Análisis del Abandono en el Sector Bancario

Predicción del abandono de clientes

Albacete, 17 de Noviembre de 2016

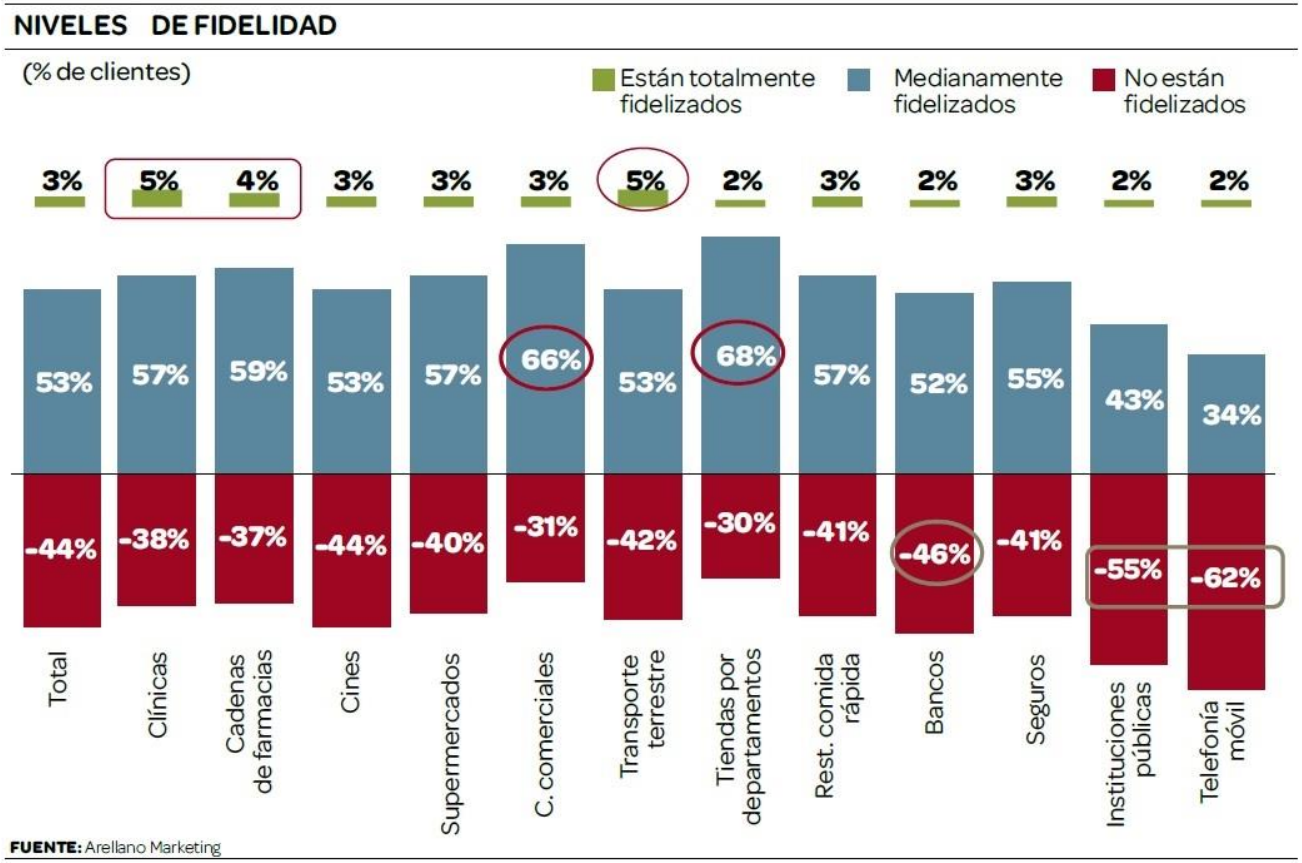
I. INDICE : Modelo Abandonos

Usuarios-R

I. Comprensión del Negocio: Modelo Abandonos

Usuarios-R

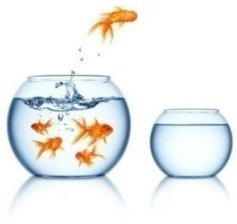
Presente en la estrategia de cualquier Empresa, las **tasas de abandono se disparan** en todos los sectores y compañías. Si hablamos de telecomunicaciones, las tasas **superan en algunos casos el 30%.** Si hablamos **de banca y seguros, oscila entre el 7 y el 18%.**



Usuarios-R

¿Tipos de abandono?

Nos centraremos en el estudio del abandono voluntario y sus patrones de comportamiento



ABANDONO

Involuntario

(Más fácil de identificar)

- Fallecimiento
- Morosidad
- Fraude
- …

Voluntario

(Difícil de identificar)

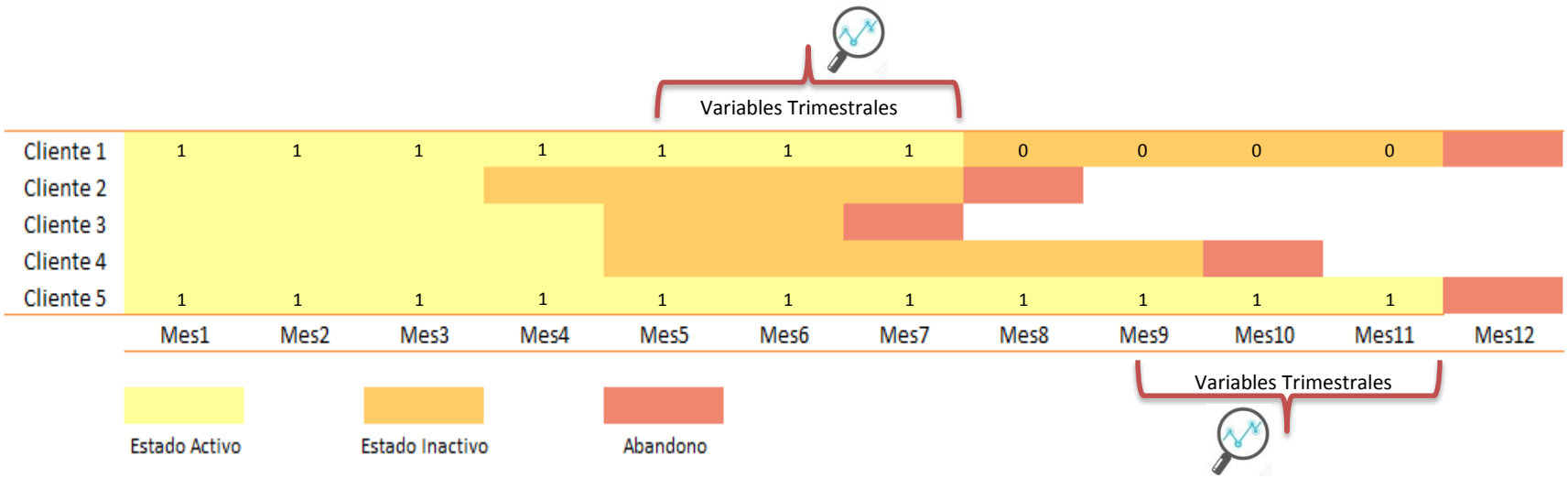
- Decisión del cliente
- Cancelar productos, reducir ingresos, etc..

II. Definición de Abandono: Modelo Abandonos

¿Tipos de abandono?

Abandono Real (Cancelación de todos los contratos) **+** **Abandono Técnico** (Disminución de posiciones y actividad) **=** **Abandono Total**

Establecemos el **momento del abandono** para cada cliente en un **horizonte temporal estable y completo**, que nos permita determinar su patrón de comportamiento antes de producirse el evento. El conocimiento profundo de estos clientes y su comportamiento es vital para establecer un patrón de fuga.



III. Análisis de Datos: Modelo Abandonos

Usuarios-R

1 **Aprovisionamiento**

Descripción

- Identificación de fuentes de datos
- Depuración de datos

2 **Pre-Procesamiento**

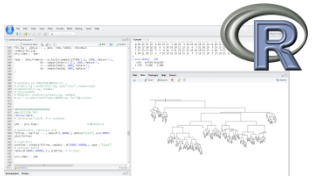
Descripción

- Identificación del cliente que contrata préstamo personal
- Análisis de las variables.
- Selección variables candidatas
- Análisis de las variables (missing, outliers, correlaciones, varianzas cero, etc…)

3 **Modelización**

Descripción

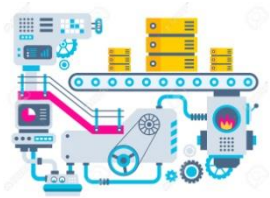
- Selección de modelos predictivos
- Validación de los modelos
- Selección del modelo final
- Creación de la segmentación de valor



4 **Producción**

Descripción

- Puesta en producción
- Calculo de las probabilidades de abandono de clientes



❑ Fichero de datos: Abandono de Clientes

Partimos de un dataset anonimizado con **62 variables**, agrupadas en variables de **tenencia, tendencia, saldos y movimientos** de los distintos productos financieros a contratar. Adicionalmente se incluyen variables socio demográficas .

Variables	Descripción	Valor	Tipo Variable
ID	Indentificador de cliente		Numerico
IND_PROD_01-17	Tenencia de los distintos productos financieros	0 = No tiene, 1 = Tiene, 2 = Ha tenido	Categorica
TEND_PROD_01-10	Tendencia de los saldos en los principales productos financieros	0 = No tiene, 1 = Tiene, 2 = Ha tenido	Categorica
IMP_PROD_01-22	Importe de los saldos de los distintos productos financieros	0 = No tiene, 1 = Tiene, 2 = Ha tenido	Categorica
NUM_PROD_01-09	Operatividad de los productos financieros		Numerica
SOCIODEMO_01-04	Variables sociodemográficas relacionas con el cliente		Numerica
INDTARGET	Variable objetivo, es una variable ordinal y determina el abandono o no del cliente	1 = Abandono, 0 = No abandono	Categorica

- **Objetivos**: Clasificar a los clientes por su propensión al abandono (v.a clase = Target).
- **Solución**: Creación de un modelo de retención de clientes.
- **Resultado**: Adecuar los esfuerzos en las campañas de marketing en función de los distintos perfiles de clientes que nos permitan definir una estrategia diferenciadora.

❑ Lectura del Dataset

- Definimos el **directorio de trabajo** y **cargamos las librerías** correspondientes.
- **Leemos el fichero** de datos y realizamos una **primera auditoria** de datos.
- Exploración de la **estructura** del Dataset

```
#################################################################
#####    LECTURA DEL DATASET
#################################################################

# Establecemos el directorio de trabajo

getwd()
setwd("C:/Users/INTELIGENCIA NEGOCIO/Desktop/MachineLearning")


# Paquetes a cargar
library(dplyr)

# Lectura del Dataset


churn <- read.table("MODABANDONO.txt", header = TRUE, sep = "\t", quote = "\"", dec = ".",
                    na.strings = "NA", fill = TRUE, comment.char = "",stringsAsFactors = T)

# Asignamos formato

clientfactor <- select(churn,starts_with("ID"),starts_with("IND"), SOCIODEMO_02, SOCIODEMO_04)
clientnumeric <- select(churn,-starts_with("ID"),-starts_with("IND"), -SOCIODEMO_02, -SOCIODEMO_04)
clientfactor<- data.frame(apply(clientfactor, 2, as.factor))
clientnumeric<- data.frame(apply(clientnumeric, 2, as.numeric))
churn <- bind_cols(clientfactor,clientnumeric)

# Estructura del dataset y principales estadisticos

str(churn)
summary(churn)
```

III. Análisis de Datos: Modelo Abandonos

❑ Estructura del Dataset: Churn

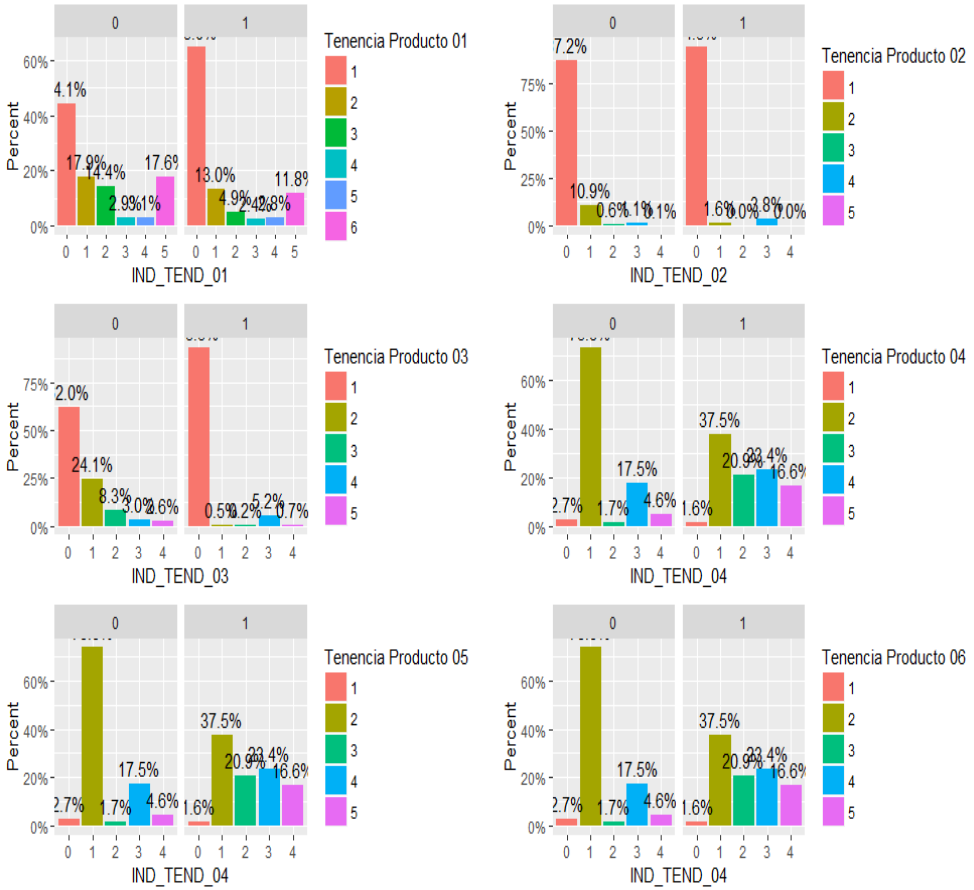
```
> str(churn)
'data.frame':   541423 obs. of  62 variables:
 $ ID          : Factor w/ 541424 levels "      1683","      4989",..: 1 2 3 4
 $ INDTARGET   : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 1 ...
 $ IND_PROD_01 : Factor w/ 3 levels "0","1","2": 1 1 1 2 1 2 2 1 2 1 ...
 $ IND_PROD_02 : Factor w/ 3 levels "0","1","2": 2 2 2 2 2 2 2 2 2 2 ...
 $ IND_PROD_03 : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ IND_PROD_04 : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ IND_PROD_05 : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 2 1 1 1 1 ...
 $ IND_PROD_06 : Factor w/ 3 levels "0","1","2": 1 1 2 1 2 1 1 1 2 1 ...
 $ IND_PROD_07 : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ IND_PROD_08 : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ IND_PROD_09 : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 2 1 1 1 ...
 $ IND_PROD_10 : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ IND_PROD_11 : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ IND_PROD_12 : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 2 1 1 1 ...
 $ IND_PROD_13 : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 2 1 1 1 ...
 $ IND_PROD_14 : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ IND_PROD_15 : Factor w/ 3 levels "0","1","2": 1 1 1 2 1 1 2 2 1 1 ...
 $ IND_PROD_16 : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ IND_TEND_01 : Factor w/ 6 levels "0","1","2","3",..: 1 1 1 2 1 2 6 1 2 1 ...
 $ IND_TEND_02 : Factor w/ 5 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ IND_TEND_03 : Factor w/ 5 levels "0","1","2","3",..: 1 1 1 1 1 3 2 1 1 1 ...
 $ IND_TEND_04 : Factor w/ 5 levels "0","1","2","3",..: 5 2 2 2 2 2 2 4 2 4 ...
 $ IND_TEND_05 : Factor w/ 5 levels "0","1","2","3",..: 5 1 2 2 2 5 4 3 2 1 ...
 $ IND_TEND_06 : Factor w/ 5 levels "0","1","2","3",..: 3 2 2 2 2 2 2 5 2 4 ...
 $ IND_TEND_07 : Factor w/ 5 levels "0","1","2","3",..: 2 1 2 4 2 2 2 3 2 1 ...
 $ IND_TEND_08 : Factor w/ 5 levels "0","1","2","3",..: 2 2 2 2 2 4 4 4 2 2 ...
 $ IND_TEND_09 : Factor w/ 5 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ IND_TEND_10 : Factor w/ 5 levels "0","1","2","3",..: 2 2 2 3 2 5 4 5 5 2 ...
 $ IND_PROD_17 : Factor w/ 3 levels "0","1","2": 1 2 2 1 1 1 1 1 1 1 ...
 $ SOCIODEMO_02: Factor w/ 2 levels " 1"," 2": 1 1 1 1 2 2 1 2 2 2 ...
 $ SOCIODEMO_04: Factor w/ 5 levels "ALTA","BAJA",..: 3 2 3 4 5 4 3 3 4 3 ...
 $ NUM_PROD_02 : num  0 0 0 0 0 0 0 0 0 0 ...
 $ NUM_PROD_03 : num  0 0 0 0 0 0 0 0 0 0 ...
 $ NUM_PROD_04 : num  0 0 0 5 0 0 3 1 0 0 ...
 $ NUM_PROD_05 : num  1 3 9 7 4 20 28 2 9 0 ...
```

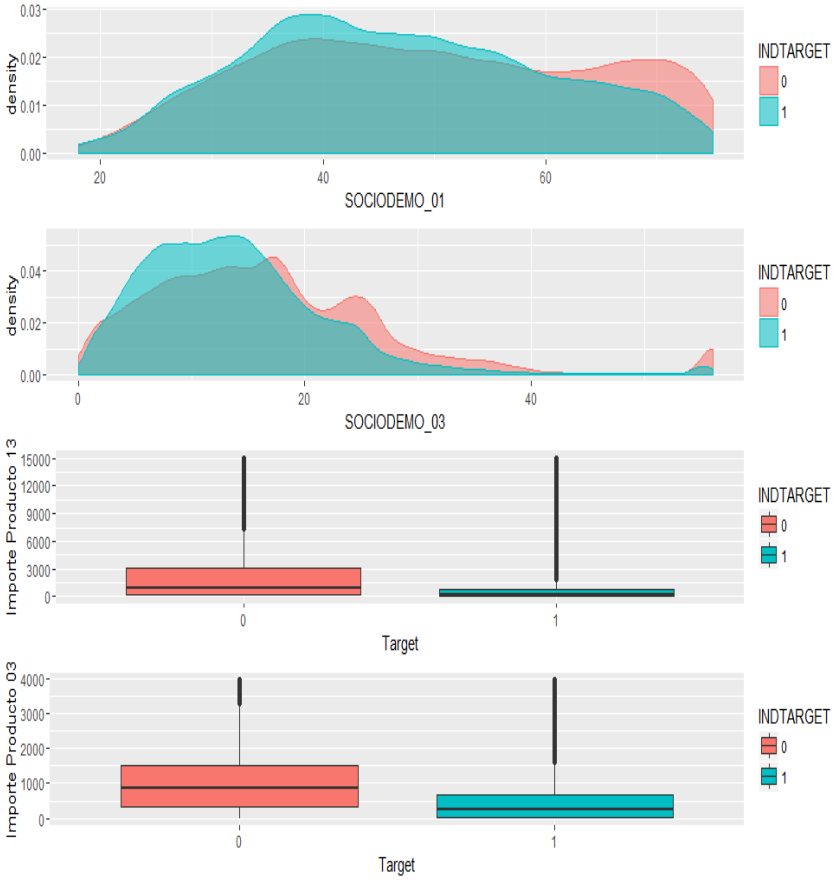
```
> summary(churn[,-1])
 INDTARGET  IND_PROD_01 IND_PROD_02 IND_PROD_03 IND_PROD_04 IND_PROD_05 IND_PROD_06 IND_PROD_07
 0:522345   0:241788    0:   584    0:468316    0:532959    0:401686    0:451838    0:512456
 1: 19078   1:298778    1:540785    1: 64321    1:  7922    1:124635    1: 87983    1: 28320
            2:   857    2:    54    2:  8786    2:   542    2: 15102    2:  1602    2:   647


 IND_PROD_08 IND_PROD_09 IND_PROD_10 IND_PROD_11 IND_PROD_12 IND_PROD_13 IND_PROD_14 IND_PROD_15
 0:512665    0:463318    0:537061    0:533365    0:475777    0:451462    0:449406    0:268004
 1: 25939    1: 76845    1:  3556    1:  7028    1: 63309    1: 86238    1: 88341    1:264536
 2:  2819    2:  1260    2:   806    2:  1030    2:  2337    2:  3723    2:  3676    2:  8883


 IND_PROD_16 IND_TEND_01 IND_TEND_02 IND_TEND_03 IND_TEND_04 IND_TEND_05 IND_TEND_06 IND_TEND_07
 0:507004    0:242645    0:473754    0:341601    0: 14593    0: 35187    0: 14593    0: 35187
 1: 34046    1: 96038    1: 57408    1:126127    1:391459    1:203265    1:391459    1:216238
 2:   373    2: 76157    2:  2931    2: 43175    2: 12627    2: 68103    2: 13470    2: 41578
             3: 15559    3:  6619    3: 16706    3: 95707    3:139203    3: 87368    3:162702
             4: 16826    4:   711    4: 13814    4: 27037    4: 95665    4: 34533    4: 85718
             5: 94198
 IND_TEND_08 IND_TEND_09 IND_TEND_10 IND_PROD_17 SOCIODEMO_02          SOCIODEMO_04
 0:   786    0:533253    0: 16456    0:526357     1:364276    ALTA           : 86602
 1:279595    1:  7245    1:182704    1: 14728     2:177147    BAJA           :149602
 2: 82705    2:   569    2:102924    2:   338                 MEDIA          :244694
 3:109618    3:   346    3:125336                             MUY ALTA       : 44448
 4: 68719    4:    10    4:114003                             SIN COMPETENCIA: 16077

  NUM_PROD_02        NUM_PROD_03       NUM_PROD_04       NUM_PROD_05       NUM_PROD_06
 Min.   :  0.0000   Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   :  0.000
 1st Qu.:  0.0000   1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:  3.00   1st Qu.:  1.000
 Median :  0.0000   Median :  0.00   Median :  0.00   Median :  8.00   Median :  2.000
 Mean   :  0.7875   Mean   :  0.83   Mean   :  3.89   Mean   : 12.15   Mean   :  2.615
 3rd Qu.:  0.0000   3rd Qu.:  0.00   3rd Qu.:  5.00   3rd Qu.: 18.00   3rd Qu.:  3.000
 Max.   :587.0000   Max.   :201.00   Max.   :231.00   Max.   :327.00   Max.   :427.000
```

III. Análisis de Datos: Modelo Abandonos

Usuarios-R

❑ Estructura del Dataset: Churn





❑ Creación del Training/Testing

- Dividimos el conjunto de datos inicial en una muestra de training y testing para controlar el Overfitting.
- Utilizamos **createDataPartition** del **paquete caret** para separar ambas muestras.
- Observaremos la distribución y los problemas de Oversampling que presenta la muestra.

```
# TRAIN y TEST

library(caret)

set.seed(1976)

Index_Partition <- createDataPartition(churn$INDTARGET, p = .7, list = F, times = 1)

train <- churn[Index_Partition,]
test <- churn[-Index_Partition,]

# Dimension del Dataset
dim(train)
```

```
[1] 378997     62
```

```
dim(test)
```

```
[1] 162426     62
```

```
# Distribución Target

prop.table(table(train$INDTARGET))
```

```
         0          1
0.96476225 0.03523775
```

```
prop.table(table(test$INDTARGET))
```

```
         0          1
0.96476549 0.03523451
```

Usuarios-R

❑. Pre-procesamiento: Reducción del conjunto de variables

Conjunto de datos con **predictores** que presencia muy significativa de un **único valor o "varianza casi cero":**

- La **función nearZerovar** del paquete caret nos permite identificar variables con varianza cercana a cero.
- Identificamos aquellas cuya proporción entre las categorías este en 95/5 y con al menos 10 valores. Con este criterio identificamos 21 variables.

```
## Pre-Procesamiento de datos
## Near-Zero Variance Predictor

Ind_Var_Cte <- nearZeroVar(train, freqCut = 99/9, uniqueCut = 10)
Ind_Var_Cte

[1]  4 12 40 43 44 45 47 48 49 52 54 55

names(train[,Ind_NearZeroVar])

 [1] "INDTARGET"    "IND_PROD_02" "IND_PROD_04"
 [4] "IND_PROD_08" "IND_PROD_10" "IND_PROD_11"
 [7] "IND_TEND_09" "IND_PROD_17" "NUM_PROD_03"
[10] "NUM_PROD_08" "IMP_PROD_01" "IMP_PROD_04"
[13] "IMP_PROD_05" "IMP_PROD_08" "IMP_PROD_10"
[16] "IMP_PROD_11" "IMP_PROD_12" "IMP_PROD_15"
[19] "IMP_PROD_17" "IMP_PROD_18" "NUM_PROD_01"


train <- train[,-Ind_Var_Cte]
test <- test[,-Ind_Var_Cte]
```

Usuarios-R

❑. Pre-procesamiento: Reducción del conjunto de variables

Conjuntos de datos con variables muy correlacionadas:

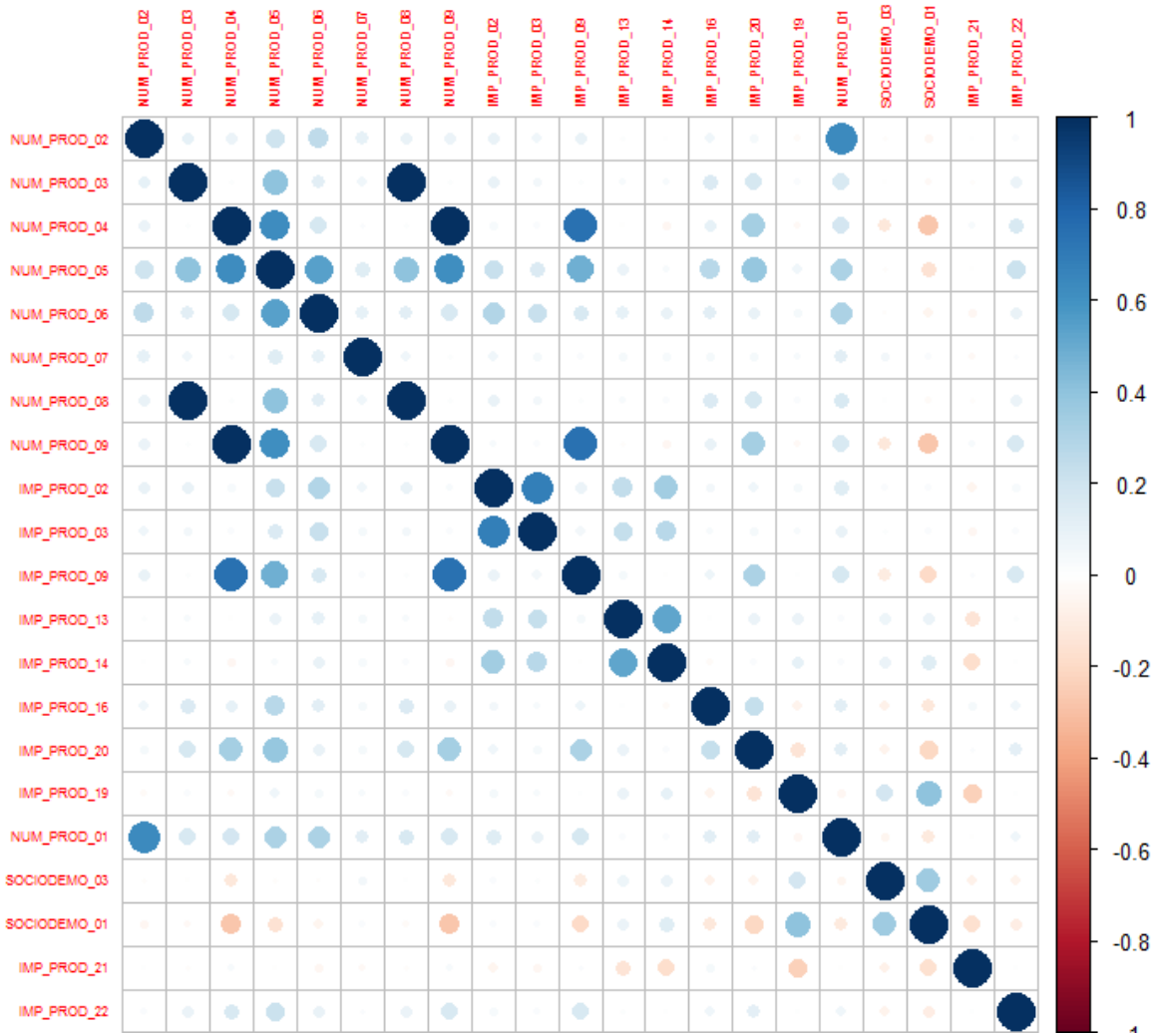
- Eliminaremos **variables** que estén **correlacionadas** (por encima de un umbral mínimo)
- Seleccionamos las dos variables mas correlacionadas y de entre ellas se eliminan la mas correlacionada con el resto.

```
# Matriz de correlaciones de los predictores
nums <- sapply(train, is.numeric)
corr_train <- cor(train[,nums])
library(corrplot) # Visualización correlaciones
# Gráfico de la matriz de correlaciones
corrplot(corr_train, tl.cex = 0.5)
# Correlaciones alta por parejas
indcorr <- findCorrelation(corr_train,cutoff = 0.7)
names(train[,nums][indcorr])
 NUM_PROD_04    NUM_PROD_09    NUM_PROD_03
train <- select(train,-NUM_PROD_04, -NUM_PROD_09, -NUM_PROD_03)
```

- Utilizamos **findCorrelation()** del **paquete caret** para determinar aquellos predictores correlacionados por encima de un umbral.
- La función **corrplot** del paquete **corrplot** nos permite visualizar la matriz de correlaciones.
- Conseguimos identificar 3 variables con correlación por encima de 0.7.
- Se elimina tanto de la muestra de training como del testing.

Usuarios-R

❑ Reducción del conjunto de variables



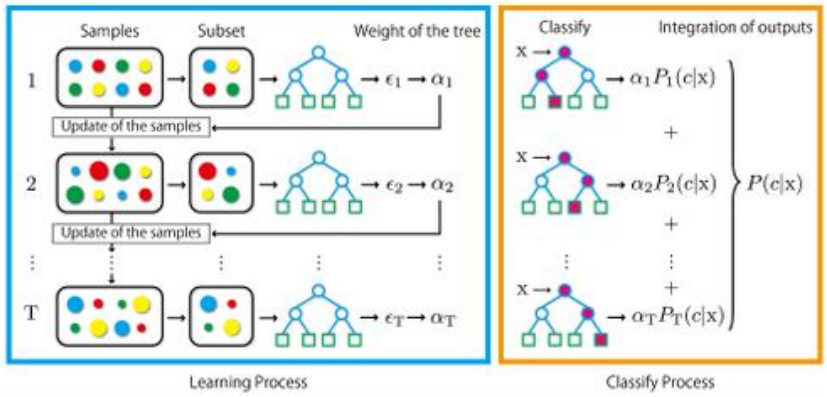
❑. Pre-procesamiento: Problemas con el balanceo de clases - OverSampling

```
## Oversampling - Muestras desbalanceadas

set.seed(1976)
down_train <- downSample(x = train[,-indY], y = train$INDTARGET)

table(down_train$Class)
prop.table(table(down_train$Class))


set.seed(1976)
up_train <- upSample(x = train[, -ncol(train)], y = train$INDTARGET)
table(up_train$Class)
prop.table(table(up_train$Class))


library(ROSE)

set.seed(1976)
rose_train <- ROSE(INDTARGET ~ ., data  = train)$data
table(rose_train$INDTARGET)
```

```
library(DMwR)

set.seed(1976)
smote_train <- SMOTE(INDTARGET ~ ., train, k = 5, perc.over = 100, perc.under = 200)
table(smote_train$Class)

    0    1
4816 4816
```

Usuarios-R

❑. Modelización: Random Forest (librería randomForest)



PRINCIPALES CARACTERÍSTICAS:

- **Random Forest** es un tipo de método **de particionamiento recursivo**, donde cada nodo es dividido en dos nodos hijos, seleccionando el predictor que maximiza la diferencia en abandono en los nodos hijos.
- **Adecuado para pequeñas n y grandes p** (pocos datos y muchas variables), analizando un gran numero de variables, sin tener que hacer selección previa. Utilizado tanto para clasificación como regresión.
- Calculamos subconjuntos de árboles al azar, subconjuntos de X frente a Y. Se basa en muestras bootstrap**. Cada división del árbol está basada en una muestra aleatoria de los predictores.**
- Los árboles no se cortan, son tan largos como sea posible. **No existe la poda.**
- El resultado de un conjunto de árboles **clasificación/regresión se han demostrado mejores para producir predicciones** que los resultados **de un solo árbol de clasificación.**
- **Alto coste computacional**. Establecer un numero de árboles adecuado para que todas las variables puedan participar en la construcción de suficientes árboles.

Usuarios-R

❑. Modelización: Evaluación

El objetivo es **evaluar la calidad del modelo** a través de su capacidad predictiva, mediante la **comparación de valores observados frente a valores predichos**. Evitaremos el sobreajuste (overfitting) evaluando el modelo en una muestra de test diferente, medida de error honesta.

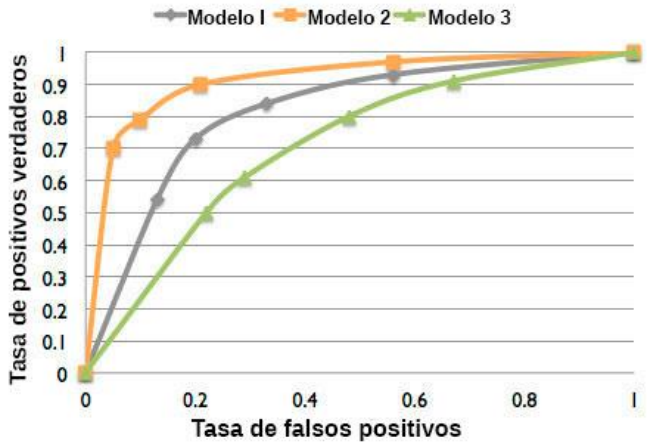
	Predicho		
Observado	**Positivos**	**Negativos**	**Total**
Positivos	**VP**	**FN**	VP+FN
Negativos	**FP**	**VN**	FP+VN
Total	VP+FP	FN+VN	**N**

	Predicho		
Observado	**Positivos**	**Negativos**	**Total**
Positivos	**1500**	**30**	1530
Negativos	**50**	**200**	250
Total	1550	230	**1780**

Accuracy = 95% de clasificaciones correctas
Error Rate = 5% de clasificaciones incorrectas
Sensibilidad = 97% capacidad de detectar VP
Especificidad = 87% capacidad de detectar VN
Valor predictivo positivo = 98%
Valor predictivo negativo = 80%

- **La curva ROC representa :**

  (1-especificidad, sensibilidad)
- Evalúa la **capacidad de clasificación** con dos clases.
- Su forma de medida la define el **área bajo la curva** (AUC) y mide la capacidad de predicción
- Predictor ideal AUC = 1 y aleatorio AUC = 0.5

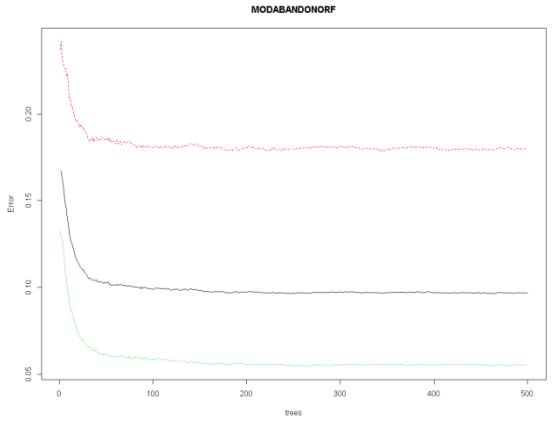


Usuarios-R

❑. Modelización: Random Forest (librería randomForest)

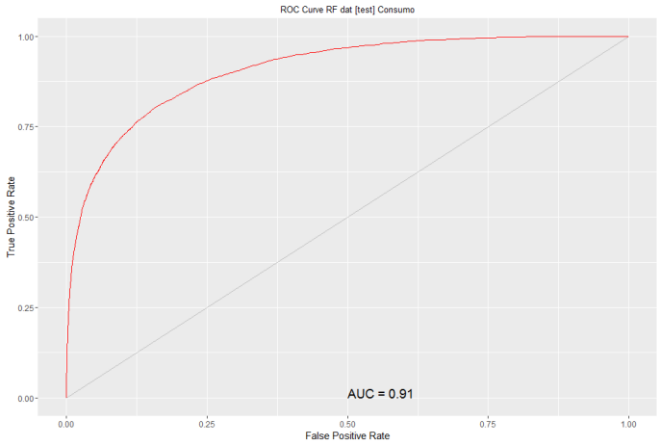
SOLUCIÓN

```
#######################################################
####         MODELIZACIÓN PREDICTIVA AVANZADA
#######################################################

# RANDOM FOREST

library(randomForest)

set.seed(1976)
MODABANDONORF <- randomForest(INDTARGET ~ .,
                              data = smote_train, ntree = 500, nsplit = 10)
MODABANDONORF
plot(MODABANDONORF)
```



EVALUACIÓN

```
## CURVA ROC

#Calculo probabilidades usuando el modelo, seleccionando la segunda columna
probs <- predict(MODABANDONORF, test, type ="prob")[,2]

#Predicción del objeto: pred
pred <- prediction(probs, test$INDTARGET)

#Representando el objeto: perf
pe <- performance(pred, "tpr","fpr")


#Area sobre la curva (AUC)

au <- performance(pred, "auc")@y.values[[1]]

pd <- data.frame(fpr=unlist(pe@x.values), tpr=unlist(pe@y.values))
p <- ggplot(pd, aes(x=fpr, y=tpr))
p <- p + geom_line(colour="red")
p <- p + xlab("False Positive Rate") + ylab("True Positive Rate")
p <- p + ggtitle("ROC Curve RF dat [test] Consumo")
p <- p + theme(plot.title=element_text(size=10))
p <- p + geom_line(data=data.frame(), aes(x=c(0,1), y=c(0,1)), colour="grey")
p <- p + annotate("text", x=0.50, y=0.00, hjust=0, vjust=0, size=5, label=paste("AUC =", round(au, 2)))
print(p)
```



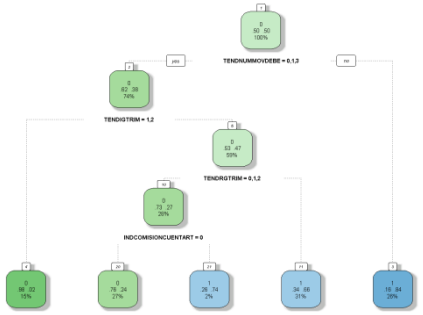
Usuarios-R

❑. Modelización: Árboles de Decisión (Librería Rpart)

ID	IND_PROD_01	IND_PROD_02	IND_TEND_PROD_01	IND_TEND_PROD_02	SOCIO_DEMO_01	IND_TARGET	
1	1	1	0	1	ALTA	SI	
2	0	3	1	1	MEDIA	SI	
3	1	4	2	1	MEDIA	NO	
4	2	2	2	2	ALTA	NO	
5	1	1	3	3	ALTA	SI	
6	0	0	4	4	BAJA	NO	
7	0	2	4	2	ALTA	NO	
8	0	3	2	2	BAJA	SI	
9	2	4	2	3	BAJA	NO	
10	1	1	3	4	BAJA	NO	
11	1	0	4	1	MEDIA	NO	
12	0	1	1	2	MEDIA	NO	
13	0	2	2	3	ALTA	SI	
14	2	2	1	4	BAJA	NO	
15	1	1	1	1	BAJA	NO	
154	0	3	2	2	ALTA	?	SI
167	2	4	2	3	BAJA	?	NO
243	1	1	3	4	ALTA	?	SI

Crea Algoritmo con rpart

Predicción del Abandono de la base de clientes



PRINCIPALES CARACTERÍSTICAS:

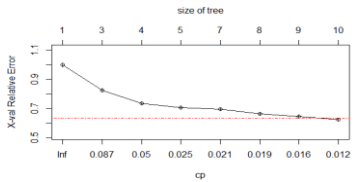
- Es una **técnica supervisada**, la clasificación se hace en función de una variable clase.
- **Partición recursiva** a partir de un conjunto de entrenamiento mostrando una **organización jerárquica.** La estructura tipo : Nodo interior (Pregunta) y Nodo hoja (clasificación).
- Robustos frente a datos faltantes y anómalos.
- **Ventajas:** Permite trabajar con **todo tipo de variables** y su **interpretación es sencilla**.
- **Desventajas:** Dificultad al **elegir árbol óptimo** y necesidad de **gran número de datos** para asegurar nodos de hojas significativas.
- **Tipos de árboles: CART, Chaid, C4.5 y QUEST** (Estructura similar)

❑. Modelización: Árboles de Decisión (Librería Rpart)

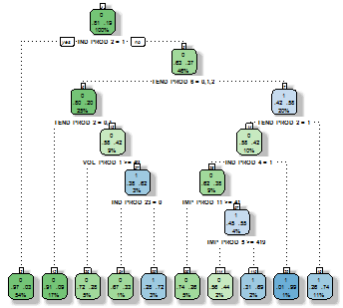
SOLUCIÓN

```
# Modelización avanzada con rpart

#Establecemos la semilla aleatoria
set.seed(1976)

# Cargamos los siguientes parquetes rpart,rattle,rpart.plot y RColorBrewer

library(rpart)
library(rattle)
library(rpart.plot)
library(RColorBrewer)

#Construimos el modelo: tree


table(smote_train$INDTARGET)
summary(smote_train)

tree <- rpart(INDTARGET ~ ., data = smote_train, method = "class")
tree


#Dibujamos el árbol de decisión
fancyRpartPlot(tree)
```

```
printcp(tree)
plotcp(tree,lty=4,col="red")
```

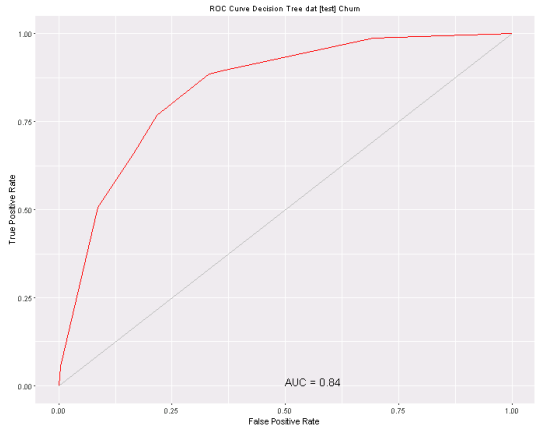



```
#Podar el árbol: Poda
pruned <- prune(tree, cp = 0.010000)
pruned

#Dibujamos el árbol de decisión
fancyRpartPlot(pruned)
```



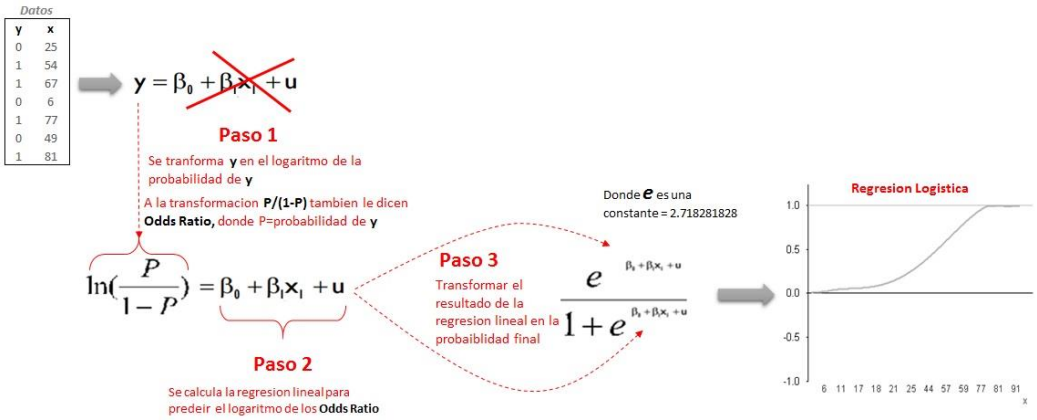
EVALUACIÓN

```
## CURVA ROC

#Calculo probabilidades usuando el modelo, seleccionando la segunda columna
probs <- predict(pruned, test, type ="prob")[,2]

#Predicción del objeto: pred
pred <- prediction(probs, test$INDTARGET)

#Representando el objeto: perf
pe <- performance(pred, "tpr","fpr")


#Area sobre la curva (AUC)

au <- performance(pred, "auc")@y.values[[1]]


pd <- data.frame(fpr=unlist(pe@x.values), tpr=unlist(pe@y.values))
p <- ggplot(pd, aes(x=fpr, y=tpr))
p <- p + geom_line(colour="red")
p <- p + xlab("False Positive Rate") + ylab("True Positive Rate")
p <- p + ggtitle("ROC Curve Decision Tree dat [test] Churn")
p <- p + theme(plot.title=element_text(size=10))
p <- p + geom_line(data=data.frame(), aes(x=c(0,1), y=c(0,1)), colour="grey")
p <- p + annotate("text", x=0.50, y=0.00, hjust=0, vjust=0, size=5, label=paste("AUC =", round(au, 2)))
print(p)
```



Usuarios-R

❑. Modelización: Regresión Logística (librería stats)



PRINCIPALES CARACTERÍSTICAS:

- **La regresión logística** es un método para el ajuste de una curva de regresión y = f(x), con y como variable categórica.
- Los **predictores** pueden ser **continuos, categóricos o mezcla de ambos.**
- La variable **categórica** y, en general, puede asumir **valores diferentes**.
- La **regresión logística en R** se ajusta mediante el comando **glm**(), Generalized Linear Model
- La **distribución** apropiada para este tipo de datos es la **Binomial**

❑. Modelización: Regresión Logística (Librería stast)

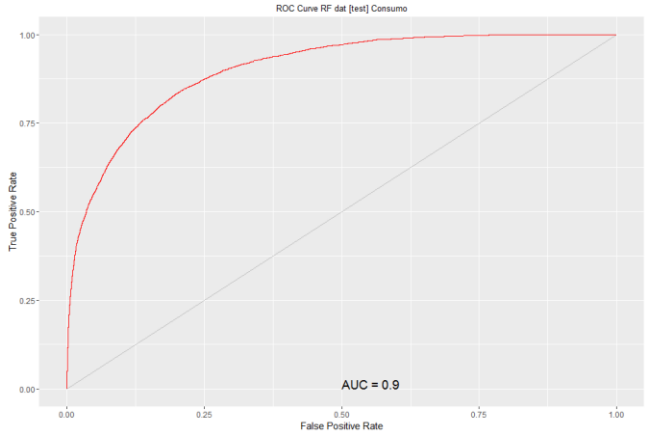
SOLUCIÓN

```
#####################################################
####         MODELIZACIÓN PREDICTIVA AVANZADA
#####################################################


### REGRESIÓN LOGISTICA


MODABANDONORL <- glm(INDTARGET ~.,
                     family=binomial(link='logit'),
                     data=smote_train)

summary(MODABANDONORL)
```

```
Call:
glm(formula = INDTARGET ~ ., family = binomial(link = "logit"),
    data = smote_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.8703  -0.5025  -0.0905   0.3539   3.6258

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    2.0604794  0.1921855  10.721  < 2e-16 ***
IND_PROD_011  -0.2179962  0.0321889  -6.772 1.27e-11 ***
IND_PROD_012   3.2561181  0.2158070  15.088  < 2e-16 ***
IND_PROD_031  -0.2688973  0.0753151  -3.570 0.000357 ***
IND_PROD_032   1.1405631  0.0664118  17.174  < 2e-16 ***
IND_PROD_051   0.2638664  0.0280536   9.406  < 2e-16 ***
IND_PROD_052   0.4983214  0.0482448  10.329  < 2e-16 ***
IND_PROD_061  -2.0088226  0.0710268 -28.283  < 2e-16 ***
IND_PROD_062   2.6690755  0.1184695  22.530  < 2e-16 ***
IND_PROD_091  -2.1704475  0.0968401 -22.413  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 127683  on 93484  degrees of freedom
Residual deviance:  60192  on 93420  degrees of freedom
AIC: 60322

Number of Fisher Scoring iterations: 7
```

EVALUACIÓN

```
## CURVA ROC

library(ROCR)

#Calculo probabilidades usuando el modelo, seleccionando la segunda columna
probs <- predict(MODABANDONORL, test, type="response")

#Predicción del objeto: pred
pred <- prediction(probs, test$INDTARGET)

#Representando el objeto: perf
pe <- performance(pred, "tpr","fpr")


#Area sobre la curva (AUC)

au <- performance(pred, "auc")@y.values[[1]]

pd <- data.frame(fpr=unlist(pe@x.values), tpr=unlist(pe@y.values))
p <- ggplot(pd, aes(x=fpr, y=tpr))
p <- p + geom_line(colour="red")
p <- p + xlab("False Positive Rate") + ylab("True Positive Rate")
p <- p + ggtitle("ROC Curve RF dat [test] Consumo")
p <- p + theme(plot.title=element_text(size=10))
p <- p + geom_line(data=data.frame(), aes(x=c(0,1), y=c(0,1)), colour="grey")
p <- p + annotate("text", x=0.50, y=0.00, hjust=0, vjust=0, size=5, label=paste("AUC =", round(au, 2)))
print(p)
```



Usuarios-R

❑. Selección del mejor modelo predictivo

Random Forest

Regresión Logística

Árbol de decisición

MUCHAS GRACIAS