Format d´entrega

Aquest document s´ha realitzat mitjançant **Markdown**[1] amb l´ajuda de l´entorn de desenvolupament **RStudio**[2] utilitzant les característiques que aquest ofereix per a la creació de documents **R** reproduïbles.

La documentació generada en la realització de la pràctica es troba allotjada en **GitHub** al següent repositori:

- **https://github.com/rsanchezs/data-minig**

En aquest repositori es poden trobar els següents fitxers:

- Aquest document en formats **pdf** i **docx** amb el nom rsanchezs_practica.
- Un document **R Markdown**[3] que es pot utilitzar per a reproduir tots els exemples presentats a la PAC.
- El conjunt de dades utilitzades.

Nota: Propietat intel·lectual

Sovint és inevitable, al produir una obra multimèdia, fer ús de recursos creats per terceres persones. És per tant comprensible fer-lo en el marc d'una pràctica dels Estudis, sempre que això es documenti clarament i no suposi plagi en la pràctica.

[1] **https://es.wikipedia.org/wiki/Markdown**

[2] **https://www.rstudio.com/**

[3] **https://rmarkdown.rstudio.com/**

Per tant, al presentar una pràctica que faci ús de recursos aliens, s'ha de presentar juntament amb ella un document en que es detallin tots ells, especificant el nom de cada recurs, el seu autor, el lloc on es va obtenir i el seu estatus legal: si l'obra està protegida pel copyright o s'acull a alguna altra llicència d'ús (Creative Commons, llicència GNU, GPL ...). L'estudiant haurà d'assegurar-se que la llicència no impedeix específicament el seu ús en el marc de la pràctica. En cas de no trobar la informació corresponent haurà d'assumir que l'obra està protegida per copyright. Hauríeu a més, d'adjuntar els fitxers originals quan les obres utilitzades siguin digitals, i el seu codi font si correspon

Un altre punt a considerar és que qualsevol pràctica que faci ús de recursos protegits pel copyright no podrà en cap cas publicar-se en Mosaic, la revista del Graduat en Multimèdia de la UOC, llevat que els propietaris dels drets intel·lectuals donin la seva autorització explícita

PART I Preparació de les Dades

Definició de la tasca de mineria de dades

Aquesta pràctica tracta de plantejar com podria ser un projecte real de mineria de dades. Com a analistes de dades a partir de la presentació del client que exposa un problema de negoci difús i molt genèric haurem de reconduir-lo com a projecte de mineria de dades.

El client ens proporciona un conjunt de dades extretes del seu ERP, format per les següent taules: `cabeceraticket`, `client`, `familia`, `lineasticket`, `pais`, `pedido`, `producto`, `promocion`, `proveedor`, `regiongeografica`, `seccion`, `subfamilia`, `tienda`.

Els objectius principals del projecte de mineria de dades seran els següents:

En primer lloc, com què tenim poca informació del domini i volem començar a tenir-ne una idea més clara, intentarem **trobar similituds i agrupar objectes semblants**.

En segon lloc, a partir de la situació més informada obtinguda en el pas anterior, tractarem de **classificar els objectes**. El que es vol és estudiar millor les diferències entre grups i les seves característiques peculiars.

PRIMER OBJECTIU

Trobar grups de clients semblants.

SEGON OBJECTIU

Un cop separats els clients en diversos grups, volem saber quin és l´atribut que distingueix millor un grup de clients o l´altre.

Pre-processament de les Dades

Carrega i exàmen preliminar del conjunt de dades

El conjunt de dades `client`

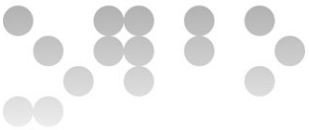
En primer lloc, instal.larem el paquet `readr`[4] que forma part del ecosistema `tidyverse`[5] i que ens permetrà llegir les dades:

```
# La forma més sencilla de instal.lar readr es instal.lar tidyverse
install.packages("tidyverse")

# Alternativament, podem instal.lar només readr
install.packages("readr")
```

[4] Paquet per a la lectura de dades amb format rectangular: **https://readr.tidyverse.org/**

[5] Conjunt de paquets R per a la Ciència de les Dades :**https://www.tidyverse.org/**



Un cop instal.lat el paquet el carregarem a la sessió R mitjançant la següent línia de codi:

```
# Carrega de readr
library(readr)

# Alternativament, com que forma part de tidyverse
library(tidyverse)
```

Observem que, hem fet ús de la segona opció que carrega tots els paquets de `tidyverse`, ja que utilitzarem per a la realització de la pràctica altres paquets, com per exemple: `dplyr` (per a la transformació de dades),`tibble` (per a un tractament més refinat de `data.frames`), `ggplot2` (per a la visualització de les dades), etc.

Un cop carregat el paquet a la sessió R, ja podem fer ús de les funcions. Per a importar les dades dels clients utilitzarem la funció `read_csv()`:

```
# Carreguem la llibreria que ens permet importar arxius CSV
if (!require("readr")) {
  # Instal.lació de la llibreria
install.packages("readr")
# Carreguem la llibreria
library(readr)
}
client <- read_csv("data/gourmetdb/cliente.csv",
                   col_names = FALSE)
```

Convertim el conjunt de dades `client` que és del tipus `data.frame` a `tibble`:

```
# Convertim el dataframe a tibble
as_tibble(client)
```

```
## # A tibble: 4,069 x 12
##    X1    X2    X3        X4 X5    X6    X7       X8 X9    X10    X11
X12
##    <chr> <chr> <chr>  <int> <chr> <chr> <chr> <int> <chr> <ch> <int>
<int>
```

```
##  1 0000~ Roca~ Homb~ 1.96e7 Solt~ Piaz~ Econ~     0 Sur ~ Esp~     4
7
##  2 0065~ Fuen~ Mujer 1.94e7 Casa~ C/ N~ Inge~     1 Sur ~ Esp~    16
13
##  3 0065~ Prat~ Homb~ 1.94e7 Casa~ cors~ Doct~     2 Sur ~ Esp~    14
10
##  4 0000~ Jone~ Homb~ 1.91e7 Solt~ 1 Pl~ Inge~     0 Nort~ Rei~     2
9
##  5 0000~ Burt~ Homb~ 1.94e7 Casa~ 46 S~ Doct~     2 Nort~ Rei~    13
9
##  6 0065~ Sale~ Mujer 1.94e7 Casa~ Leop~ Econ~     1 Nort~ Rei~     7
11
##  7 0065~ Crui~ Homb~ 1.96e7 Solt~ 2 Re~ Inge~     0 Nort~ Rei~    10
12
##  8 0131~ Cole~ Homb~ 1.94e7 Casa~ 67 E~ Doct~     0 Nort~ Est~     2
6
##  9 0131~ Shav~ Mujer 1.96e7 Casa~ 432 ~ Econ~     3 Nort~ Est~    21
15
## 10 0196~ Mill~ Homb~ 1.94e7 Divo~ 68 A~ Econ~     0 Nort~ Rei~     5
9
## # ... with 4,059 more rows
```

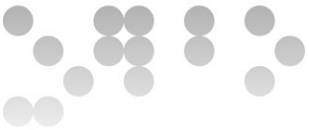
Podem adonar-nos que, el conjunt de dades està format per 4.069 observacions i 12 variables. A més, amb l´ajuda de `tibble` també podem observar el tipus per a cada columna.

Com que el nom de les columnes es poc descriptiu per alguns dels atributs, personalitzarem els noms mitjançant la següent línia de codi:

```
# Noms dels atributs
names(client) <- c("codi", "nom", "genere", "naixement",
                   "estatcivil", "direccio", "professio", "nombrefills",
                   "regio", "nacionalitat", "totalcompres",
"puntsacumulats")
```

Podem comprovar el nom de les columnes mitjançant la funció `colnames`:

```
# Comprovem es nom de les columnes
colnames(client)
```



```
##  [1] "codi"           "nom"            "genere"         "naixement"
##  [5] "estatcivil"     "direccio"       "professio"      "nombrefills"
##  [9] "regio"          "nacionalitat"   "totalcompres"
"puntsacumulats"
```

Exploració i tractament de valors desconeguts

Per altra banda, ens caldria comprovar que el nostre conjunt de dades no conté valors desconeguts:

```
# Estadístiques de valors buits.
colSums(is.na(client))
```

```
##           codi            nom         genere      naixement
estatcivil
##              0              0              0              0
805
##       direccio      professio    nombrefills          regio
nacionalitat
##              0              0            805              0
0
##   totalcompres puntsacumulats
##              0              0
```

Com es pot observar la variable `estatcivil` conté 805 observacions amb valors desconeguts. Amb l´objectiu de fer aquest grup més descriptiu podríem canviar aquests valors per la constant `Desconegut`:

```
# Amb l´ajuda de un test lògic descobrim els valors desconeguts
missing_values_estat_civil <- is.na(client$estatcivil)
# Reemplacem els valors desconeguts amb la  constant
client$estatcivil[missing_values_estat_civil] <- "Desconegut"
```

A més, fixe-mos que la variable `nombrefills` també conté 805 observacions amb valors desconeguts. En aquest cas, reemplaçarem els valor desconeguts amb un valor

aleatori de la distribució de la variable. Em primer lloc, amb l´ajuda d´un test lógic descobrim els valors desconeguts:

```
# Amb l´ajuda de un test lògic descobrim els valors desconeguts
missing_values_nombrefills <- is.na(client$nombrefills)
```

A continuació, generem un valor aleatori de la distribució de la variable:

```
# Generem observacions aleàtories
random_nombrefills_obs <- sample(na.omit(client$nombrefills), 1)
random_nombrefills_obs

## [1] 2
```

Finalment, reemplacem els valors desconeguts amb el valor aleatori calculat en en fragment de codi anterior:

```
# Reemplacem els valors desconeguts amb la observació aleàtoria

client$nombrefills[missing_values_nombrefills] <- random_nombrefills_obs
```

Transformació d´atributs

Per a facilitar l´ànalisi sería convenient canviar els atributs de tipus `character` a `factor`, que és la manera que té R de tractar amb les variables de tipus categòric:

```
# Carreguem ecosistema tidyverse
if (!require("tidyverse")) {
  # Instal.lació de la llibreria
install.packages("tidyverse")
# Carreguem la llibreria
library(tidyverse)
}
# Canviem les variables de tipus `character` a `factor`
```

```
cols <- c('codi', 'nom', 'genere', 'estatcivil', 'direccio',
          'professio', 'regio', 'nacionalitat')
client <- mutate_at(client, cols, as.factor)
```

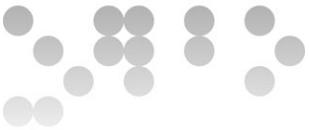
Fixe-mos amb el codi anterior que amb l´ajuda de la funció `dplyr::mutate_at`[6] hem canviat les columnes de tipus `character` al tipus `factor`.

Amb el següent fragment de codi i amb l´ajuda de la funció `lapply()` verifiquem que s´han produït els canvis:

```
# Retorna el tipus de cada variable
lapply(client, class)

## $codi
## [1] "factor"
##
## $nom
## [1] "factor"
##
## $genere
## [1] "factor"
##
## $naixement
## [1] "integer"
##
## $estatcivil
## [1] "factor"
##
## $direccio
## [1] "factor"
##
## $professio
## [1] "factor"
##
## $nombrefills
## [1] "integer"
##
```

[6] La notació `paquet::nom_funció` s´utilitza per a indicar a R que es vol fer ús de la funció del paquet indicat, en el cas que existeixi ambigüitat amb el nom d´una funció en un altre paquet.



```
## $regio
## [1] "factor"
##
## $nacionalitat
## [1] "factor"
##
## $totalcompres
## [1] "integer"
##
## $puntsacumulats
## [1] "integer"
```

Així mateix, ens sería práctic convertir la variable `naixement` del tipus `numeric` a `date`:

```
# Carreguem lubridate per al tractament de dades de tipus date
if (!require("lubridate")) {
  # Instal.lació de la llibreria
install.packages("lubridate")
# Carreguem la llibreria
library(lubridate)
}
# Convertim la variable naixement a tipus date
client <- client %>% mutate_at("naixement", funs(ymd))
```

Gràcies a l´anterior canvi de tipus podem calcular l´edat del client de la següent manera:

```
# Carreguem ecosistema tidyverse
if (!require("tidyverse")) {
  # Instal.lació de la llibreria
install.packages("tidyverse")
# Carreguem la llibreria
library(tidyverse)
}
client <- client %>%
        mutate(edat = year(Sys.Date()) - year(client$naixement))
```

El següent punt a considerar es realitzar una classificació dels nostres clients

Reducció de la dimensionalitat

Per a la simplificació de l´anàlisi les següents variables són descartades:

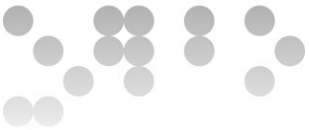
```
# Reducció de la dimensionalitat
client$codi <- NULL
client$nom <- NULL
client$naixement <- NULL
client$direccio <- NULL
```

Els motius són els següents:

- La variable `codi` representa la clau primaria en la base de dades i no ens proporcionará informació per a estudiar millor les diferències entre grups.
- La variable `nom` pot ser el.liminada pel fet que, es tracta del nom del individu i que té funcions de particularització o individualització i no ens serveix per a les tasques d´agrupació i classificació.
- La variable `direccio` es tracta de la direcció del domicili del client i no ens proporciona informació rellevant per a la nostra anàlisi. En canvi, les variables `regio` i `nacionalitat` són atributs més adequats per al nostre proposit.
- La variable `naixement` pot ser el.liminada, ja que representa la data de naixement del client i no proporciona informació per al nostre model. No obstant, la edat del client es una característica peculiar i es pot estimar a partir de la data de naixement i la data actual. Aquest atribut ja ha sigut calculat i representat amb la variable `edat`.

Pel que fa a, la variable `genere` podem observar amb el següent fragment de codi

```
# Resum dels valors que conté la variable workclass
unique(client$genere)
```



```
## [1] Hombre  Mujer   Empresa
## Levels: Empresa Hombre Mujer

unique(client$professio)

##  [1] Economistas,Abogados & Admin.Empresas
##  [2] Ingenieros & Especialistas
##  [3] Doctores & Profesionales de la Salud
##  [4] Gerentes & Directivos
##  [5] Catering
##  [6] Architectos,Decoradores & Humanistas
##  [7] Servicios
##  [8] Ama de Casa
##  [9] Alimentación
## [10] Food
## 10 Levels: Alimentación ... Servicios

summary(client$professio)

##                          Alimentación
##                                   271
##                           Ama de Casa
##                                   199
##  Architectos,Decoradores & Humanistas
##                                   485
##                              Catering
##                                   339
##  Doctores & Profesionales de la Salud
##                                   663
## Economistas,Abogados & Admin.Empresas
##                                   707
##                                  Food
##                                     2
##                 Gerentes & Directivos
##                                   703
##            Ingenieros & Especialistas
##                                   507
##                             Servicios
##                                   193

summary(client$genere)

## Empresa  Hombre   Mujer
##     805    2076    1188
```

Recollint tot el que s´ha realitzat, tenim 9 atributs per a la nostra anàlisi i 4069 observacions en el conjunt de dades `client`:

```
# Obtenim el nom de les variables
colnames(client)

## [1] "genere"         "estatcivil"     "professio"      "nombrefills"
## [5] "regio"          "nacionalitat"   "totalcompres"
"puntsacumulats"
## [9] "edat"

# Obtenim la dimensió
dim(client)

## [1] 4069    9
```

Anàlisi Exploratori de les Dades

https://artyco.com/como-clasificar-segmentar-clientes/

PART II Clustering

PART III Classificació

PART IV Regles d´associació

PART V Conclusions i Recomanacions al Client

Bibliografia

[1] Daniel T. Larouse, Chantal D. Larouse: Data Mininig and Predictive Analytics.USA, John Wiley & Sons,2015,ISBN 978-1-118-11619-7

[2] Jordi Gironés Roig, Jordi Casas Roma, Julià Minguillón Alfonso, Ramon Caihuelas Quiles : Minería de Datos: Modelos y Algoritmos. Barcelona, Editorial UOC, 2017, ISBN: 978-84-9116-904-8.

[3] Jiawe Han, Michellie Chamber & Jian Pei: Data mining : concepts and techniques. 3º Edition. USA, Editorial Elsevier, 2012, ISBN 978-0-12-381479-1