MAS: Exame 2ª Época - Parte II (Prática do R)

André Filipe Gomes Silvestre

20 de junho, 2023

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))# Remove everything!
# Incluir as libraries de que necessita

library(HSAUR2)# para obter dados CHFLS
library(psych)
library(lsr)
library(nnet) # for Multinomial Logistic Regression
library(tree)
```

1. Os dados CHFLS

1.1) Leitura dos dados CHFLS

```
data("CHFLS")

# O estudo realizado na China sobre Saúde e Vida Familiar efectuou uma
amostragem sobre aldeias e bairros urbanos de forma a representar toda a
gama geográfica e socioeconômica da China contemporânea.
#NOTA: para mais detalhes sobre este data set consulte https://search.r-
project.org/CRAN/refmans/HSAUR2/html/CHFLS.html

dim(CHFLS)

## [1] 1534   10

str(CHFLS)

## 'data.frame':    1534 obs. of  10 variables:
##  $ R_region: Factor w/ 6 levels "Coastal South",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ R_age   : num  54 46 48 46 45 36 48 36 20 30 ...
##  $ R_edu   : Ord.factor w/ 6 levels "Never attended school"<..: 4 4 4 3 3 4 3 3 3 4
...
##  $ R_income: num  900 500 800 300 300 500 0 100 200 400 ...
##  $ R_health: Ord.factor w/ 5 levels "Poor"<"Not good"<..: 4 3 4 3 3 5 2 4 3 4 ...
##  $ R_height: num  165 156 163 164 162 161 167 156 158 160 ...
##  $ R_happy : Ord.factor w/ 4 levels "Very unhappy"<..: 3 3 3 3 3 3 4 2 2 3 ...
```

```
##  $ A_height: num  172 170 172 174 172 180 168 173 178 176 ...
##  $ A_edu   : Ord.factor w/ 6 levels "Never attended school"<..: 4 4 3 2 3 5 3 4 5 4
...
##  $ A_income: num  500 800 700 700 400 900 300 800 200 600 ...
```

```
# NOTA: usamos factor() para considerar como tal a variável R_happy, já
que não iremos, nesta análise, considerar a sua ordem
str(CHFLS$R_happy)
```

```
##  Ord.factor w/ 4 levels "Very unhappy"<..: 3 3 3 3 3 3 4 2 2 3 ...
```

```
CHFLS$R_happy<- factor(CHFLS$R_happy , ordered = FALSE)
str(CHFLS$R_happy)
```

```
##  Factor w/ 4 levels "Very unhappy",..: 3 3 3 3 3 3 4 2 2 3 ...
```

1.2) [1.5 valores] Sumário de CHFLS e apresentação tabela de frequências relativas (com 3 c.d.) da variável R_happy

```
# Sumário dos dados CHFLS
summary(CHFLS)
```

```
##           R_region       R_age                          R_edu
##  Coastal South:319   Min.   :20.00   Never attended school: 90
##  Coastal Easth:331   1st Qu.:32.00   Elementary school    :267
##  Inlands      :156   Median :38.00   Junior high school   :583
##  North        :241   Mean   :38.99   Senior high school   :425
##  Northeast    :279   3rd Qu.:45.00   Junior college       :125
##  Central West :208   Max.   :64.00   University           : 44
##
##     R_income          R_health      R_height               R_happy
##  Min.   :    0   Poor     : 10   Min.   :140.0   Very unhappy  :  14
##  1st Qu.:  200   Not good :139   1st Qu.:156.0   Not too happy : 185
##  Median :  500   Fair     :461   Median :160.0   Somewhat happy:1055
##  Mean   :  617   Good     :582   Mean   :159.3   Very happy    : 280
##  3rd Qu.:  800   Excellent:342   3rd Qu.:162.8
##  Max.   :10000                   Max.   :178.0
##
##     A_height                        A_edu        A_income
##  Min.   :155.0   Never attended school: 30   Min.   :    0.0
##  1st Qu.:168.0   Elementary school    :204   1st Qu.:  400.0
##  Median :170.0   Junior high school   :587   Median :  700.0
##  Mean   :171.2   Senior high school   :464   Mean   :  986.7
##  3rd Qu.:175.0   Junior college       :146   3rd Qu.: 1000.0
##  Max.   :190.0   University           :100   Max.   :10000.0
##                  NA's                 :  3
```

```
# Tabela de frequências relativas
round(prop.table(table(CHFLS$R_happy)), 3)
```

```
##
##   Very unhappy  Not too happy Somewhat happy     Very happy
##          0.009          0.121          0.688          0.183
```

1.3) [1.5 valores] Apresentação, para as variáveis R_income e A_income, dos valores mínimo, máximo, média e desvio padrão, assimetria e curtose (apresentados por esta ordem).

```
# Mínimo, máximo, média e desvio padrão, assimetria e curtose das
variáveis R_income e A_income
describe(CHFLS[,c("R_income", "A_income")])[c(8,9,3,4,11,12)]
```

```
##          min   max   mean      sd skew kurtosis
## R_income   0 10000 617.00  749.76 5.25    45.95
## A_income   0 10000 986.69 1195.96 4.41    25.98
```

1.4) [1 valor] Completação das frases seguintes em comentário do script:

```
#Os dados CHFLS são compostos por ________ observações e por ________
variáveis; a média da variável A_income é  ______; o enviesamento de
R_income é _______ (inferior/superior) ao de A_income.

# 1 - 1534      nrow(CHFLS)
# 2 - 10        ncol(CHFLS)
# 3 - 986.69    round(mean(CHFLS$A_income), 2)
# 4 - superior  (pq skew maior)
```

Os dados CHFLS são compostos por **1534** observações e por **10** variáveis; a média da variável A_income é **986.69**; o enviesamento de R_income é **superior** ao de A_income.

2. Aprendizagem, sobre os dados CHFLS de Regressão Logística (considerando R_income e um preditor qualitativo) para prever R_happy e avaliação do seu desempenho

2.1) [2 valores] Escolha do preditor qualitativo mediante associação com o alvo

```
# Medir a associação entre preditores qualitativos R_region, R_edu,
R_health e A_edu e o target qualitativo R_happy – V de Cramer
(cramersV(CHFLS$R_region,CHFLS$R_happy))
```

```
## [1] 0.09190743
```

```
(cramersV(CHFLS$R_edu,CHFLS$R_happy))
```

```
## [1] 0.06985263
```

```
(cramersV(CHFLS$R_health,CHFLS$R_happy))
```

```
## [1] 0.2761125
```

```
(cramersV(CHFLS$A_edu,CHFLS$R_happy))
```

```
## [1] 0.07856557
```

Com base nos valores de associação de *V de Cramer* obtidos, a variável qualitativa que mais se correlaciona com o *target* é **R_health**, pelo que escolho esta variável como 2º preditor para o modelo.

2.2) [2 valores] Obtenção do modelo considerando "Very unhappy" como categoria de referência; sumário do modelo obtido.

```
# Considerando "Very unhappy" como categoria de referência
CHFLS$R_happy <- relevel(CHFLS$R_happy, ref = "Very unhappy")

# Obtendo o modelo de regressão logística multinomial
rlog.R_happy <- multinom(R_happy ~ R_income + R_health, data = CHFLS)
```

```
## # weights:  28 (18 variable)
## initial  value 2126.575550
## iter  10 value 1366.123802
## iter  20 value 1186.797818
## iter  30 value 1185.274666
## iter  40 value 1185.195104
## iter  50 value 1185.194166
## final  value 1185.193971
## converged
```

```r
# Obtendo o sumário do Modelo de Regressão Logística produzido
summary(rlog.R_happy)
```

```
## Call:
## multinom(formula = R_happy ~ R_income + R_health, data = CHFLS)
##
## Coefficients:
##                (Intercept)     R_income R_health.L R_health.Q R_health.C
## Not too happy     3.450792 0.0004207301   5.766589   2.169526  1.4576302
## Somewhat happy    4.436129 0.0009244590   7.633566   1.393015  0.9669483
## Very happy        2.913182 0.0010472334   8.361829   2.972128  1.0357330
##                R_health^4
## Not too happy   0.4699834
## Somewhat happy  0.4282686
## Very happy      0.5813625
##
## Std. Errors:
##                (Intercept)     R_income R_health.L R_health.Q R_health.C
## Not too happy   0.08452347 0.0005841321 0.10250627 0.10275349  0.1250971
## Somewhat happy  0.06167665 0.0005714066 0.08092369 0.07039265  0.1012626
## Very happy      0.07458647 0.0005731120 0.10940005 0.08409352  0.1371049
##                R_health^4
## Not too happy  0.11794554
## Somewhat happy 0.09558495
## Very happy     0.13348284
##
## Residual Deviance: 2370.388
## AIC: 2406.388
```

2.3) [3 valores] Apresentação da Confusion matrix e dos correspondentes número e percentagem de casos corretamente classificados; estimativas das probabilidades de pertença às classes alvo associadas às primeiras 6 observações de CHFLS

```r
# Obtenção das previsões do modelo
predictions <- predict(rlog.R_happy, CHFLS, type = "class")

# Matriz de Confusão
(confusion_matrix <- table(CHFLS$R_happy, predictions))
```

```
##                  predictions
##                   Very unhappy Not too happy Somewhat happy Very happy
##   Very unhappy               0             2             12          0
##   Not too happy              0             4            178          3
##   Somewhat happy             0             3           1041         11
##   Very happy                 0             1            266         13
```

```r
# Número de casos corretamente classificados
correctly_classified <- sum(diag(confusion_matrix))

# Percentagem de casos corretamente classificados
percentage_correct <- correctly_classified / sum(confusion_matrix) * 100
```

```
cat("Número de casos corretamente classificados:",
round(correctly_classified,2))

## Número de casos corretamente classificados: 1058

cat("Percentagem de casos corretamente classificados:",
round(percentage_correct,2), "%")

## Percentagem de casos corretamente classificados: 68.97 %

# Estimativas das probabilidades de pertença às classes alvo para as
primeiras 6 observações
probabilities <- predict(rlog.R_happy, CHFLS, type = "probs")
head(probabilities, 6)

##    Very unhappy Not too happy Somewhat happy Very happy
## 2  1.203684e-03    0.06106752      0.7954216 0.14230716
## 3  8.749926e-03    0.14954306      0.7574315 0.08427554
## 10 1.318242e-03    0.06412403      0.7942025 0.14035522
## 11 1.036540e-02    0.16285604      0.7458090 0.08096956
## 22 1.036540e-02    0.16285604      0.7458090 0.08096956
## 23 9.686916e-06    0.07732877      0.4882521 0.43440942
```

2.4) [1 valor] Completação das frases seguintes em comentário do script

```
# A accuracy obtida pelo modelo de regressão logística multinomial é
________; o total de observações corretamente classificadas é _______
(número de observações); a probabilidade da primeira observação pertencer
à classe alvo "Very happy", estimada pelo modelo, é _____________; a
sexta observação é classificada em ________.

# 1 - round(correctly_classified/nrow(CHFLS),2)
# 2 - correctly_classified
# 3 - 0.14230716
# 4 - Somewhat happy
```

A accuracy obtida pelo modelo de regressão logística multinomial é **0.69**; o total de observações corretamente classificadas é **1058** ; a probabilidade da primeira observação pertencer à classe alvo *"Very happy"*, estimada pelo modelo, é **0.14230716**; a sexta observação é classificada em ***Somewhat happy***.

3. Aprendizagem, de uma Árvore de Classificação para prever R_happy (usando todos os preditores disponíveis) e avaliação do seu desempenho

3.1) [1 valor] Divisão dos dados em amostra de treino (70%) e de teste (30%) usando set.seed(123) e apresentação de tabela de frequências relativas da variável R_happy em cada amostra

```
#####
# Renomear os níveis de R_happy para facilitar a visualização da árvore
levels(CHFLS$R_happy)

## [1] "Very unhappy"   "Not too happy"  "Somewhat happy" "Very happy"

levels(CHFLS$R_happy)<-c("v.unhappy","n.t.happy","s.happy","v.happy")
#####

# Definir o set.seed para permitir reprodutibilidade dos resultados
set.seed(123)

# Divisão em Conjunto Treino/Teste (70/30)
ind_train <- sample(nrow(CHFLS),0.7*nrow(CHFLS))

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

## [1] "O Conjunto de Treino tem 1073 observações."

# Tabela de frequências relativas da variável R_happy - Conjunto de
Treino
prop.table(table(CHFLS_train$R_happy))

##   v.unhappy   n.t.happy     s.happy     v.happy
## 0.009319664 0.124883504 0.675675676 0.190121156

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

## [1] "O Conjunto de Teste tem 461 observações."

# Tabela de frequências relativas da variável R_happy - Conjunto de Teste
prop.table(table(CHFLS_test$R_happy))

##  v.unhappy  n.t.happy    s.happy    v.happy
## 0.00867679 0.11062907 0.71583514 0.16485900
```

3.2) [2 valores] Considere a árvore ctree_large.CHFLS. obtenha (a partir desta árvore, sobre a amostra de treino) uma árvore podada com 15 nós folha e apresente-a em formato lista indentada e o summary correspondente

```
ctree_large.CHFLS<-tree(R_happy~. ,
                        data=CHFLS_train,
                        control=tree.control(nrow(CHFLS_train),
                                             mincut = 10,
                                             minsize = 20,
                                             mindev = 0.001),
                        split = "deviance")

# Utilizando o tamanho de 15 como referido no enunciado, obtermos a seguinte Árvore Podada
ctree.CHFLS <- prune.tree(ctree_large.CHFLS, best=15)

# Representação da Árvore de Classificação em Lista indentada
ctree.CHFLS
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 1071 1892.00 s.happy ( 0.009337 0.124183 0.676004 0.190476 )
##    2) R_health: Poor,Not good,Fair,Good 839 1362.00 s.happy ( 0.011919 0.137068 0.729440 0.121573 )
##      4) R_health: Poor,Not good 106  221.00 s.happy ( 0.066038 0.339623 0.537736 0.056604 )
##        8) A_edu: Never attended school,Elementary school,Junior high school,Junior college 71  162.50 s.happy ( 0.098592 0.309859 0.507042 0.084507 )
##         16) R_income < 450 51  120.80 n.t.happy ( 0.117647 0.411765 0.392157 0.078431 )
##           32) R_edu: Never attended school,Junior high school 23   47.65 n.t.happy ( 0.043478 0.608696 0.217391 0.130435 ) *
##           33) R_edu: Elementary school,Senior high school 28   62.02 s.happy ( 0.178571 0.250000 0.535714 0.035714 ) *
##         17) R_income > 450 20   28.33 s.happy ( 0.050000 0.050000 0.800000 0.100000 ) *
##        9) A_edu: Senior high school,University 35   47.11 s.happy ( 0.000000 0.400000 0.600000 0.000000 ) *
##      5) R_health: Fair,Good 733 1084.00 s.happy ( 0.004093 0.107776 0.757162 0.130969 )
##       10) R_health: Fair 329  502.40 s.happy ( 0.006079 0.151976 0.744681 0.097264 )
##         20) A_income < 350 58   61.72 s.happy ( 0.000000 0.224138 0.775862 0.000000 ) *
##         21) A_income > 350 271  425.20 s.happy ( 0.007380 0.136531 0.738007 0.118081 )
##           42) R_income < 550 139  261.80 s.happy ( 0.014388 0.187050 0.647482 0.151079 )
##             84) A_edu: Never attended school,Elementary school,Junior high school,Senior high school,University 125  217.70 s.happy ( 0.000000 0.176000 0.664000 0.160000 ) *
##             85) A_edu: Junior college 14   32.79 s.happy ( 0.142857 0.285714 0.500000 0.071429 ) *
##           43) R_income > 550 132  149.40 s.happy ( 0.000000 0.083333 0.833333 0.083333 ) *
##       11) R_health: Good 404  564.80 s.happy ( 0.002475 0.071782 0.767327 0.158416 )
##         22) R_region: Coastal South,Coastal Easth 190  222.80 s.happy ( 0.000000 0.026316 0.794737 0.178947 )
##           44) R_edu: Never attended school,Junior high school,Senior high school,University 143  135.60 s.happy ( 0.000000 0.000000 0.818182 0.181818 ) *
##           45) R_edu: Elementary school,Junior college 47   72.76 s.happy ( 0.000000 0.106383 0.723404 0.170213 ) *
##         23) R_region: Inlands,North,Northeast,Central West 214  328.10 s.happy ( 0.004673 0.112150 0.742991 0.140187 )
##           46) A_edu: Never attended school,Elementary school,Junior high school,Senior high school 180  278.70 s.happy ( 0.000000 0.133333 0.727778 0.138889 ) *
##           47) A_edu: Junior college,University 34   37.09 s.happy ( 0.029412 0.000000 0.823529 0.147059 ) *
##    3) R_health: Excellent 232  422.80 s.happy ( 0.000000 0.077586 0.482759 0.439655 )
##      6) A_edu: Never attended school,Senior high school,University 83  144.00 v.happy ( 0.000000 0.072289 0.337349 0.590361 )
##       12) R_height < 158.5 32   38.52 v.happy ( 0.000000 0.062500 0.125000 0.812500 ) *
##       13) R_height > 158.5 51   93.18 s.happy ( 0.000000 0.078431 0.470588 0.450980 ) *
##      7) A_edu: Elementary school,Junior high school,Junior college 149  266.30 s.happy ( 0.000000 0.080537 0.563758 0.355705 ) *
```

```
# Sumário da Árvore produzida
summary(ctree.CHFLS)
```

```
##
## Classification tree:
## snip.tree(tree = ctree_large.CHFLS, nodes = c(20L, 9L, 33L, 44L,
## 12L, 17L, 47L, 45L, 84L, 32L, 13L, 7L, 43L, 46L))
## Variables actually used in tree construction:
## [1] "R_health" "A_edu"    "R_income" "R_edu"    "A_income" "R_region"
"R_height"
## Number of terminal nodes:  15
## Residual mean deviance:  1.486 = 1569 / 1056
## Misclassification error rate: 0.2951 = 316 / 1071
```

3.3) [2 valores] A partir da árvore obtida e considerando a amostra de treino: estimação de R_happy e apresentação da correspondente matriz de classificação e % de casos incorretamente classificados

```
# Estimação de R_happy usando a árvore obtida na amostra de treino
predicted_train <- predict(ctree.CHFLS, newdata = CHFLS_train, type =
"class")

# Matriz de Classificação
(confusion_matrix_train <- table(CHFLS_train$R_happy, predicted_train))
```

```
##             predicted_train
##              v.unhappy n.t.happy s.happy v.happy
##   v.unhappy          0         1       9       0
##   n.t.happy          0        14     118       2
##   s.happy            0         5     716       4
##   v.happy            0         3     175      26
```

```
# Número de casos incorretamente classificados
incorrectly_classified_train <- sum(confusion_matrix_train) -
sum(diag(confusion_matrix_train))

# Percentagem de casos incorretamente classificados
percentage_incorrect_train <- incorrectly_classified_train /
sum(confusion_matrix_train) * 100

cat("Percentagem de casos incorretamente classificados
(treino):",round(percentage_incorrect_train,2), "%")
```

```
## Percentagem de casos incorretamente classificados (treino): 29.54 %
```

3.4) [2 valores] A partir da árvore obtida e considerando a amostra de teste: estimação de R_happy e apresentação da correspondente matriz de classificação e % de casos incorretamente classificados

```
# Estimação de R_happy usando a árvore obtida na amostra de teste
predicted_test <- predict(ctree.CHFLS, newdata = CHFLS_test, type =
"class")

# Matriz de Classificação
(confusion_matrix_test <- table(CHFLS_test$R_happy, predicted_test))
```

```
##             predicted_test
##              v.unhappy n.t.happy s.happy v.happy
##   v.unhappy          0         0       4       0
##   n.t.happy          0         4      46       1
##   s.happy            0         6     316       8
##   v.happy            0         0      72       4
```

```
# Número de casos incorretamente classificados
incorrectly_classified_test <- sum(confusion_matrix_test) -
sum(diag(confusion_matrix_test))

# % de casos incorretamente classificados
percentage_incorrect_test <- incorrectly_classified_test /
sum(confusion_matrix_test) * 100

cat("Percentagem de casos incorretamente classificados (teste):",
percentage_incorrect_test, "%")
```

```
## Percentagem de casos incorretamente classificados (teste): 29.718 %
```

3.5) [1 valor] Completação das frases seguintes em comentário do script (com eventual obtenção de resultados adicionais):

```
# A Árvore de Classificação é constituída por ______ nós folha; sobre a
amostra de treino a Deviance inicial é _______ e a Residual Deviance é
___________; a percentagem de casos incorretamente classificados nas
amostras de treino e teste _________ (indica/ não indica) overfitting.
```

A Árvore de Classificação é constituída por **15** nós folha; sobre a amostra de treino a Deviance inicial é **1892** e a Residual Deviance é **1569**; a percentagem de casos incorretamente classificados nas amostras de treino e teste **não indica** *overfitting*.