Estudio de Reglas Asociativas

2025-10-24

```
## Cargando paquete requerido: Matrix

##
## Adjuntando el paquete: 'arules'

## The following objects are masked from 'package:base':
##
##     abbreviate, write

##
## Adjuntando el paquete: 'dplyr'

## The following objects are masked from 'package:arules':
##
##     intersect, recode, setdiff, setequal, union

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

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

Adaptación de los datos

```
data_ar <- subset(data_transformada, select = -c(Surname, ID, group))
data_ar <- data_ar %>%
  filter(!is.na(Exited))
str(data_ar)
```

```
## 'data.frame':    7000 obs. of  21 variables:
##  $ Exited                : Factor w/ 2 levels "0","1": 1 1 2 2 1 1 1 1 1 1 ...
##  $ Tenure                : num  2 1 8 9 3 2 4 4 3 5 ...
##  $ Gender                : Factor w/ 2 levels "Female","Male": 1 1 2 1 2 2 2 2 2 1 ...
##  $ EducationLevel        : Factor w/ 4 levels "High School",..: 4 4 4 1 1 4 1 4 4 1 ...
##  $ LoanStatus            : Factor w/ 3 levels "Active loan",..: 2 3 3 1 1 3 1 1 3 1 ...
##  $ NetPromoterScore      : num  10 9 8 9 6 9 4 8 10 5 ...
##  $ TransactionFrequency  : num  34 31 26 32 41 33 22 31 23 29 ...
##  $ Age                   : num  29 41 43 48 34 32 34 26 29 30 ...
```

```
##  $ Geography             : Factor w/ 3 levels "France","Germany",..: 1 1 3 2 1 3 3 2 1 2 ...
##  $ HasCrCard             : Factor w/ 2 levels "0","1": 2 1 2 1 2 2 1 2 1 2 ...
##  $ EstimatedSalary       : num  105760 95623 135651 102641 83773 ...
##  $ IsActiveMember        : Factor w/ 2 levels "0","1": 1 1 1 2 2 1 1 2 2 2 ...
##  $ AvgTransactionAmount  : num  157.9 110.8 228.8 133.9 91.6 ...
##  $ CustomerSegment       : Factor w/ 3 levels "Affluent","High Net Worth",..: 3 3 3 1 1 3 3 3 3 1 ...
##  $ MaritalStatus         : Factor w/ 4 levels "Divorced","Married",..: 2 2 3 2 3 2 3 2 2 2 ...
##  $ DigitalEngagementScore: num  60 73 41 55 67 43 67 56 69 45 ...
##  $ CreditScore           : num  832 513 577 482 635 656 745 748 797 485 ...
##  $ SavingsAccountFlag    : Factor w/ 2 levels "0","1": 2 2 2 1 1 2 1 2 2 2 ...
##  $ Balance               : num  108122 65190 79757 109472 190067 ...
##  $ ComplaintsCount_bin   : Factor w/ 2 levels "No_queja","Queja": 1 1 1 1 1 1 1 1 1 1 ...
##  $ NumOfProducts_grupo   : Factor w/ 3 levels "1","2","3 o más": 2 1 1 1 1 2 2 1 1 1 ...
```

Para poder utilizar arules será necesario transformar nuestros datos numéricos y categóricos a factor. Para los valores numéricos se hará una partición en intervalos, los categóricos pasarán a ser factor directamente.

Categóricas:

```r
data_ar <- data_ar %>%
  mutate(across(where(is.character), as.factor))
```

Numéricas (transformación 1 a 1 con cortes personalizados)

```r
data_ar <- data_ar %>%
  mutate(
    Tenure = cut(Tenure,
                 breaks = c(0, 3, 6, 10),
                 labels = c("Nuevo (0-3 años)", "Medio (4-6 años)", "Antiguo (7-10 años)"),
                 include.lowest = TRUE)
  )
data_ar <- data_ar %>%
  mutate(
    NetPromoterScore = cut(NetPromoterScore,
                           breaks = c(-1, 6, 8, 10),  # -1 para incluir el 0
                           labels = c("0-6", "7-8", "9-10"),
                           include.lowest = TRUE)
  )
data_ar <- data_ar %>%
  mutate(
    TransactionFrequency = cut(TransactionFrequency,
                               breaks = c(0, 20, 30, 40, max(TransactionFrequency, na.rm = TRUE)),
                               labels = c("0-20", "21-30", "31-40", "41+"),
                               include.lowest = TRUE)
  )
data_ar <- data_ar %>%
  mutate(
    Age = cut(Age,
              breaks = c(0, 25, 35, 45, 55, 65, 100),
              labels = c("18-25", "26-35", "36-45", "46-55", "56-65", "65+"),
              include.lowest = TRUE)
  )
data_ar <- data_ar %>%
  mutate(
```

```
    EstimatedSalary = cut(EstimatedSalary,
                          breaks = c(0, 30000, 60000, 90000, 120000, 150000, 180000,
                                     max(EstimatedSalary, na.rm = TRUE)),
                          labels = c("0-30K", "31-60K", "61-90K", "91-120K",
                                     "121-150K", "151-180K", "180K+"),
                          include.lowest = TRUE)
  )
data_ar <- data_ar %>%
  mutate(
    AvgTransactionAmount = cut(AvgTransactionAmount,
                               breaks = quantile(AvgTransactionAmount,
                                                 probs = c(0, 0.5, 0.8, 0.95, 1),
                                                 na.rm = TRUE),
                               labels = c("Bajo (0-50%)", "Medio (51-80%)",
                                          "Alto (81-95%)", "Muy Alto (96-100%)"),
                               include.lowest = TRUE)
  )

data_ar <- data_ar %>%
  mutate(
    DigitalEngagementScore = cut(DigitalEngagementScore,
                                 breaks = c(0, 25, 50, 75, 100),
                                 labels = c("0-25", "26-50", "51-75", "76-100"),
                                 include.lowest = TRUE)
  )
data_ar <- data_ar %>%
  mutate(
    CreditScore = cut(CreditScore,
                      breaks = c(300, 580, 670, 740, 800, 850),
                      labels = c("Muy Bajo (300-579)", "Bajo (580-669)",
                                 "Medio (670-739)", "Bueno (740-799)",
                                 "Excelente (800-850)"),
                      include.lowest = TRUE)
  )
data_ar <- data_ar %>%
  mutate(
    Balance = cut(Balance,
                  breaks = c(0, 1000, 5000, 15000, 50000, Inf),
                  labels = c("Muy Bajo (0-1K)", "Bajo (1-5K)",
                             "Medio (5-15K)", "Alto (15-50K)",
                             "Muy Alto (50K+)"),
                  include.lowest = TRUE)
  )

str(data_ar)
```

```
## 'data.frame':    7000 obs. of  21 variables:
##  $ Exited                : Factor w/ 2 levels "0","1": 1 1 2 2 1 1 1 1 1 1 ...
##  $ Tenure                : Factor w/ 3 levels "Nuevo (0-3 años)",..: 1 1 3 3 1 1 2 2 1 2 ...
##  $ Gender                : Factor w/ 2 levels "Female","Male": 1 1 2 1 2 2 2 2 2 1 ...
##  $ EducationLevel        : Factor w/ 4 levels "High School",..: 4 4 4 1 1 4 1 4 4 1 ...
##  $ LoanStatus            : Factor w/ 3 levels "Active loan",..: 2 3 3 1 1 3 1 1 3 1 ...
##  $ NetPromoterScore      : Factor w/ 3 levels "0-6","7-8","9-10": 3 3 2 3 1 3 1 2 3 1 ...
```

```
##  $ TransactionFrequency  : Factor w/ 4 levels "0-20","21-30",..: 3 3 2 3 4 3 2 3 2 2 ...
##  $ Age                   : Factor w/ 6 levels "18-25","26-35",..: 2 3 3 4 2 2 2 2 2 2 ...
##  $ Geography             : Factor w/ 3 levels "France","Germany",..: 1 1 3 2 1 3 3 2 1 2 ...
##  $ HasCrCard             : Factor w/ 2 levels "0","1": 2 1 2 1 2 2 1 2 1 2 ...
##  $ EstimatedSalary       : Factor w/ 7 levels "0-30K","31-60K",..: 4 4 5 4 3 4 2 5 5 5 ...
##  $ IsActiveMember        : Factor w/ 2 levels "0","1": 1 1 1 2 2 1 1 2 2 2 ...
##  $ AvgTransactionAmount  : Factor w/ 4 levels "Bajo (0-50%)",..: 3 2 4 2 1 1 1 1 3 1 ...
##  $ CustomerSegment       : Factor w/ 3 levels "Affluent","High Net Worth",..: 3 3 3 1 1 3 3 3 3 1 ..
##  $ MaritalStatus         : Factor w/ 4 levels "Divorced","Married",..: 2 2 3 2 3 2 3 2 2 2 ...
##  $ DigitalEngagementScore: Factor w/ 4 levels "0-25","26-50",..: 3 3 2 3 3 2 3 3 3 2 ...
##  $ CreditScore           : Factor w/ 5 levels "Muy Bajo (300-579)",..: 5 1 1 1 2 2 4 4 4 1 ...
##  $ SavingsAccountFlag    : Factor w/ 2 levels "0","1": 2 2 2 1 1 2 1 2 2 2 ...
##  $ Balance               : Factor w/ 5 levels "Muy Bajo (0-1K)",..: 5 5 5 5 5 1 1 5 1 5 ...
##  $ ComplaintsCount_bin   : Factor w/ 2 levels "No_queja","Queja": 1 1 1 1 1 1 1 1 1 1 ...
##  $ NumOfProducts_grupo   : Factor w/ 3 levels "1","2","3 o más": 2 1 1 1 1 2 2 1 1 1 ...
```

Las 21 variables son factores. Puede comenzar el análisis por Association Rules.

Transformación a una base de datos transaccional:

```
data_tr <- as(data_ar,"transactions")
data_tr
```

```
## transactions in sparse format with
##  7000 transactions (rows) and
##  73 items (columns)
```

Lor registros ahora son transacciones, y las variables se han desdoblado en sus categorías, obteniendo así la estructura para la base transaccional.

Un par de ejemplos de "transacciones"

```
inspect(data_tr[1:2])
```

```
##     items                                   transactionID
## [1] {Exited=0,
##      Tenure=Nuevo (0-3 años),
##      Gender=Female,
##      EducationLevel=University,
##      LoanStatus=Default risk,
##      NetPromoterScore=9-10,
##      TransactionFrequency=31-40,
##      Age=26-35,
##      Geography=France,
##      HasCrCard=1,
##      EstimatedSalary=91-120K,
##      IsActiveMember=0,
##      AvgTransactionAmount=Alto (81-95%),
##      CustomerSegment=Mass Market,
##      MaritalStatus=Married,
##      DigitalEngagementScore=51-75,
##      CreditScore=Excelente (800-850),
##      SavingsAccountFlag=1,
```

```
##       Balance=Muy Alto (50K+),
##       ComplaintsCount_bin=No_queja,
##       NumOfProducts_grupo=2}                              1
## [2] {Exited=0,
##       Tenure=Nuevo (0-3 años),
##       Gender=Female,
##       EducationLevel=University,
##       LoanStatus=No loan,
##       NetPromoterScore=9-10,
##       TransactionFrequency=31-40,
##       Age=36-45,
##       Geography=France,
##       HasCrCard=0,
##       EstimatedSalary=91-120K,
##       IsActiveMember=0,
##       AvgTransactionAmount=Medio (51-80%),
##       CustomerSegment=Mass Market,
##       MaritalStatus=Married,
##       DigitalEngagementScore=51-75,
##       CreditScore=Muy Bajo (300-579),
##       SavingsAccountFlag=1,
##       Balance=Muy Alto (50K+),
##       ComplaintsCount_bin=No_queja,
##       NumOfProducts_grupo=1}                              2
```

```r
SIZE <- size(data_tr)
summary(SIZE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      21      21      21      21      21      21
```

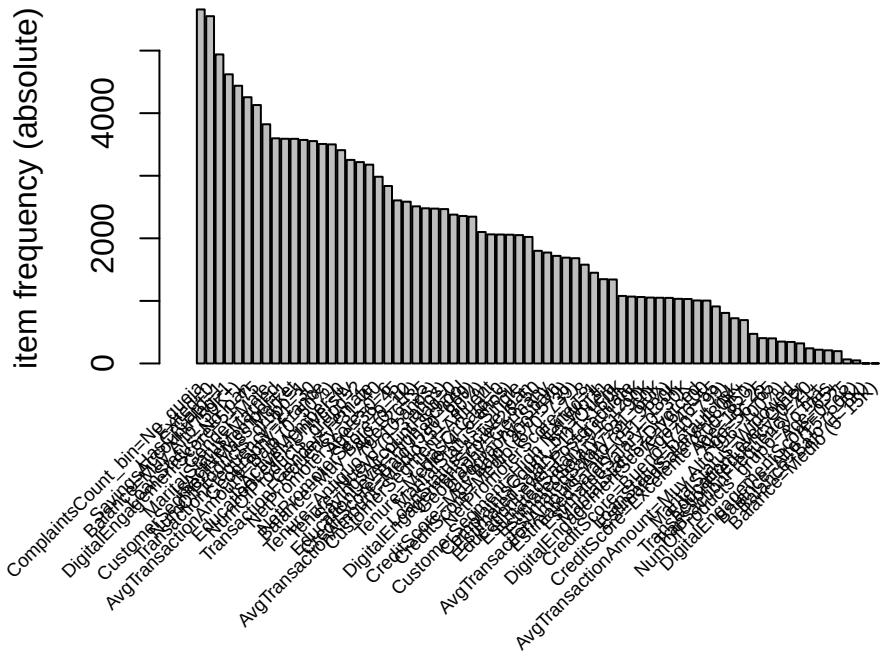
Todas las transacciones tienen 21 valores, por tanto la transformación se ha realizado correctamente (no hay valores faltantes).

Soporte mínimo

El soporte mínimo es el umbral de frecuencia que debe superar un conjunto de artículos para ser considerado relevante en la minería de reglas de asociación. Garantiza que las reglas descubiertas representen patrones significativos y no ruido estadístico.

Mediante una visualización se pretende ver cuántas veces aparece cada ítem en las transacciones, para poder elegir arbitrariamente un soporte mínimo que elimine el ruido estadístico.

```r
itemFrequencyPlot(data_tr, topN=100,type="absolute", cex.names = 0.6)
```



Para este caso se ha optado por un soporte mínimo de 0.05. Se trata del umbral por defecto en estadística para determinar significancia, y en esta base de datos en concreto puede servir para eliminar la cola de la derecha, compuesta por items prácticamente extintos. Todo lo que supere el 0.05 puede darnos información relevante sobre sus relaciones con otros items (sus reglas de asociación).

Reglas de asociación

Se utilizará el soporte mínimo de 0.05 establecido en el apartado anterior, y únicamente interesa estudiar itemsets compuestos por entre 1 y 5 items de la base de datos transaccional:

```
itemsets <- apriori(data = data_tr,
                    parameter = list(support = 0.05,
                                     minlen = 1,
                                     maxlen = 5,
                                     target = "frequent itemset"))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##          NA    0.1    1 none FALSE            TRUE       5    0.05      1
##  maxlen            target  ext
##       5 frequent itemsets TRUE
##
## Algorithmic control:
```

```
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 350
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[73 item(s), 7000 transaction(s)] done [0.01s].
## sorting and recoding items ... [63 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5

## Warning in apriori(data = data_tr, parameter = list(support = 0.05, minlen = 1,
## : Mining stopped (maxlen reached). Only patterns up to a length of 5 returned!

##  done [0.19s].
## sorting transactions ... done [0.00s].
## writing ... [52294 set(s)] done [0.01s].
## creating S4 object  ... done [0.01s].
```

```
summary(itemsets)
```

```
## set of 52294 itemsets
##
## most frequent items:
## ComplaintsCount_bin=No_queja                    Exited=0
##                        17390                       15011
##         SavingsAccountFlag=1                  HasCrCard=1
##                        13013                       12069
##           LoanStatus=No loan                     (Other)
##                        11742                      152471
##
## element (itemset/transaction) length distribution:sizes
##     1     2     3     4     5
##    63  1271  8066 19577 23317
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   4.000   4.000   4.239   5.000   5.000
##
## summary of quality measures:
##     support            count
##  Min.   :0.05000   Min.   : 350.0
##  1st Qu.:0.05700   1st Qu.: 399.0
##  Median :0.06814   Median : 477.0
##  Mean   :0.08137   Mean   : 569.6
##  3rd Qu.:0.09000   3rd Qu.: 630.0
##  Max.   :0.80829   Max.   :5658.0
##
## includes transaction ID lists: FALSE
##
## mining info:
##     data ntransactions support confidence
##  data_tr          7000    0.05          1
##
##  apriori(data = data_tr, parameter = list(support = 0.05, minlen = 1, maxlen = 5, target = "frequent
```

Se ha generado 52.000 itemsets de entre 1 y 5 items que superan el soporte mínimo. Unos 42.000 corresponden a itemsets de entre 4 y 5 items. Ojeamos los 5 itemsets más frecuentes:

```
top_5_itemsets <- sort(itemsets, by = "support", decreasing = TRUE)[1:5]
(inspect(top_5_itemsets))
```

```
##     items                                       support   count
## [1] {ComplaintsCount_bin=No_queja}              0.8082857 5658
## [2] {Exited=0}                                  0.7928571 5550
## [3] {HasCrCard=1}                               0.7055714 4939
## [4] {SavingsAccountFlag=1}                      0.6601429 4621
## [5] {Exited=0, ComplaintsCount_bin=No_queja} 0.6410000 4487
```

```
##                                         items   support count
## [1]           {ComplaintsCount_bin=No_queja} 0.8082857  5658
## [2]                                 {Exited=0} 0.7928571  5550
## [3]                              {HasCrCard=1} 0.7055714  4939
## [4]                     {SavingsAccountFlag=1} 0.6601429  4621
## [5] {Exited=0, ComplaintsCount_bin=No_queja} 0.6410000  4487
```

- El 80% de los clientes no han emitido ninguna queja.
- Cerca del 80% de los clientes no han dejado el banco
- Un 70% de los clientes tiene tarjeta de crédito
- El 66% de los clientes tiene cuenta de ahorros
- Un 64% de los clientes no han emitido ninguna queja y además se han quedado en el banco.

De los 5 itemsets más frecuentes únicamente se obtiene una información que implica más de una variable. Los clientes que no emiten quejas son propensos a quedarse en el banco.

A continuación se comenzará a buscar reglas de asociación mediante itemsets de entre 2 y 5 items. Inicialmente se fijará una confianza alta (80%) y se mantendrá el soporte escogido (0.05)

```
rules = apriori (data_tr, parameter = list (support=0.05, confidence=0.80, maxlen = 5, minlen=2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5    0.05      2
##  maxlen target  ext
##       5  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 350
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[73 item(s), 7000 transaction(s)] done [0.01s].
## sorting and recoding items ... [63 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5
```

```
## Warning in apriori(data_tr, parameter = list(support = 0.05, confidence = 0.8,
## : Mining stopped (maxlen reached). Only patterns up to a length of 5 returned!
```

```
##  done [0.19s].
## writing ... [42462 rule(s)] done [0.00s].
## creating S4 object  ... done [0.01s].
```

```
summary(rules)
```

```
## set of 42462 rules
##
## rule length distribution (lhs + rhs):sizes
##     2     3     4     5
##    67  1587 11319 29489
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   4.000   5.000   4.654   5.000   5.000
##
## summary of quality measures:
##     support          confidence        coverage            lift
##  Min.   :0.05000   Min.   :0.8000   Min.   :0.05000   Min.   :0.9897
##  1st Qu.:0.05843   1st Qu.:0.8247   1st Qu.:0.06714   1st Qu.:1.1276
##  Median :0.07157   Median :0.8585   Median :0.08214   Median :1.2085
##  Mean   :0.08527   Mean   :0.8755   Mean   :0.09780   Mean   :1.2460
##  3rd Qu.:0.09686   3rd Qu.:0.9263   3rd Qu.:0.11100   3rd Qu.:1.3304
##  Max.   :0.64100   Max.   :1.0000   Max.   :0.79286   Max.   :5.0885
##      count
##  Min.   : 350.0
##  1st Qu.: 409.0
##  Median : 501.0
##  Mean   : 596.9
##  3rd Qu.: 678.0
##  Max.   :4487.0
##
## mining info:
##     data ntransactions support confidence
##  data_tr          7000    0.05        0.8
##                                                                                                                call
##  apriori(data = data_tr, parameter = list(support = 0.05, confidence = 0.8, maxlen = 5, minlen = 2))
```

Se ha generado 42462 reglas, la mayoría (+40.000) compuestas por 4 o 5 items.

Eliminar reglas redundantes

Una regla es redundante cuando una regla más corta con el mismo consecuente tiene igual o mayor confianza. Si añadir condiciones no mejora la predicción, la regla extensa sobra.

Por ejemplo, si tenemos dos reglas de asociación: la regla {A} -> {C} con una confianza del 80%, y la regla {A , B} -> {C} que también tiene una confianza del 80%. En este caso, la segunda regla es redundante, ya que la adición del ítem B en el antecedente no incrementa el poder predictivo de la regla hacia el consecuente C.

```
reglas_Noredund <- rules[!is.redundant(x = rules, measure = "confidence")]
reglas_Noredund
```

```
## set of 14696 rules
```

El número de reglas se reduce drásticamente (ha quedado 1/3 de las reglas). Esto es muy positivo, dado que hemos eliminado información que estaba "duplicada" o que podemos simplificar.

Detección de patrones

Patrones de abandono del banco

Queremos encontrar patrones que nos demuestren qué características debe tener un cliente para las variables de la base de datos para que sea altamente probable que deje el banco, y así poder tomar medidas para que eso no suceda.

```
filtrado_reglas <- subset(x = reglas_Noredund,
                          subset = rhs %pin% "Exited=1")
filtrado_reglas_ordenadas <- sort(filtrado_reglas, by = "lift")
inspect(head(filtrado_reglas_ordenadas,10))
```

No se han obtenido reglas que cumplan los parámetros de soporte mínimo 0.05 y confianza 80%. Esto puede deberse a que el nivel de confianza exigido es muy elevado, considerando que los clientes que abandonan representan solo el 20% de la base de datos.

Una confianza del 50% ya identificaría combinaciones donde la probabilidad de abandono es 2.5 veces superior a la tasa base (pasando de 20% a 50%), lo que constituye un patrón de riesgo significativo para el negocio. También se bajará el soporte a la mitad para recoger más reglas que podrían ser importantes pero quedan escondidas dado el desbalanceo de la variable Exited.

Veamos si ahora se detectan reglas asociativas.

```
rules = apriori (data_tr, parameter = list (support=0.025, confidence=0.50, maxlen = 5, minlen=2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.5    0.1    1 none FALSE            TRUE       5   0.025      2
##  maxlen target  ext
##       5  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 175
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[73 item(s), 7000 transaction(s)] done [0.01s].
## sorting and recoding items ... [69 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5
```

```
## Warning in apriori(data_tr, parameter = list(support = 0.025, confidence = 0.5,
## : Mining stopped (maxlen reached). Only patterns up to a length of 5 returned!

##  done [0.33s].
## writing ... [508538 rule(s)] done [0.05s].
## creating S4 object  ... done [0.26s].
```

```
summary(rules)
```

```
## set of 508538 rules
##
## rule length distribution (lhs + rhs):sizes
##      2      3      4      5
##    851  17422 125512 364753
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00    4.00    5.00    4.68    5.00    5.00
##
## summary of quality measures:
##     support          confidence        coverage            lift
##  Min.   :0.02500   Min.   :0.5000   Min.   :0.02500   Min.   :0.6319
##  1st Qu.:0.03014   1st Qu.:0.5585   1st Qu.:0.04614   1st Qu.:1.0068
##  Median :0.03843   Median :0.6619   Median :0.05914   Median :1.0600
##  Mean   :0.04868   Mean   :0.6805   Mean   :0.07351   Mean   :1.1896
##  3rd Qu.:0.05500   3rd Qu.:0.7898   3rd Qu.:0.08414   3rd Qu.:1.2704
##  Max.   :0.64100   Max.   :1.0000   Max.   :0.80829   Max.   :5.2498
##      count
##  Min.   : 175.0
##  1st Qu.: 211.0
##  Median : 269.0
##  Mean   : 340.7
##  3rd Qu.: 385.0
##  Max.   :4487.0
##
## mining info:
##     data ntransactions support confidence
##  data_tr          7000   0.025        0.5
##                                                                                                                cal
##  apriori(data = data_tr, parameter = list(support = 0.025, confidence = 0.5, maxlen = 5, minlen = 2)
```

Evidentemente se generan más reglas (de ~42.000 a 155.293).

```
reglas_Noredund <- rules[!is.redundant(x = rules, measure = "confidence")]
filtrado_reglas <- subset(x = reglas_Noredund,
                          subset = rhs %pin% "Exited=1")
filtrado_reglas_ordenadas <- sort(filtrado_reglas, by = "lift")
inspect(head(filtrado_reglas_ordenadas,10))
```

```
##     lhs                                rhs           support confidence   coverage     lift count
## [1] {Age=46-55,
##      IsActiveMember=0,
##      ComplaintsCount_bin=No_queja} => {Exited=1} 0.02685714  0.5251397 0.05114286 2.535157   188
```

```
## [2] {Age=46-55,
##      SavingsAccountFlag=1,
##      NumOfProducts_grupo=1}        => {Exited=1} 0.02557143  0.5218659 0.04900000 2.519353   179
## [3] {Age=46-55,
##      IsActiveMember=0}             => {Exited=1} 0.03285714  0.5180180 0.06342857 2.500777   230
## [4] {Age=46-55,
##      NumOfProducts_grupo=1}        => {Exited=1} 0.03914286  0.5179584 0.07557143 2.500489   274
```

Se ha obtenido 4 reglas, todas ellas a tener en cuenta dado su elevado valor lift.

El lift es cercano a 2.5, lo que indica que la probabilidad de abandono entre clientes que cumplen el antecedente es 2.5 veces mayor que la probabilidad de abandono en la población general. Sabiendo esto, las reglas generadas tienen bastante valor. Son las siguientes:

- Un cliente de entre 46-55 años inactivo es mas probable (2.53 veces mas) que deje el banco, aunque no haya emitido ninguna queja
- Cliente de entre 46-55 años que posee cuenta de ahorros y tiene un producto también es probable que busque otros bancos (hipótesis: alternativas mejores en el mercado)

La recomendación que se haría a la entidad bancaria es centrar sus esfuerzos en ofrecer unas condiciones más competitivas a clientes entre los 46-55 años, que tengan cuenta de ahorros y un solo producto. El hecho de que no emitan quejas no significa que sea más probable que se queden.

Patrones de permanencia en el banco

De la misma manera que interesa saber qué patrones hacen más probable que un cliente se vaya, es muy importante saber qué se está haciendo bien. ¿Qué contenta al cliente y le hace permanecer en el banco?

Dado que el 80% de los clientes no se van, será necesario un nivel de confianza mucho más elevado que en el caso anterior para encontrar reglas significativas. Impondremos un soporte mínimo de 0.1 y una confianza del 90%

```
rules = apriori (data_tr, parameter = list (support=0.1, confidence=0.85, maxlen = 5, minlen=2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.85    0.1    1 none FALSE            TRUE       5     0.1      2
##  maxlen target  ext
##       5  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 700
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[73 item(s), 7000 transaction(s)] done [0.01s].
## sorting and recoding items ... [58 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5
```

```
## Warning in apriori(data_tr, parameter = list(support = 0.1, confidence = 0.85,
## : Mining stopped (maxlen reached). Only patterns up to a length of 5 returned!

##  done [0.07s].
## writing ... [5386 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
summary(rules)
```

```
## set of 5386 rules
##
## rule length distribution (lhs + rhs):sizes
##    2    3    4    5
##   13  379 1897 3097
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     2.0     4.0     5.0     4.5     5.0     5.0
##
## summary of quality measures:
##     support         confidence        coverage           lift
##  Min.   :0.1000   Min.   :0.8500   Min.   :0.1003   Min.   :1.054
##  1st Qu.:0.1109   1st Qu.:0.8786   1st Qu.:0.1210   1st Qu.:1.154
##  Median :0.1274   Median :0.9251   Median :0.1394   Median :1.205
##  Mean   :0.1435   Mean   :0.9207   Mean   :0.1562   Mean   :1.240
##  3rd Qu.:0.1588   3rd Qu.:0.9670   3rd Qu.:0.1731   3rd Qu.:1.325
##  Max.   :0.6136   Max.   :1.0000   Max.   :0.6601   Max.   :4.507
##      count
##  Min.   : 700
##  1st Qu.: 776
##  Median : 892
##  Mean   :1004
##  3rd Qu.:1112
##  Max.   :4295
##
## mining info:
##     data ntransactions support confidence
##  data_tr          7000     0.1       0.85
##                                                                                                  call
##  apriori(data = data_tr, parameter = list(support = 0.1, confidence = 0.85, maxlen = 5, minlen = 2))
```

```
reglas_Noredund <- rules[!is.redundant(x = rules, measure = "confidence")]
filtrado_reglas <- subset(x = reglas_Noredund,
                          subset = rhs %pin% "Exited=0")
filtrado_reglas_ordenadas <- sort(filtrado_reglas, by = "lift")
inspect(head(filtrado_reglas_ordenadas,10))
```

```
##      lhs                                rhs          support confidence  coverage     lift count
## [1]  {Age=26-35,
##       HasCrCard=1,
##       NumOfProducts_grupo=2}         => {Exited=0} 0.1154286  0.9439252 0.1222857 1.190536   808
## [2]  {IsActiveMember=1,
##       Balance=Muy Bajo (0-1K),
```

```
##       ComplaintsCount_bin=No_queja,
##       NumOfProducts_grupo=2}         => {Exited=0} 0.1031429  0.9425587 0.1094286 1.188813   722
## [3]  {IsActiveMember=1,
##       Balance=Muy Bajo (0-1K),
##       NumOfProducts_grupo=2}         => {Exited=0} 0.1247143  0.9407328 0.1325714 1.186510   873
## [4]  {Age=26-35,
##       NumOfProducts_grupo=2}         => {Exited=0} 0.1611429  0.9384359 0.1717143 1.183613  1128
## [5]  {Gender=Male,
##       Balance=Muy Bajo (0-1K),
##       NumOfProducts_grupo=2}         => {Exited=0} 0.1297143  0.9331963 0.1390000 1.177004   908
## [6]  {Gender=Male,
##       IsActiveMember=1,
##       NumOfProducts_grupo=2}         => {Exited=0} 0.1252857  0.9300106 0.1347143 1.172986   877
## [7]  {TransactionFrequency=31-40,
##       Balance=Muy Bajo (0-1K),
##       NumOfProducts_grupo=2}         => {Exited=0} 0.1018571  0.9259740 0.1100000 1.167895   713
## [8]  {Geography=France,
##       Balance=Muy Bajo (0-1K),
##       NumOfProducts_grupo=2}         => {Exited=0} 0.1397143  0.9252602 0.1510000 1.166995   978
## [9]  {HasCrCard=1,
##       Balance=Muy Bajo (0-1K),
##       NumOfProducts_grupo=2}         => {Exited=0} 0.1657143  0.9243028 0.1792857 1.165787  1160
## [10] {MaritalStatus=Married,
##       Balance=Muy Bajo (0-1K),
##       NumOfProducts_grupo=2}         => {Exited=0} 0.1191429  0.9225664 0.1291429 1.163597   834
```

En este caso los lift son mas bajos, pero tiene sentido si partimos de la premisa del desbalanceo (80% de Exited es 0). Una confianza de 0.94 nos dice que un cliente con esos antecedentes tiene un 94% de probabilidades de quedarse contra el 80% si no tenemos los datos.

- En todas las reglas aparecen clientes con 2 productos.
- Clientes con un balance muy bajo (<1.000 € en la cuenta) son propensos a quedarse.
- Clientes franceses con dos productos y poco dinero en la cuenta son fieles al banco.

Diversas reglas refuerzan la tesis de que clientes activos y con más de un producto son más fieles al banco. Una buena iniciativa sería promover mediante ofertas que el cliente esté más conectado al banco mediante la titularidad de más de un producto.

Conclusiones reglas asociativas con salida del banco como consecuente

Clientes con potencial abandono

- Un cliente de entre 46-55 años inactivo es mas probable (2.53 veces mas) que deje el banco, aunque no haya emitido ninguna queja
- Cliente de entre 46-55 años que posee cuenta de ahorros y tiene un producto también es probable que busque otros bancos (hipótesis: alternativas mejores en el mercado)

Clientes que tienden a ser fieles al banco

Diversas reglas refuerzan la tesis de que clientes activos y con más de un producto son más fieles al banco. Una buena iniciativa sería promover mediante ofertas que el cliente esté más conectado al banco mediante la titularidad de más de un producto.

Clientes casados o con poco dinero tienden a quedarse en el banco.

Significancia de variables

Se observa que las variables que aparecen en las reglas de asociación mediante el estudio de la base de datos en forma transaccional coinciden con las declaradas como significativas para la explicación de Exited en otros puntos del proyecto habiendo utilizado otros métodos estadísticos.

Esto es buena señal, dado que se ha podido comprobar la relevancia de estas variables mediante dos caminos estadísticos independientes entre sí, hecho que nos permite reforzar el planteamiento realizado anteriormente sobre significancia estadística.