# Boosting try

2025-12-02

```
##
## 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
```

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

duplicados:

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

nrow(dup_completos_train)
```

```
## [1] 1905
```

duplicados sin exited ni group:

```
## [1] 2104
```

Hay filas idénticas con distinto Exited?

```
## [1] 176
```

Qué valores de Exited tienen los duplicados?

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

```
##
##    0    1
## 1785  120
```

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

```
##
##          0          1
## 0.93700787 0.06299213
```

Contra la proporcion total:

```r
table(train_df$Exited)
```

```
##
##    0    1
## 5550 1450
```

```r
prop.table(table(train_df$Exited))
```

```
##
##         0         1
## 0.7928571 0.2071429
```

Cuáles son los duplicados más frecuentes en train:

```
## # A tibble: 5,367 x 10
##    Geography Gender IsActiveMember NumOfProducts_grupo   Age Balance     n   n_0
##    <fct>     <fct>  <fct>          <fct>                <dbl>   <dbl> <int> <int>
##  1 France    Male   0              2                       37       0    22    18
##  2 France    Male   0              2                       38       0    21    19
##  3 France    Male   1              2                       32       0    18    18
##  4 France    Male   1              2                       33       0    18    18
##  5 France    Female 0              2                       35       0    17    15
##  6 France    Male   0              2                       30       0    16    16
##  7 France    Male   1              2                       37       0    16    16
##  8 France    Male   1              2                       38       0    16    16
##  9 France    Male   1              2                       40       0    16    16
## 10 France    Female 0              2                       34       0    14    13
## # i 5,357 more rows
## # i 2 more variables: n_1 <int>, prop_0 <dbl>
```

Ahora nos centramos en los patrones con Exited=1

```
## # A tibble: 471 x 10
##    Geography Gender IsActiveMember NumOfProducts_grupo   Age Balance n_total
##    <fct>     <fct>  <fct>          <fct>                <dbl>   <dbl>   <int>
##  1 France    Female 0              1                       55       0       3
##  2 France    Female 0              1                       49       0       2
##  3 France    Female 1              1                       47       0       2
##  4 France    Female 1              1                       49       0       2
##  5 France    Female 1              1                       64       0       2
##  6 France    Female 1              3 o más                 43       0       2
##  7 France    Male   0              1                       33       0       2
```

```
##  8 France    Male   0             1                      45       0       2
##  9 France    Male   1             1                      49       0       2
## 10 Germany   Female 0             2                      52       0       2
## # i 461 more rows
## # i 3 more variables: n_1 <int>, n_0 <int>, prop_1 <dbl>
```

## Duplicados sin Balance=0

Para Exited=0

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


freq_dup_train <- train_df %>%
  filter(Balance != 0) %>%
  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(prop_0), desc(n))

freq_dup_train
```

```
## # A tibble: 4,488 x 10
##    Geography Gender IsActiveMember NumOfProducts_grupo   Age Balance     n   n_0
##    <fct>     <fct>  <fct>          <fct>                <dbl>   <dbl> <int> <int>
##  1 Germany   Female 1              1                       33 180075.     2     2
##  2 France    Female 0              1                       20 134398.     1     1
##  3 France    Female 0              1                       22  89493.     1     1
##  4 France    Female 0              1                       22 102347.     1     1
##  5 France    Female 0              1                       23 131255.     1     1
##  6 France    Female 0              1                       24  88162.     1     1
##  7 France    Female 0              1                       24 106234.     1     1
##  8 France    Female 0              1                       24 140454.     1     1
##  9 France    Female 0              1                       24 148299.     1     1
## 10 France    Female 0              1                       25  79544.     1     1
## # i 4,478 more rows
## # i 2 more variables: n_1 <int>, prop_0 <dbl>
```

No se observan repeticiones significativas

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

freq_dup_train <- train_df %>%
  filter(Balance != 0) %>%
  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),
    prop_1 = mean(Exited == 1),
    .groups = "drop"
  ) %>%
  arrange(desc(prop_1), desc(n))

freq_dup_train
```

```
## # A tibble: 4,488 x 11
##    Geography Gender IsActiveMember NumOfProducts_grupo   Age Balance     n   n_0
##    <fct>     <fct>  <fct>          <fct>                <dbl>   <dbl> <int> <int>
##  1 France    Female 0              1                       22 150126.     1     0
##  2 France    Female 0              1                       23  83739.     1     0
##  3 France    Female 0              1                       26 108349.     1     0
##  4 France    Female 0              1                       27 127472.     1     0
##  5 France    Female 0              1                       28  91858.     1     0
##  6 France    Female 0              1                       28  93249.     1     0
##  7 France    Female 0              1                       28 103458.     1     0
##  8 France    Female 0              1                       30  87773.     1     0
##  9 France    Female 0              1                       30 169743.     1     0
## 10 France    Female 0              1                       31  81554.     1     0
## # i 4,478 more rows
## # i 3 more variables: n_1 <int>, prop_0 <dbl>, prop_1 <dbl>
```

Tampoco

# DESCURBIMIENTO FINAL

Dado que se ha descubierto que existen ciertos patrones significativos sobre la variable Exited cuando Balance=0 y ninguno cuando es diferente de 0, nos interesa comprobar cuantas observaciones hay con Balance=0 en test.

```
nrow(test_df[test_df$Balance == 0, ])
```

```
## [1] 1108
```

Más de 1/3 de los datos test son contienen Balance=0. Proximos pasos: imponer reglas asociativas al modelo que impongan una clase para Exited para ciertas combinaciones de variables.