Pruebas duplicados

2025-12-02

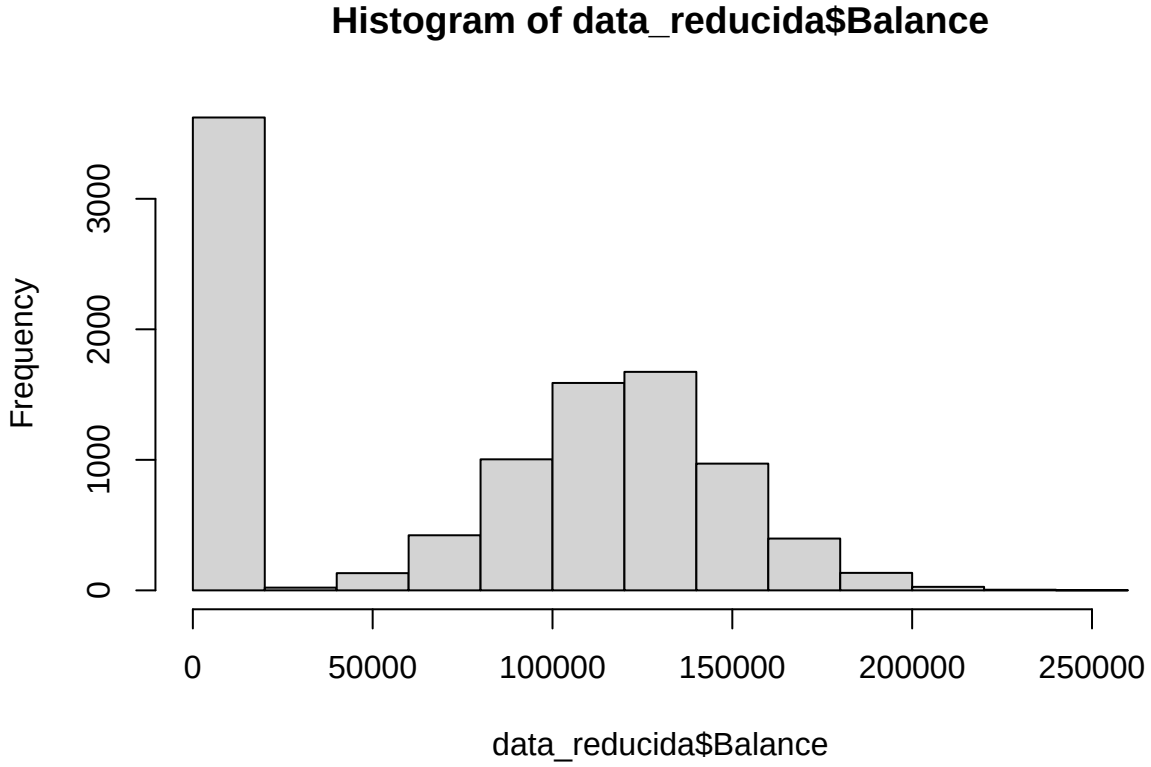
```
load("~/GitHub/Mineria/DATA/dataaaaaaaaaaaaaa.RData")
library(dplyr)
```

```
##
## Adjuntando el paquete: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
hist(data_reducida$Balance)
```



A continuación se hará la prueba de buscar valores duplicados discretizando la variable Balance

```
cortes <- c(-Inf, 0, 30000, 60000, 90000, 120000, 150000, 180000, 210000, 240000, Inf)
etiquetas <- c("Cero", "0-30k", "30k-60k", "60k-90k", "90k-120k", "120k-150k",
                       "150k-180k", "180k-210k", "210k-240k", ">240k")
data_reducida$Balance<- cut(
    data_reducida$Balance,
    breaks = cortes,
    labels = etiquetas,
    right = TRUE,
    include.lowest = TRUE
)
```

```
train_df <- data_reducida[data_reducida$group == "train", ]
test_df  <- data_reducida[data_reducida$group == "test", ]
```

Conteo duplicados

```
feature_cols <- setdiff(names(train_df), c("Exited", "group"))

dup_completos_train <- train_df[
  duplicated(train_df) |
  duplicated(train_df, fromLast = TRUE),
]

nrow(dup_completos_train)
```

```
## [1] 4479
```

Número idéntico

```
table(dup_completos_train$Exited)
```

```
##
##    0    1
## 3987  492
```

```
prop.table(table(dup_completos_train$Exited))
```

```
##
##         0         1
## 0.8901541 0.1098459
```

Duplicados más frecuentes en train:

```
freq_dup_train <- train_df %>%
  group_by(across(all_of(feature_cols))) %>%
  summarise(
    n      = n(),
    n_0    = sum(Exited == 0),
    n_1    = sum(Exited == 1),
    prop_0 = mean(Exited == 0),
```

```
    .groups = "drop"
  ) %>%
    arrange(desc(n), desc(n_0))
freq_dup_train
```

```
## # A tibble: 3,283 x 10
##    Geography Gender IsActiveMember NumOfProducts_grupo   Age Balance     n   n_0
##    <fct>     <fct>  <fct>          <fct>                <dbl> <fct>   <int> <int>
##  1 France    Male   0              2                       37 Cero       22    18
##  2 France    Male   0              2                       38 Cero       21    19
##  3 France    Male   1              2                       32 Cero       18    18
##  4 France    Male   1              2                       33 Cero       18    18
##  5 France    Female 0              2                       35 Cero       17    15
##  6 France    Male   0              2                       30 Cero       16    16
##  7 France    Male   1              2                       37 Cero       16    16
##  8 France    Male   1              2                       38 Cero       16    16
##  9 France    Male   1              2                       40 Cero       16    16
## 10 Spain     Male   0              2                       36 Cero       14    14
## # i 3,273 more rows
## # i 2 more variables: n_1 <int>, prop_0 <dbl>
```

Imputación de coincidencias perfectas

```
library(dplyr)

MIN_N <- 7
# Identificar combinaciones perfectamente predictivas de Exited = 0 con soporte mínimo
perfect_combinations <- freq_dup_train %>%
  filter(prop_0 == 1, n >= MIN_N) %>%
  select(all_of(feature_cols))
# Convertir columnas a caracter para la comparacion
perfect_combinations_clean <- perfect_combinations %>%
  mutate(across(all_of(feature_cols), as.character))

# poner todas como caracte para que coincidan con las combinaciones de antes, excepto exited
test_df_cleaned <- test_df %>%
  mutate(across(all_of(feature_cols), as.character),
         Exited = as.numeric(as.character(Exited)))

# Unir el data frame de prueba con las combinaciones perfectas e imputar Exited = 0 en caso de coincide
test_df_imputed <- test_df_cleaned %>%
  left_join(
    perfect_combinations_clean %>% mutate(match_found = TRUE),
    by = feature_cols
  ) %>%
  mutate(
    Exited = if_else(
      match_found == TRUE,
      0,
      Exited
```

```
    )
  ) %>%
  select(-match_found)
# Crear el data frame final añadiendo la columna Exited imputada al data frame original
test_df_final <- test_df %>%
  mutate(Exited = test_df_imputed$Exited)
```

```
summary(test_df_final)
```

```
##    Geography       Gender         Exited     IsActiveMember NumOfProducts_grupo
##  France :1527   Female:1368   Min.   :0      0:1434         1      :1491
##  Germany: 738   Male  :1632   1st Qu.:0      1:1566         2      :1409
##  Spain  : 735                 Median :0                     3 o más: 100
##                               Mean   :0
##                               3rd Qu.:0
##                               Max.   :0
##                               NA's   :2772
##       Age             Balance       group
##  Min.   :18.00   Cero     :1108   test :3000
##  1st Qu.:32.00   90k-120k : 653   train:   0
##  Median :37.00   120k-150k: 645
##  Mean   :39.02   60k-90k  : 253
##  3rd Qu.:44.00   150k-180k: 237
##  Max.   :84.00   180k-210k:  50
##                  (Other)  :  54
```