evitar problemas de multicolinearidade

2.2) [2 valores] Obtenção do modelo e das correspondentes estimativas de score_r sobre amostra de teste

```
# Modelo de KNN com k=3
knn <- knn(cellular_train_norm[,4:5], cellular_test_norm[,4:5],
cellular_train_norm$score_r, k=3,prob = TRUE)
```

2.3) [2 valores] Apresentação da Confusion matrix sobre amostra de teste e do índice de Huberty correspondente

```
# Matriz de Classificação para o conjunto de teste
table(knn ,cellular_test_norm$score_r)

##
## knn         No churn Churn
##   Churn            4     2
##   No churn        71    11

# Accuracy
accuracy <- mean(knn== cellular_test_norm$score_r)

# Índice de Huberty
default_p <- max(mean(cellular_test_norm == "No Churn"),
mean(cellular_test_norm == "Churn")) # majorit y class frequency
(Huberty<-(accuracy-default_p)/(1-default_p))

## [1] 0.8258706
```

2.4) [2 valores] Completação das frases seguintes em comentário do script (com eventual obtenção de resultados adicionais):

```
#Na aprendizagem foram usados dados ___________(normalizados/ não
normalizados); as observações mais próximas da primeira observação do
conjunto de teste são_____________ (números das observações); a
probabilidade da última observação do conjunto de teste pertencer à
classe alvo "No churn", estimada pelo modelo, é_____________;
segundo os resultados estimados, o churn dos clientes na amostra de teste
será _____________%.

# 1 - normalizados
# 2 - 146  10  20
# 3 - 1
# 4 -
```

Na aprendizagem foram usados dados **normalizados**; as observações mais próximas da primeira observação do conjunto de teste são **`r attr(knn,"nn.index")[1,]`** ; a probabilidade da última observação do conjunto de teste pertencer à classe alvo "No churn", estimada pelo modelo, é **1**; segundo os resultados estimados, o churn dos clientes na amostra de teste será **0.07** %.

3. Aprendizagem, sobre a amostra de treino, de uma Árvore de Regressão para prever score e avaliação do seu desempenho

3.1) [1.5 valores] Obtenção do modelo, com cerca de 10 nós folha, e apresentação da árvore correspondente

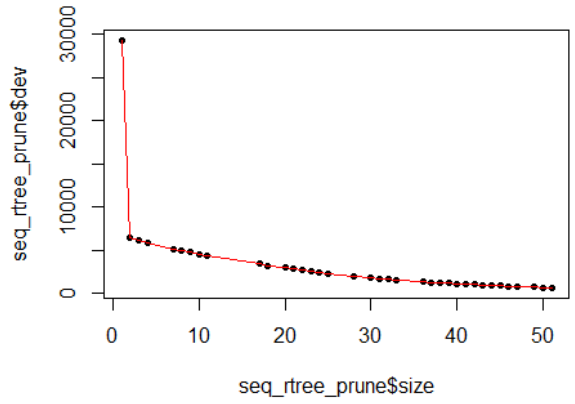
```
# Começamos por criar uma Árvore Grande
rtree_large <-tree(score~. ,data=cellular_train,
                           control=tree.control(nrow(cellular_train),
                                                mincut = 1,
                                                minsize = 2,
                                                mindev = 0.001),
                           split = "deviance")

# Resultados da Árvore
summary(rtree_large)

##
## Regression tree:
## tree(formula = score ~ ., data = cellular_train, control =
tree.control(nrow(cellular_train),
##     mincut = 1, minsize = 2, mindev = 0.001), split = "deviance")
## Number of terminal nodes:  51
## Residual mean deviance:  6.136 = 681.1 / 111
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -5.085  -1.292   0.000   0.000   1.225   6.374

### Custo-Complexidade - Poda da Árvore

# Gráfico de Custo/Complexidade
seq_rtree_prune <- prune.tree(rtree_large)
plot(seq_rtree_prune$size,seq_rtree_prune$dev,pch =20)
lines(seq_rtree_prune$size,seq_rtree_prune$dev, col = "red")
```



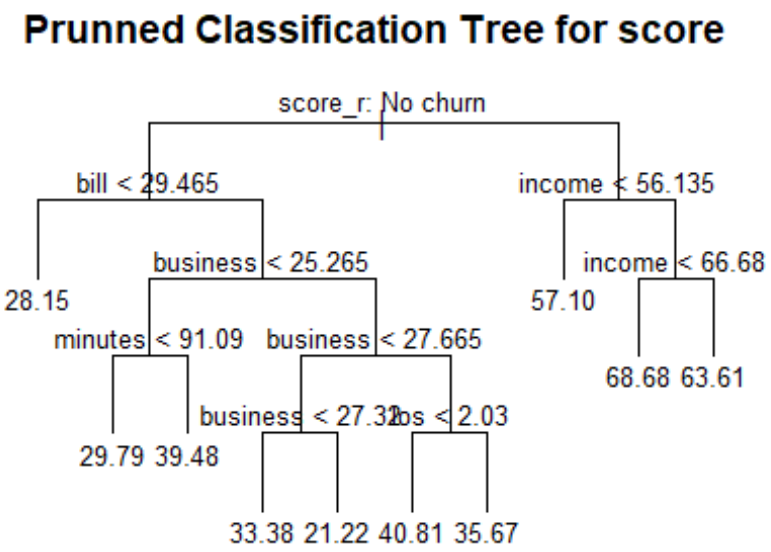
```r
# Utilizando o "melhor" tamanho de 15 como referido no enunciado,
obtermos a seguinte Árvore Podada
rtree.cellular<-prune.tree(rtree_large, best=10)
summary(rtree.cellular)
```

```
##
## Regression tree:
## snip.tree(tree = rtree_large, nodes = c(20L, 21L, 46L, 15L, 4L,
## 47L, 14L))
## Number of terminal nodes:  10
## Residual mean deviance:  30.36 = 4615 / 152
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -16.07000  -3.20700   0.09735   0.00000   3.36000  14.16000
```

```r
# a) Representações da Árvore de Classificação - Lista indentada
rtree.cellular
```

```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 162 29300.000 42.41
##    2) score_r: No churn 125  5053.000 35.96
##      4) bill < 29.465 5   228.300 28.15 *
##      5) bill > 29.465 120  4507.000 36.29
##       10) business < 25.265 27   763.300 38.40
##         20) minutes < 91.09 3    86.120 29.79 *
##         21) minutes > 91.09 24   427.100 39.48 *
##       11) business > 25.265 93  3589.000 35.68
##         22) business < 27.665 7   230.600 29.91
##           44) business < 27.32 5    14.740 33.38 *
##           45) business > 27.32 2     4.836 21.22 *
##         23) business > 27.665 86  3106.000 36.15
##           46) los < 2.03 8   273.300 40.81 *
##           47) los > 2.03 78  2641.000 35.67 *
##    3) score_r: Churn 37  1490.000 64.20
##      6) income < 56.135 6    20.890 57.10 *
##      7) income > 56.135 31  1109.000 65.57
##       14) income < 66.68 12   444.400 68.68 *
##       15) income > 66.68 19   475.100 63.61 *
```

```r
# b) Representações da Árvore de Classificação - Gráfico da Árvore
plot(rtree.cellular, type="uniform")
text(rtree.cellular, pretty =0, cex=0.8)
title(main = "Prunned Classification Tree for score")
```



3.2) [1.5 valores] Estimação de score sobre amostra de teste, a partir da árvore obtida, e apresentação das estimativas correspondentes às 6 primeiras observações desta amostra

```
# Estimação de score sobre amostra de teste
pred_ctree.cellular_test <- predict(rtree.cellular, cellular_test)

# 6 primeiras observações
head(pred_ctree.cellular_test)

##        4        5       12       13       21       23
## 63.60579 63.60579 68.67917 68.67917 63.60579 57.10000
```

3.3) [1.5 valores] Apresentação de 3 métricas de regressão associadas ao modelo aplicado sobre a amostra de teste

```
# Accuracy sobre cellular_test
confusion_mat_tree_test <- table(cellular_test$score,
pred_ctree.cellular_test)
(accuracy.test <-
sum(diag(confusion_mat_tree_test))/sum(confusion_mat_tree_test))

## [1] 0.02272727

# R-Squared
RSS <-sse(cellular_test$score, pred_ctree.cellular_test)
(RSQ <-1-RSS/sse(cellular_test$score,mean(cellular_test$score)))
```

```
## [1] 0.6277124

# MAPE Test | Erro de Previsão
actual2 <- cellular_test$score
n<-length(cellular_test$score)
MAPE2 <- (1/n) * sum(abs((actual2 - pred_ctree.cellular_test)/actual2))
MAPE2

## [1] 0.1747792
```

3.4) [1 valor] Apresentação, com base nas estimativas obtidas em 3.2), de uma tabela de frequências para as categorias Churn e No churn

```
# Vetor de categorias com base nas estimativas de score
table(ifelse(pred_ctree.cellular_test > 60, "Churn", "No churn"))

##
##    Churn No churn
##       11       77
```

3.5) [2 valores] Completação das frases seguintes em comentário do script (com eventual obtenção de resultados adicionais):

```
#Na aprendizagem foram usados dados ________ (normalizados/ não
normalizados);  o R-Square associado ao modelo sobre o teste
é_____________; o nó folha com menor frequência inclui
_____________observações do teste;  segundo os resultados estimados, a
% de observações da amostra de teste suscetíveis de fazer churn será
_____________.

# 1 - não normalizados
# 2 - 0.6277124
# 3 - 45)
# 4 - 11/77 = 12.5%
```

Na aprendizagem foram usados dados **normalizados**; o R-Square associado ao modelo sobre o teste é **0.6277124**; o nó folha com menor frequência inclui **45)** observações do teste; segundo os resultados estimados, a % de observações da amostra de teste suscetíveis de fazer churn será **12.5**.