PEC 4

Adrián Valls Carbó y Javier Herrero Martín

2023-01-08

Índice

1. Detalles de la actividad

1.1. Descripción

1.2. Objetivos

1.3. Competencias

2. Resolución

2.1. Descripción del DataSet

Este conjunto de datos contiene información de una muestra extraída a partir de un censo estadounidense, en el que para cada persona (sin datos personales), se registran los salarios aparte de información personal adicional. Los datos han sido obtenidos en el sitio web de Kaggle. El conjunto de datos contiene 32.560 registros y 15 variables y se encuentra en formato `.csv`, bajo el nombre `adult.csv`

Las variables de esta muestra son:

- `age`: Edad del individuo. Variable continua expresada en años
- `workclass`: Categorización del individuo en base al perfil laboral. Presenta las categorías: *private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked*
- `fnlwgt`: Peso asignado a cada fila, refleja la proporción de datos que se asimilan dentro de la misma línea (misma información)
- `education`: Nivel de formación educativa del individuo. Contiene las categorías: *Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.*
- `education.num`: Número de años de formación educativa del individuo.
- `marital.status`: Estado civil del individuo. Categorizada en: *Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse*

- `occupation`: Categorización del individuo en base a la tipología de trabajo. Contiene las categorías: *Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces*
- `relationship`: Estado civil del individuo (a diferencia de marital_status, también hace referencia a hijos). Las categorías descritas sonç. *Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried*
- `race`: Grupo racial al que pertenece el individuo. Dentro de ellas se encuentran: *White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black*
- `sex`: Género del individuo: *Female, Male*
- `capital.gain`: Ganancias capitales del individuo €.
- `capital.loss`: Pérdidas capitales del individuo €.
- `native.country`: País de procedencia del individuo, dentro de los que se encuentran los siguientes: *United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands*
- `hours.per.week`: Horas por semana trabajadas por el individuo.
- `income`: Salario (anual) del individuo, en k€, hace referencia a un umbral de salario. Presenta las categorías *>50K, <=50K*

2.2. Importancia y objetivos del análisis

La idea original del dataset es analizar y predecir cuáles de dichas variables del censo tienen impacto en la probabilidad de que el individuo gane o no más de 50K de salario anual. Si bien el objetivo de la práctica no es específicamente la predicción de probabilidades, la cantidad de variables nos va a permitir realizar el preprocesado de los datos (tanto dentro de las propias variables como eligiendo qué variables son necesarias para el estudio), así como un análisis de la relevancia de dichas variables.

La importancia de este dataset podría encontrarse en el uso que pudieran hacer desde empresas financieras para conceder créditos a sus clientes en función de saber cuánto llegarán a ganar

2.3. Limpieza y cargado de los datos

Leemos el primer lugar el archivo. Para ello tenemos que emplear la función `read.csv` contenida dentro del paquete base de R.

```
# Leemos el archivo
df = read.csv("adult.csv")

# Examinamos los primeros registros
head(df[,1:5])
```

```
##   age workclass fnlwgt    education education.num
## 1  90         ?  77053      HS-grad             9
## 2  82   Private 132870      HS-grad             9
## 3  66         ? 186061 Some-college            10
## 4  54   Private 140359      7th-8th             4
## 5  41   Private 264663 Some-college            10
## 6  34   Private 216864      HS-grad             9
```

Podemos, una vez cargados los datos, examinar cómo R ha leido cada variable y si de forma correcta las ha interpretado.

```
## Llamamos a la funcion str
str(df)
```

```
## 'data.frame':    32561 obs. of  15 variables:
##  $ age           : int  90 82 66 54 41 34 38 74 68 41 ...
##  $ workclass     : chr  "?" "Private" "?" "Private" ...
##  $ fnlwgt        : int  77053 132870 186061 140359 264663 216864 150601 88638 422013 70037 ...
##  $ education     : chr  "HS-grad" "HS-grad" "Some-college" "7th-8th" ...
##  $ education.num : int  9 9 10 4 10 9 6 16 9 10 ...
##  $ marital.status: chr  "Widowed" "Widowed" "Widowed" "Divorced" ...
##  $ occupation    : chr  "?" "Exec-managerial" "?" "Machine-op-inspct" ...
##  $ relationship  : chr  "Not-in-family" "Not-in-family" "Unmarried" "Unmarried" ...
##  $ race          : chr  "White" "White" "Black" "White" ...
##  $ sex           : chr  "Female" "Female" "Female" "Female" ...
##  $ capital.gain  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ capital.loss  : int  4356 4356 4356 3900 3900 3770 3770 3683 3683 3004 ...
##  $ hours.per.week: int  40 18 40 40 40 45 40 20 40 60 ...
##  $ native.country: chr  "United-States" "United-States" "United-States" "United-States" ...
##  $ income        : chr  "<=50K" "<=50K" "<=50K" "<=50K" ...
```

Vemos en el epígrafe anterior varias cosas. Por un lado podemos ver que los datos perdidos son codificados como '?'. Esto nos conllevará problemas más adelante a la hora de analizar los datos, así que vamos a sustituirlo. En este caso podemos usar R base

```
# Sustituimos los datos
df[df=="?"]<-NA
```

También podemos ver que realmente los datos que son de tipo `chr` deberían serlo del tipo `factor`, por lo que podemos definir una función en la que si la columna es de tipo caracter la transforme en factor

```
# Transformamos todas las columnas que sean caracteres en factor
df[sapply(df, is.character)] <- lapply(df[sapply(df, is.character)],
                                       as.factor)

# Comprobamos que han cambiado
str(df)
```

```
## 'data.frame':    32561 obs. of  15 variables:
##  $ age           : int  90 82 66 54 41 34 38 74 68 41 ...
##  $ workclass     : Factor w/ 8 levels "Federal-gov",..: NA 4 NA 4 4 4 4 7 1 4 ...
##  $ fnlwgt        : int  77053 132870 186061 140359 264663 216864 150601 88638 422013 70037 ...
##  $ education     : Factor w/ 16 levels "10th","11th",..: 12 12 16 6 16 12 1 11 12 16 ...
##  $ education.num : int  9 9 10 4 10 9 6 16 9 10 ...
##  $ marital.status: Factor w/ 7 levels "Divorced","Married-AF-spouse",..: 7 7 7 1 6 1 6 5 1 5 ...
##  $ occupation    : Factor w/ 14 levels "Adm-clerical",..: NA 4 NA 7 10 8 1 10 10 3 ...
##  $ relationship  : Factor w/ 6 levels "Husband","Not-in-family",..: 2 2 5 5 4 5 5 3 2 5 ...
##  $ race          : Factor w/ 5 levels "Amer-Indian-Eskimo",..: 5 5 3 5 5 5 5 5 5 5 ...
##  $ sex           : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 2 1 1 2 ...
##  $ capital.gain  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ capital.loss  : int  4356 4356 4356 3900 3900 3770 3770 3683 3683 3004 ...
##  $ hours.per.week: int  40 18 40 40 40 45 40 20 40 60 ...
##  $ native.country: Factor w/ 41 levels "Cambodia","Canada",..: 39 39 39 39 39 39 39 39 39 NA ...
##  $ income        : Factor w/ 2 levels "<=50K",">50K": 1 1 1 1 1 1 1 2 1 2 ...
```

2.4. Examinando los datos perdidos

Tenemos que examinar en nuestro conjunto de datos si disponemos de datos que no estén disponibles (NA o *Not Available*).

```r
# Buscamos los datos perdidos
sapply(df, function (x) paste0(sum(is.na(x)),
                               " (", round(sum(is.na(x))/length(x)*100, 2), "%)"))
```

```
##            age      workclass         fnlwgt      education  education.num
##       "0 (0%)" "1836 (5.64%)"       "0 (0%)"       "0 (0%)"       "0 (0%)"
## marital.status     occupation   relationship           race            sex
##       "0 (0%)" "1843 (5.66%)"       "0 (0%)"       "0 (0%)"       "0 (0%)"
##   capital.gain   capital.loss hours.per.week native.country         income
##       "0 (0%)"       "0 (0%)"       "0 (0%)"  "583 (1.79%)"       "0 (0%)"
```

Vemos que tanto `workclass` como `occupation` tienen 1836 registros perdidos. En el caso de occupation vemos que tiene unos 7 registros perdidos más. Esto supone alrededor de un 6 % de los datos. Por otro lado en `native.country` hay 583 registros perdidos, lo que supone un 1.79 % de los datos perdidos.

Con los datos perdidos podemos realizar varias acciones:

- Etiquetado: Simplemente podríamos asignarles una etiqueta y analizarlos como una categoría más
- Reemplazarlos por una medida de distribución central: podríamos reemplazarlos por la mediana. El problema es que los datos perdidos se agrupan en nuestro caso dentro de variables categóricas, por lo que podríamos sustituirlo en este caso por la moda.
- Imputarlos: es decir, estimar la probabilidad en función a las otras variables de a qué categoría pertence el dato en concreto.
- Omitirlos: es decir, eliminar aquellos registros que contengan datos perdidos

De cara a imputarlos o no habría que determinar cuál es el mecanismo de generación de los datos perdidos:

- Perdidos completamente aleatorios (MCAR por sus siglas en inglés): esto es que la probabilidad de que los datos estén perdidos es igual para todos los casos. Esto sería que entre todas las categorías la probabilidad de encontrar un dato perdido es constante
- Perdidos aleatorios (MAR por sus siglas en inglés): esto es que la probabilidad de encontrarse perdidos es constante según una categoría observada en los datos. Por ejemplo podría ser que dentro de una categoría concreta los encuestados no quisieran dar su salario, pero tenemos datos de otros de la misma categoría, por lo que podríamos deducir.
- Perdidos no aleatorios (MNAR por sus siglas en inglés): en este caso no sabemos el mecanismo por el que los datos se encuentran perdidos, y este no es debido al azar, por lo que no podemos de hecho deducir las categorias