Association Rules

Contents

1 Adaptación de los datos

```
data_ar <- subset(data_transformada, select = -c(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:

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

Numéricas (transformación 1 a 1 con cortes personalizados)

```
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
```

```
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).

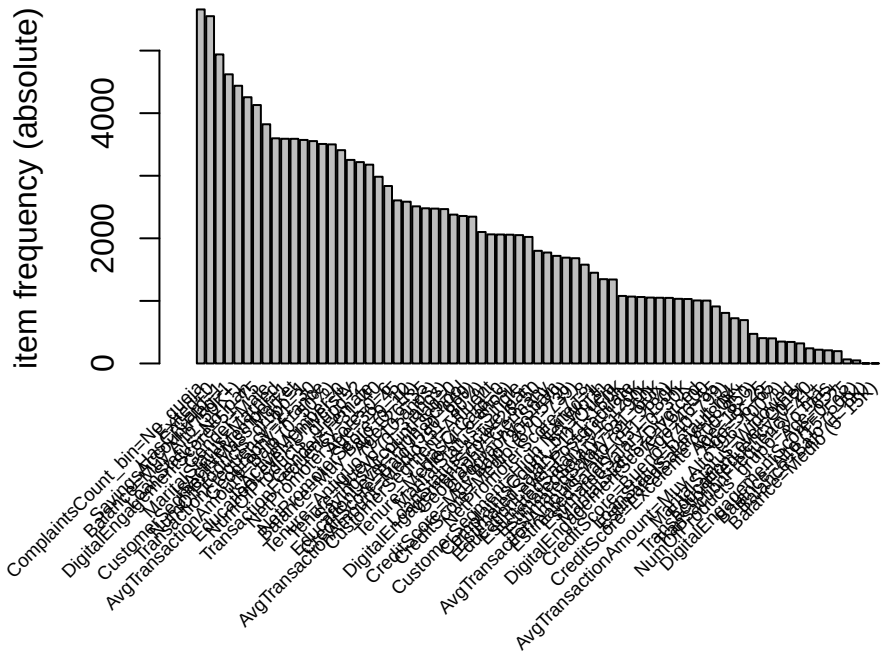
2 Parametrización

2.1 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 obtener algo de información sobre nuestra base de datos transaccional que nos ayude a fijar un soporte mínimo adecuado.

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



Normalmente, a la vista del gráfico, se establecería un soporte mínimo de 0.05 o similar dependiendo del caso de estudio. En este caso, el desbalanceo nos obliga a ser más permisivos.

Se es consciente de la problemática que supone el desbalanceo para la variable exited, por lo que será necesario aplicar un soporte mínimo **extremadamente bajo**. Se ha escogido 0.0125.

2.2 Confianza

Indica qué porcentaje de las veces que aparece el antecedente, también aparece el consecuente.Para nuestro caso, dado el desbalanceo, también se fijará una confianza permisiva. Se ha escogido confianza=0.6, lo que significa que cuando se da la condición del antecedente, en el 60% de los casos se cumple el consecuente.

De este modo se ponen unos filtros laxos para detectar reglas donde el consecuente sea que el cliente se va del banco (Exited=1) dada la circunstancia del desbalanceo.

3 Creación conjunto de reglas a explorar

3.1 Reglas de asociación

Se utilizan los parámetros establecidos y justificados en el apartado anterior con la esperanza de obtener reglas que den informacion valiosa sobre las circunstancias que llevan a un cliente a quedarse o irse del banco. Los parámetros se fijan para cualquier tipo de búsqueda en lhs o consecuente.

Los itemsets a estudiar serán de longitud entre 2 y 5.

```r
itemsets <- apriori(data = data_tr,
                    parameter = list(support = 0.0125,
                                     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  0.0125      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: 87
##
## 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 = data_tr, parameter = list(support = 0.0125, minlen =
## 1, : Mining stopped (maxlen reached). Only patterns up to a length of 5
## returned!

##  done [0.88s].
## sorting transactions ... done [0.00s].
## writing ... [509325 set(s)] done [0.06s].
## creating S4 object  ... done [0.11s].
```

```r
summary(itemsets)
```

```
## set of 509325 itemsets
##
## most frequent items:
## ComplaintsCount_bin=No_queja                    Exited=0
##                       106997                      106712
##                  HasCrCard=1         SavingsAccountFlag=1
##                        92940                       84275
##      Balance=Muy Alto (50K+)                     (Other)
##                        82992                     1884816
##
## element (itemset/transaction) length distribution:sizes
##      1      2      3      4      5
##     69   1941  24183 133428 349704
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
##   1.000   4.000   5.000   4.631   5.000   5.000
##
## summary of quality measures:
##     support            count
##  Min.   :0.01257   Min.   :  88.0
##  1st Qu.:0.01557   1st Qu.: 109.0
##  Median :0.02057   Median : 144.0
##  Mean   :0.02833   Mean   : 198.3
##  3rd Qu.:0.03143   3rd Qu.: 220.0
##  Max.   :0.80829   Max.   :5658.0
##
## includes transaction ID lists: FALSE
##
## mining info:
##     data ntransactions support confidence
##  data_tr          7000  0.0125          1
##
##  apriori(data = data_tr, parameter = list(support = 0.0125, minlen = 1, maxlen = 5, target = "freque
```

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.

```
rules = apriori (data_tr, parameter = list (support=0.0125, confidence=0.6, maxlen = 5, minlen=2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    0.1    1 none FALSE            TRUE       5  0.0125      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: 87
##
## 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 = 0.0125, confidence =
## 0.6, : Mining stopped (maxlen reached). Only patterns up to a length of 5
## returned!

##  done [0.56s].
## writing ... [728420 rule(s)] done [0.09s].
## creating S4 object  ... done [0.29s].
```

```
summary(rules)
```

```
## set of 728420 rules
##
## rule length distribution (lhs + rhs):sizes
##      2      3      4      5
##    406  10904 123723 593387
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   5.000   5.000   4.799   5.000   5.000
##
## summary of quality measures:
##     support          confidence        coverage            lift
##  Min.   :0.01257   Min.   :0.6000   Min.   :0.01257   Min.   : 0.7423
##  1st Qu.:0.01643   1st Qu.:0.6656   1st Qu.:0.02229   1st Qu.: 1.0147
##  Median :0.02286   Median :0.7359   Median :0.03100   Median : 1.1321
##  Mean   :0.03208   Mean   :0.7511   Mean   :0.04308   Mean   : 1.2725
##  3rd Qu.:0.03657   3rd Qu.:0.8225   3rd Qu.:0.04929   3rd Qu.: 1.3283
##  Max.   :0.64100   Max.   :1.0000   Max.   :0.80829   Max.   :14.8760
##      count
##  Min.   :  88.0
##  1st Qu.: 115.0
##  Median : 160.0
##  Mean   : 224.6
##  3rd Qu.: 256.0
```

```
##  Max.   :4487.0
##
## mining info:
##     data ntransactions support confidence
##  data_tr          7000  0.0125        0.6
##                                                                                                                ca
##  apriori(data = data_tr, parameter = list(support = 0.0125, confidence = 0.6, maxlen = 5, minlen = 2
```

Se ha generado ~728.000 reglas, la mayoría (+615.000) compuestas por 4 o 5 items.

3.2 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 201409 rules
```

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

4 Detección de patrones

4.1 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))
```

```
##     lhs                                rhs           support confidence   coverage     lift count
## [1] {ComplaintsCount_bin=No_queja,
##      NumOfProducts_grupo=3 o más}   => {Exited=1} 0.01628571  0.6867470 0.02371429 3.315330   114
## [2] {NumOfProducts_grupo=3 o más}   => {Exited=1} 0.02057143  0.6857143 0.03000000 3.310345   144
## [3] {Gender=Female,
##      Age=46-55,
##      IsActiveMember=0,
##      NumOfProducts_grupo=1}         => {Exited=1} 0.01328571  0.6326531 0.02100000 3.054187    93
## [4] {Age=46-55,
```

```
##       Geography=Germany,
##       Balance=Muy Alto (50K+),
##       NumOfProducts_grupo=1}         => {Exited=1} 0.01300000  0.6148649 0.02114286 2.968313    91
## [5] {Age=46-55,
##       IsActiveMember=0,
##       ComplaintsCount_bin=No_queja,
##       NumOfProducts_grupo=1}         => {Exited=1} 0.01900000  0.6018100 0.03157143 2.905289   133
```

Vemos que, pese a haber fijado parámetros muy permisivos, el lift de las que han pasado el filtro es muy alto. Esto se debe al desbalanceo, que provoca que la regla se de pocas veces (partimos de solo un 20% de Exited=1), pero que proporcionalmente da una regla muy significativa.

Un lift de 3 indica que un cliente con los antecedentes que especifica la regla es 3 veces más propenso a abandonar el banco que la media.

Información de valor que se obtiene: - Un cliente que tiene 3.3 productos o más es 3 veces más propenso a abandonar el banco. - El hecho de que no haya emitido quejas no aumenta su probabilidad de permanencia. Al contrario: la disminuye - Mujeres de entre 46-55 años inactivas y con un solo producto son 3 veces más propensas a dejar el banco. - Clientes alemanes de entre 46-55 años con un balance de +50k en el banco y un solo producto son 3 veces más propensos a dejar el banco. - Clientes de entre 46-55 años, inactivos con un solo producto y sin quejas también son 2.9 veces más propensos a dejar el banco.

Conclusiones/recomendaciones: - Prestar atención a la franja de edad de entre 46-55 años. Ver qué pasa para ese segmento de población en el mercado alemán y en las mujeres. - Revisar las condiciones que ofrece el banco a clientes con 3 productos o más - Clientes inactivos de entre 46-55 años abandonan con facilidad el banco.

4.2 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?

Se utlizan las mismas reglas encontradas con los parámetros fijados y justificados previamente.

```
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]  {LoanStatus=Default risk,
##       NetPromoterScore=0-6,
##       Age=26-35,
##       NumOfProducts_grupo=2}         => {Exited=0} 0.01257143  1.0000000 0.01257143 1.261261    88
## [2]  {Age=26-35,
##       DigitalEngagementScore=76-100,
##       Balance=Muy Bajo (0-1K),
##       NumOfProducts_grupo=2}         => {Exited=0} 0.01428571  1.0000000 0.01428571 1.261261   100
## [3]  {Age=26-35,
##       MaritalStatus=Married,
##       DigitalEngagementScore=76-100,
##       NumOfProducts_grupo=2}         => {Exited=0} 0.01300000  1.0000000 0.01300000 1.261261    91
## [4]  {Age=26-35,
##       DigitalEngagementScore=76-100,
```

```
##       ComplaintsCount_bin=No_queja,
##       NumOfProducts_grupo=2}          => {Exited=0} 0.02214286  0.9935897 0.02228571 1.253176   155
## [5]  {Age=26-35,
##       HasCrCard=1,
##       DigitalEngagementScore=76-100,
##       NumOfProducts_grupo=2}          => {Exited=0} 0.01814286  0.9921875 0.01828571 1.251408   127
## [6]  {Age=26-35,
##       DigitalEngagementScore=76-100,
##       SavingsAccountFlag=1,
##       NumOfProducts_grupo=2}          => {Exited=0} 0.01742857  0.9918699 0.01757143 1.251007   122
## [7]  {LoanStatus=Default risk,
##       Age=26-35,
##       NumOfProducts_grupo=2}          => {Exited=0} 0.01642857  0.9913793 0.01657143 1.250388   115
## [8]  {LoanStatus=No loan,
##       Age=26-35,
##       DigitalEngagementScore=76-100,
##       NumOfProducts_grupo=2}          => {Exited=0} 0.01585714  0.9910714 0.01600000 1.250000   111
## [9]  {Age=26-35,
##       Geography=France,
##       DigitalEngagementScore=76-100,
##       NumOfProducts_grupo=2}          => {Exited=0} 0.01471429  0.9903846 0.01485714 1.249134   103
## [10] {Age=18-25,
##       DigitalEngagementScore=51-75,
##       Balance=Muy Bajo (0-1K)}        => {Exited=0} 0.01457143  0.9902913 0.01471429 1.249016   102
```

En este caso los lift son mas bajos, pero tiene sentido si partimos de la premisa del desbalanceo (80% de Exited es 0). De hecho, dado que el consecuente que estudiamos es ~80% de la población, el lift teórico máximo es de ~1.25 para este caso. Vemos que es prácticamente así.

Algo que llama poderosamente la atención es la confianza de estas reglas, 1 o muy cercana. Esto quiere decir que cuando se da el antecedente SIEMPRE se da el consecuente (el cliente se queda en el banco).

Información de valor:

- Clientes de entre 18-35 años que tengan un índice de uso digitalde entre 76-100 y sean titulares de 2 productos es prácticamente seguro que se queden en el banco.
- Si al hecho de pertenecer a la franja de 18-35 años y de tener dos productos se le añade una de las siguientes características, sigue siendo casi seguro que el cliente se quedará (99-100%): estar casado/a, tener tarjeta de crédito, estado de préstamos "sin préstamos" o "riesgo estándar"
- Clientes de entre 18-25 años con un balance de <1000 en la cuenta se quedarán con un 99% de confianza.

Diversas reglas refuerzan la tesis de que clientes de entre 18-35 años, con 2 productos y casados o con solvencia para pedir préstamos es prácticamente seguro que se quedarán en el banco

5 Significancia de variables y conclusiones de Association Rules

Se observa que todas las variables recogidas como significativas como producto de diferentes tests estadísticos aparecen en las reglas que tienen como consecuente permanecer o abandonar el banco.

El número de quejas también aparece en antecedentes de reglas con Exited=1 con su categoría "0". Hay que recordar del análisis exploratorio que esta supone el 25% de los datos.

Digital_Engagement_score aparece en reglas con consecuente Exited=0 que tienen una confianza de prácticamente el 100%, pero lo hacen siempre en su categoría 75-100 y acompañadas de la franja de edad 26-35.

Por si sola, su significancia quedó categóricamente rechazada tras obtener un p-valor de 0.35, pero sí que puede decirse que clientes de esa franja con un uso digital alto son propensos a abandonar el banco.

```
## # A tibble: 24 x 4
##    Age   DigitalEngagementScore     n prop_Exited
##    <fct> <fct>                  <int>       <dbl>
##  1 56-65 26-50                    114      0.447
##  2 46-55 51-75                    526      0.430
##  3 56-65 76-100                    62      0.419
##  4 46-55 26-50                    241      0.419
##  5 46-55 76-100                   139      0.410
##  6 46-55 0-25                       5      0.4
##  7 56-65 0-25                       5      0.4
##  8 56-65 51-75                    220      0.341
##  9 26-35 0-25                      19      0.263
## 10 65+   76-100                    21      0.238
## 11 36-45 76-100                   369      0.230
## 12 36-45 26-50                    644      0.208
## 13 36-45 51-75                   1567      0.202
## 14 18-25 0-25                      5      0.2
## 15 65+   0-25                      5      0.2
## 16 36-45 0-25                     26      0.154
## 17 65+   26-50                    60      0.15
## 18 18-25 26-50                    100      0.13
## 19 26-35 26-50                    641      0.129
## 20 26-35 51-75                   1464      0.124
## 21 18-25 76-100                    58      0.121
## 22 65+   51-75                    111      0.108
## 23 26-35 76-100                   356      0.0983
## 24 18-25 51-75                    242      0.0785
```

Con un análisis de proporciones rápido por grupos vemos como las edades de entre 46-65 son mucho más propensas a dejar el banco que la media (0.2), y que gente de corta edad es bastante menos probable que abandone el banco. Esto concuerda con algunas de las conclusiones obtenidas.

También se observa que, hasta los 35 años, si observamos el número de personas por categoría de uso digital es una franja de edad que tiene mucho uso digital.

Clientes de +65 años, independientemente del uso digital, no suelen moverse de banco.

Por tanto se validan las variables escogidas previamente como significativas.

Además se ha encontrado relación entre la puntuación de uso digital, la edad y el abandono o no del banco.

Un posible consejo para el banco sería centrarse en hacer más cómoda la experiencia a clientes de avanzada edad a la vez que se apuesta por una digitalización generalizada para clientes jóvenes.(Además, la edad fue la variable más significativa de todas en un principio).