

APLICACIONES Y CASOS DE USO EMPRESARIAL



PROYECTOS TELEFÓNICA Y NETFLIX

HUGO ÁLVAREZ CHAVES 23055287-H

hugo.alvarez@um.es

MARIO LOSANTOS ALBACETE 04847925-P

mia4@um.es

MOISÉS FRUTOS PLAZA 48488132-S

moises.frutos@um.es

MÁSTER EN BIG DATA

ENTREGADO EL 10 DE JUNIO DE 2017

Índice

Introducción	2
I. Proyecto 1: Movistar en apuros	3
A. Análisis de perfiles de posibles desertores	3
1. Contextualización	3
2. Análisis del conjunto de datos proporcionado	3
3. Preprocesado de variables y selección de registros y atributos	5
4. Análisis del conjunto de datos proporcionado	6
5. Entrenamiento de modelos y evaluación de resultados.	6
6. Selección de modelo y conclusiones.	6
B. Segmentación/creación de grupos entre los desertores y los no desertores	7
C. Diseño de la campaña de incentivos	7
II. Proyecto Netflix	7
A. Recomendador basado en contenido	7
B. Recomendador basado en factorización matricial	8

Introducción

El objetivo de este proyecto es por una parte evaluar la **amenaza** que supone la entrada en el mercado de nuestro país de la plataforma Netflix para Telefónica, y, por el otro, la **oportunidad** que este nuevo mercado representa para esta nueva plataforma, de modo que podamos ver el mismo caso desde dos puntos de vista contrapuestos. Además, tenemos como propósito poner en práctica los conocimientos adquiridos en las distintas asignaturas del máster aplicando éstos al ámbito empresarial.

Para explicar el proyecto con el mayor rigor posible, dentro del tiempo que tenemos, hemos decidido dividir esta memoria en dos apartados principales, correspondientes a los dos proyectos, y sus respectivos subapartados:

- I. Proyecto 1: Movistar en apuros
 - A. Análisis de perfiles de posibles desertores
 - 1. Contextualización
 - 2. Análisis del conjunto de datos proporcionado.
 - 3. Preprocesado de variables y selección de registros y atributos.
 - 4. División del conjunto de datos en conjunto de entrenamiento y conjunto de test.
 - 5. Entrenamiento del modelo y evaluación de los resultados.
 - 6. Conclusiones.
 - B. Segmentación/creación de grupos entre los desertores y los no desertores
 - C. Diseño de la campaña de incentivos
- II. Proyecto 3: Netflix a todas horas
 - A. Recomendador basado en contenido.
 - B. Recomendador basado en factorización matricial.

I. Proyecto 1: Movistar en apuros

A. Análisis de perfiles de posibles desertores

1. Contextualización

La idea es obtener un modelo que describa el comportamiento de permanencia o fuga de clientes y proponer un mecanismo o incentivo para evitar la misma, que pueda cubrir el mayor número de posible de clientes.

2. Análisis del conjunto de datos proporcionado

El conjunto de datos está compuesto por un total de 7044 registros de comportamientos de clientes.

Como conjunto de datos de entrada de nuestro sistema tenemos un fichero CSV con una descripción de un conjunto de clientes.

Para realizar el estudio vamos a utilizar R. El motivo para utilizar R es que nos sentimos cómodos utilizando las distintas librerías que existen para generar modelos para predecir el comportamiento de los clientes objeto de estudio. El código es sencillo:

```
datos <- read.csv("~/Dropbox/Universidad de Murcia/Máster Big
Data/ACUE/Proyecto/Telco-Customer-Churn.csv", header=TRUE, row.names = 1)
summary(datos)
```

gender	SeniorCitizen	Partner	Dependents	tenure
Female:3488	Min. :0.0000	No :3641	No :4933	Min. : 0.00
No : 682				
Male :3555	1st Qu.:0.0000	Yes:3402	Yes:2110	1st Qu.: 9.00
Yes:6361				
	Median :0.0000			Median :29.00
	Mean :0.1621			Mean :32.37
	3rd Qu.:0.0000			3rd Qu.:55.00
	Max. :1.0000			Max. :72.00

MultipleLines	InternetService
OnlineSecurity	
No :3390	DSL :2421
No phone service: 682	Fiber optic:3096
Yes :2971	No :1526
	Yes :2019

OnlineBackup	DeviceProtection
TechSupport	
No :3088	No :3095
No internet service:1526	No internet service:1526
	No internet :3473

```

service:1526
Yes                :2429   Yes                :2422 Yes                :2044

                StreamingTV                StreamingMovies                Contract
                PaperlessBilling
No                :2810   No                :2785   Month-to-month:3875
No :2872
No internet service:1526   No internet service:1526   One year :1473
Yes:4171
Yes                :2707   Yes                :2732   Two year :1695

                PaymentMethod MonthlyCharges TotalCharges Churn
Bank transfer (automatic):1544   Min. : 18.25   Min. : 18.8   No :
5174
Credit card (automatic) :1522   1st Qu.: 35.50   1st Qu.: 401.4
Yes:1869
Electronic check :2365   Median : 70.35   Median :1397.5
Mailed check :1612   Mean : 64.76   Mean :2283.3
                3rd Qu.: 89.85   3rd Qu.:3794.7
                Max. :118.75   Max. :8684.8

                NA's :11

```

Haciendo ingeniería inversa y aplicando imaginación sobre la semántica de los atributos a partir de su nombre, deducimos que los atributos pueden clasificarse en:

- Atributo identificador del cliente:
 - **customerID**: representa al identificador único de cliente.
- Atributos demográficos:
 - **gender**: variable cualitativa que representa el género del cliente.
 - **SeniorCitizen**: variable cualitativa que representa si el cliente es un jubilado.
 - **Partner**: variable cualitativa que indica si el cliente está casado.
 - **Dependents**: variable cualitativa que indica si el cliente tiene personas a su cargo (por ejemplo, hijos).
- Atributos relativos al servicio:
 - **tenure**: variable cuantitativa que representa el número de meses que el cliente lleva contratados
- Atributos relativos a productos contratados:

- **PhoneService**: variable cualitativa que representa si tiene contratada línea de voz
 - **MultipleLines**: variable cualitativa que representa si tiene contratadas más de una línea de voz
 - **InternetService**: variable cualitativa que representa si tiene contratada línea de datos
 - **OnlineSecurity**: variable cualitativa que representa si tiene contratado servicio de seguridad (antivirus suponemos)
 - **OnlineBackup**: variable cualitativa que representa si tiene contratado servicio de backup
 - **DeviceProtection**: variable cualitativa que representa si tiene contratado seguro del dispositivo
 - **TechSupport**: variable cualitativa que representa si tiene contratado servicio de asistencia/soporte, (de internet suponemos)
 - **StreamingTV**: variable cualitativa que representa si tiene contratado servicio de TV
 - **StreamingMovies**: variable cualitativa que representa si tiene contratado servicio de pay per view
- Atributos relativos a la forma de pago y costes de la factura:
 - **Contract**: variable cualitativa que representa el tipo de contrato
 - **PaperlessBilling**: variable cualitativa que representa si recibe factura electrónica
 - **PaymentMethod**: variable cualitativa que representa el tipo de pago/cargo
 - **MonthlyCharges**: variable cuantitativa que representa el coste mensual aproximado
 - **TotalCharges**: variable cuantitativa que representa el coste anual aproximado
 - Atributos objeto de estudio:
 - **Churn**: variable cualitativa que representa si el cliente abandona la empresa o no

3. Preprocesado de variables y selección de registros y atributos

En esta fase debemos hacer un preprocesado del conjunto de datos. Primero, discretizamos el atributo tenure agrupando los datos por periodos de permanencia en la compañía.

```
#Discretizamos tenure en los grupos que consideramos más importantes
tenura.discretizada <- discretize(datos$tenure,
                                method = "fixed",
                                categories = c(0,6,12,24,48,Inf),
```

```

labels = c('Half-Year','Year','Two-
Years','Four-Years','More-Than-Four-Years'),
ordered = TRUE)
ternura.discretizada
datos$tenure <- ternura.discretizada

```

Segundo, vamos a analizar los registros que contienen valores desconocidos. En las tablas anteriores hemos visto que tan sólo el atributo TotalCharges contiene valores NA, concretamente existen 11 registros. Como esos 11 registros de 7044 representan poco más del 0.15% de registros, hemos tomado la elección de eliminar esos registros porque consideramos que hacerlo no altera el modelo a generar y elimina el posible ruido que podríamos introducir al intentar desarrollar un modelo para predecir esos valores desconocidos.

```

library(VIM)
#Funciones para comprobar si hay NA y donde están
proporcionNAPorColumna <- function(columna) {
  x <- countNA(columna)
  y <- dim(columna)[1]
  x/y
}

calcularVectorProporcionNA <- function(datos) {
  i <- 1
  vectorCalculado <- c()
  while(i <= dim(datos)[2]){
    vectorCalculado = append(vectorCalculado,
                             proporcionNAPorColumna(datos[i]))
    i <- i + 1
  }
  vectorCalculado
}

vectorProporcionNA <- calcularVectorProporcionNA(datos)
vectorProporcionNA
[1] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
0.000000000 0.000000000
[8] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
0.000000000 0.000000000
[15] 0.000000000 0.000000000 0.000000000 0.000000000 0.001561834
0.000000000
colnames(datos)[19]
[1] "TotalCharges"
#Efectivamente como hemos dicho antes la columna con NAs es TotalCharges
#Así que eliminamos esos 11 registros que sólo representan el 0.15% del
#conjunto de datos.
datos <- datos[complete.cases(datos),]
> any(is.na(datos))
[1] FALSE
#Con la función complete.cases() ya conseguimos que sólo nos devuelva los
#registros que no contienen NAs, aún así comprobamos con any(is.na())
#Como ahora nos devuelve FALSE significa que ya no hay NAs y podemos
#continuar

```

En tercer lugar, evaluamos de forma global, en todo el conjunto de datos, la gravedad de la amenaza calculando la probabilidad de que un cliente se vaya de la compañía.

```
#Ahora calculamos la probabilidad de desertar dentro de la totalidad del
conjunto de datos
#Para hacernos una idea global de la situación.
tabla <- table(datos$Churn)
total.casos = tabla['Yes'] + tabla['No']
total.casos
probabilidad.deserciones <- tabla['Yes'] / total.casos
probabilidad.deserciones
#Un 26.58% de los clientes están dispuestos a desertar, eso significa que
algo más de 1
#de cada 4 clientes se nos va a ir. Lo cual la amenaza es considerable.
```

La amenaza es realmente grave y real para la compañía ya que potencialmente uno de cada cuatro clientes puede marcharse de la compañía. Por eso, vamos a terminar de realizar el preprocesamiento y realizar un análisis con dos modelos de selección distintos para cotejar entre ambos las variables más significativas debido a la gravedad de la amenaza.

Por último en el preprocesamiento, vamos a reinterpretar las variables cualitativas que tienen 3 valores pero que ya están representadas en otro atributo. Por ejemplo, el atributo *MultipleLines* tiene por posibles valores "No", "No phone service" y "Yes". Como podemos ver los 682 registros que tienen por valor "No" en *PhoneService* tienen por valor "No phone service". Nuestra propuesta es eliminar esa redundancia y codificar "No phone service" como "No" para eliminar la variabilidad y mejorar la capacidad de predicción de nuestro modelo. Hacemos el tratamiento similar para los atributos *MultipleLines*, *OnlineSecurity*, *OnlineBackup*, *DeviceProtection*, *TechSupport*, *StreamingTV* y *StreamingMovies*.

```
library(car)
#Evitar datos redundantes
datos$MultipleLines <- as.factor(recode(datos$MultipleLines, "'No phone
service' = 'No';"))
datos$OnlineSecurity <- as.factor(recode(datos$OnlineSecurity, "'No
internet service' = 'No';"))
datos$OnlineBackup <- as.factor(recode(datos$OnlineBackup, "'No internet
service' = 'No';"))
datos$DeviceProtection <- as.factor(recode(datos$DeviceProtection, "'No
internet service' = 'No';"))
datos$TechSupport <- as.factor(recode(datos$TechSupport, "'No internet
service' = 'No';"))
datos$StreamingTV <- as.factor(recode(datos$StreamingTV, "'No internet
service' = 'No';"))
datos$StreamingMovies <- as.factor(recode(datos$StreamingMovies, "'No
internet service' = 'No';"))
str(datos)
'data.frame':   7032 obs. of  20 variables:
```



```

$ gender      : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 1 2 1 1
2 ...
$ SeniorCitizen : logi FALSE FALSE FALSE FALSE FALSE ...
$ Partner      : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 2
1 ...
$ Dependents   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 1
2 ...
$ tenure      : Ord.factor w/ 5 levels "Half-Year"<"Year"<...: 1 4 1 4
1 2 3 2 4 5 ...
$ PhoneService : Factor w/ 2 levels "No","Yes": 1 2 2 1 2 2 2 1 2
2 ...
$ MultipleLines : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 2 1 2
1 ...
$ InternetService : Factor w/ 3 levels "DSL","Fiber optic",...: 1 1 1 1 2
2 2 1 2 1 ...
$ OnlineSecurity : Factor w/ 2 levels "No","Yes": 1 2 2 2 1 1 1 2 1
2 ...
$ OnlineBackup  : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 2 1 1
2 ...
$ DeviceProtection: Factor w/ 2 levels "No","Yes": 1 2 1 2 1 2 1 1 2
1 ...
$ TechSupport   : Factor w/ 2 levels "No","Yes": 1 1 1 2 1 1 1 1 2
1 ...
$ StreamingTV   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 2 1 2
1 ...
$ StreamingMovies : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 1 1 2
1 ...
$ Contract      : Factor w/ 3 levels "Month-to-month",...: 1 2 1 2 1 1 1
1 1 2 ...
$ PaperlessBilling: Factor w/ 2 levels "No","Yes": 2 1 2 1 2 2 2 1 2
1 ...
$ PaymentMethod : Factor w/ 4 levels "Bank transfer (automatic)",...: 3
4 4 1 3 3 2 4 3 1 ...
$ MonthlyCharges : num 29.9 57 53.9 42.3 70.7 ...
$ TotalCharges   : num 29.9 1889.5 108.2 1840.8 151.7 ...
$ Churn          : Factor w/ 2 levels "No","Yes": 1 1 2 1 2 2 1 1 2
1 ...

```

4. Análisis del conjunto de datos proporcionado

Una vez conocemos los atributos del conjunto de datos y hemos preprocesado la información, es el momento de aplicar dos modelos de selección de atributos para quedarnos con los más significativos, antes de entrenar y evaluar un modelo predictivo. Para ello hemos seleccionado como modelos de selección **Treebag (Bola de árboles)** y **SVMLLinear (Máquina de Soporte Vectorial con kernel lineal)**. Para la selección hemos utilizado el mismo método en ambos modelos: **Validación Cruzada de 5 pliegues**.

```

library(caret)
subsets <- c(1,dim(datos)[2])

#Treebag
set.seed(3456)
ctrl.rfe.treebag <- rfeControl(functions=treebagFuncs, method = "cv",
                               number = 5, returnResamp = "final", verbose

```

```

= TRUE)
rf.rfe.treebag <- rfe(Churn~., data=datos,sizes=subsets,
                      rfeControl=ctrl.rfe.treebag)

rf.rfe.treebag
rf.rfe.treebag$fit
rf.rfe.treebag$optsize
rf.rfe.treebag$optVariables
#Y aquí podemos ver el nombre cada una de las varibales escogidas
[1] "TotalCharges"                      "MonthlyCharges"

[3] "tenure.L"                          "PaymentMethodElectronic check"

[5] "InternetServiceFiber optic"        "ContractTwo year"

[7] "PaperlessBillingYes"               "genderMale"

[9] "PartnerYes"                        "InternetServiceNo"

[11] "tenure.Q"                          "SeniorCitizenTRUE"

[13] "tenure.C"                          "OnlineSecurityYes"

[15] "DependentsYes"                     "TechSupportYes"

[17] "OnlineBackupYes"                   "MultipleLinesYes"

[19] "DeviceProtectionYes"               "StreamingMoviesYes"

[21] "StreamingTVYes"                     "PaymentMethodMailed check"

[23] "PaymentMethodCredit card (automatic)" "ContractOne year"

[25] "tenure^4"                          "PhoneServiceYes"

###SvmLinear
set.seed(3456)
ctrl.rfe.svmLinear <- rfeControl(functions=caretFuncs, method = "cv",
                                number = 5, returnResamp = "final", verbose
= TRUE)
rf.rfe.svmLinear <- rfe(Churn~., data=datos,sizes=subsets,
                      rfeControl=ctrl.rfe.svmLinear, method =
"svmLinear")
rf.rfe.svmLinear
rf.rfe.svmLinear$fit
rf.rfe.svmLinear$optsize
rf.rfe.svmLinear$optVariables
#También nos indica que el hiperparámetro seleccionado para C es 1.
rf.rfe.svmLinear$optsize
rf.rfe.svmLinear$optVariables
#Y aquí podemos ver el nombre cada una de las varibales escogidas
[1] "tenure.L"                          "InternetServiceFiber optic"

[3] "PaymentMethodElectronic check"      "TotalCharges"

[5] "ContractTwo year"                   "MonthlyCharges"

[7] "PaperlessBillingYes"                "InternetServiceNo"

[9] "OnlineSecurityYes"                  "PartnerYes"

[11] "TechSupportYes"                     "DependentsYes"

```

[13] "ContractOne year"	"tenure.C"
[15] "SeniorCitizenTRUE" (automatic)"	"PaymentMethodCredit card"
[17] "OnlineBackupYes"	"PaymentMethodMailed check"
[19] "DeviceProtectionYes"	"StreamingTVYes"
[21] "StreamingMoviesYes"	"MultipleLinesYes"
[23] "tenure.Q"	"tenure^4"
[25] "genderMale"	"PhoneServiceYes"

Podemos notar que en ambos casos selecciona 26 variables que son más que los atributos que en realidad hay (20 si contamos la variable de salida *Churn*), esto es debido a los distintos valores que pueden tomar dichas variables y su repetición, no sólo su aparición, indica precisamente cuán de importante es respecto a las otras. Ahora hacemos la intersección de las variables seleccionadas de los dos modelos y nos quedamos con aquellos atributos que aparezcan en ambos (sin contar las repeticiones).

```
#Seleccionamos la intersección de ambos modelos
intersect(rf.rfe.treebag$optVariables, rf.rfe.svmLinear$optVariables)
colnames(datos)
atributos.seleccionados <-
c('gender', 'SeniorCitizen', 'Partner', 'Dependents', 'tenure', 'PhoneService',
  'MultipleLines', 'InternetService',
  'OnlineSecurity', 'OnlineBackup',
  'DeviceProtection', 'TechSupport', 'Contract', 'PaperlessBilling',
  'StreamingTV', 'StreamingMovies', 'PaymentMethod', 'MonthlyCharges', 'TotalCharges')
atributos.seleccionados
datos.sel <- datos[, atributos.seleccionados] #Añadimos los atributos
seleccionados al nuevo dataframe
datos.sel <- data.frame(datos.sel, datos$Churn) #Añadimos la variable de
clasificación (Si se deserta o no)
colnames(datos.sel)[dim(datos.sel)[2]] <- "Churn" #Cambiamos el nombre de
la columna
datos.sel
```

Sorprendentemente para ambos modelos todas las variables, en mayor o menor medida, son significativas y como hemos detectado antes que la amenaza es real y es grave para la pérdida de clientes para la compañía, las vamos a tomar todas en cuenta según nos indican los modelos de selección. Por lo tanto, nos creamos un nuevo dataframe con las variables seleccionadas más la variable de clasificación *Churn* con el que vamos a realizar el entrenamiento y evaluación de nuestro modelo predictivo.

5. Entrenamiento del modelo y evaluación de los resultados.

En primer lugar, particionamos los datos con un 80% para los datos de entrenamiento y un 20% para los datos de test ya que consideramos que estas cantidades son idóneas para evitar tanto un underfitting como un overfitting, a la hora de entrenar el modelo.

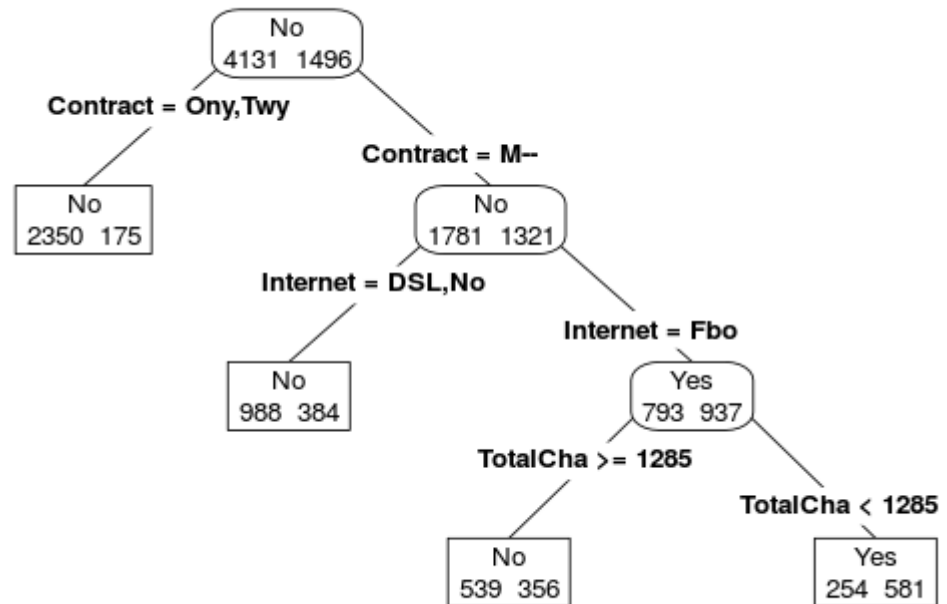
```
#Particionamos los conjuntos de entrenamiento y de test
trainIndex <- createDataPartition(datos$Churn,
                                   p = 0.8, list = FALSE, times = 1)
datosTrain <- datos.sel[trainIndex,]
datosTrain
datosTest <- datos.sel[-trainIndex,]
datosTest
```

A continuación entrenamos un modelo de Árbol de Decisión que es el que, tras deliberar los tres, hemos considerado como más adecuado para este tipo de problema.

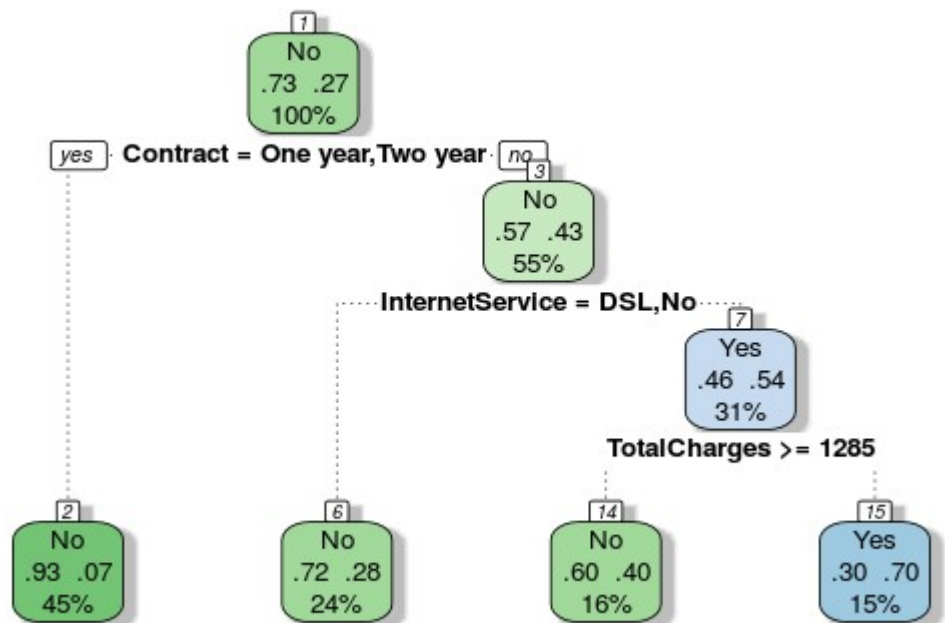
```
library(rpart)
library(rpart.plot)
library(RColorBrewer)
library(rattle)
modelo <- rpart(Churn ~ ., data = datosTrain, method = "class")
prp(modelo)
fancyRpartPlot(modelo)
prp(modelo, type=4, extra=1, clip.right.labs=FALSE, leaf.round=0)
# Variables más importantes
print(modelo$variable.importance[1:5])
      Contract      tenure  TotalCharges MonthlyCharges
PaymentMethod
    353.9083    244.0591    218.7130    125.5160    123.5429
```

Tras el entrenamieto del modelo obtenemos varias cosas:

1. El Árbol de Decisión



2. El Árbol de Decisión con sus probabilidades bayesianas



Rattle 2017-jun-10 20:47:51 moises

3. Las cinco variables más importantes:

- Contract
- tenure

- TotalCharges
- MonthlyCharges
- PaymentMethod

Pero aún nos queda una última cosa comprobar la bondad del modelo, mediante la predicción del conjunto de test y su posterior matriz de confusión.

```
predicciones <- predict(modelo, datosTest, type='class')
predicciones
matrizConfusion <- confusionMatrix(predicciones, datosTest$Churn)
matrizConfusion
Confusion Matrix and Statistics
```

	Reference	
Prediction	No	Yes
No	969	250
Yes	63	123

```

Accuracy : 0.7772
95% CI : (0.7545, 0.7987)
No Information Rate : 0.7345
P-Value [Acc > NIR] : 0.0001278

Kappa : 0.3199
McNemar's Test P-Value : < 2.2e-16

Sensitivity : 0.9390
Specificity : 0.3298
Pos Pred Value : 0.7949
Neg Pred Value : 0.6613
Prevalence : 0.7345
Detection Rate : 0.6897
Detection Prevalence : 0.8676
Balanced Accuracy : 0.6344

'Positive' Class : No
```

Obtenemos una precisión del 77.72% con un intervalo de confianza del 95% entre el rango (75.45% y 79.87%), y un índice Kappa del 31.99%. Lo cual nos da bastante confianza.

6. Conclusiones.

A la hora de evaluar un problema, en este caso cuando supone una amenaza, siempre es bueno evaluar el nivel de gravedad de dicha amenaza y asegurarse con dos o más modelos de selección antes de descartar ninguna variable, ya que en mayor o menor medida todas pueden ser significativas como ha sido el caso.

A la hora de entrenar un modelo tenemos que contar con toda la información que sea necesaria para evitar errores que puedan resultar, en este caso, pérdidas de clientes, tal vez, irrecuperables para la compañía y por tanto pérdida de facturación.

Teniendo ya en cuenta este modelo de Árbol de Decisión podemos proceder al siguiente apartado para segmentar los grupos tanto de los posibles clientes que deserten como de los que no, para ofrecerles campañas adecuadas a sus perfiles para retenerlos y no se nos vayan de la compañía.

B. Segmentación/creación de grupos entre los desertores y los no desertores

En este apartado se nos pide discernir entre los usuarios que abandonan la compañía para agruparlos en distintos perfiles. Estos perfiles serán clave para posteriormente crear las campañas de incentivos.

Para llevar a cabo dicha finalidad nos hemos decidido por, partiendo del conjunto de datos original, dividirlo entre los desertores y los no desertores (variable Churn de nuestro dataset). Una vez hecho esto, procedemos a extraer los distintos perfiles dentro de cada uno de los grupos mediante una técnica de clustering jerárquico (de esta forma, podremos visualizar las particiones que se hacen de los h)

Comenzaremos para ello cogiendo los registros de los usuarios que han decidido dejar la compañía y procedemos a...

C. Diseño de la campaña de incentivos

II. Proyecto 3: Netflix a todas horas

A. Recomendador basado en contenido

B. Recomendador basado en factorización matricial