Regresión Logística. Detección de churn. Caso Telecomunicaciones

Adolfo Sánchez Burón

Introducción al análisis de regresión logística

El Análisis de regresión logística es el indicado cuando queremos predecir una variable categórica binaria, en función de un conjunto de predictores. Cuando solo tenemos un predictor se le denomina RL simple, y cuando hay más de dos, RL múltiple.

La lógica de este estadístico es la siguiente: en función de una serie de predictores queremos predecir la probabilidad de que un caso pertenezca a una de las dos posibilidades de la variable binaria.

Si bien, la RL aporta una determinada probabilidad de ocurrencia para cada caso de pertenecer a una de las dos categorías, posteriormente se tendrá que establecer un límite (threshold) para determinar la pertenencia o no a una de esas dos categorías, el cual podrá ser más o menos restrictivo dependiendo de las características del estudio.

Por tanto, la función logística producirá una curva en S, indicando los valores 0 y 1, lo que se diferencia del análisis de regresión lineal, cuya variable dependiente es una variable cuantitativa continua, y reporta una linea recta.

Características del caso

El caso empleado en este análisis es el 'Telco Customer Churn', que puede descargarse el dataset original de Kaggle (https://www.kaggle.com/blastchar/telco-customer-churn). Este dataset ha sido previamente trabajado en cuanto a:

- análisis descriptivo
- limpieza de anomalías, missing y outliers
- peso predictivo de las variables mediante random forest
- discretización de las variables continuas para facilitar la interpretación posterior

Por lo que finalmente se emplea en este caso un dataset prepardo para iniciar el análisis de RL, que puede descargarse de Github.

El objetivo del caso es predecir la probabilidad de que un determinado cliente puede abandonar (churn) la empresa. La explicación de esta conducta estará basada en toda una serie de variables predictoras que se pueden clasificar en cuatro grupos:

- Churn: la variable TARGET, con puntuaciones de 0 (no abandonó la empresa) y 1 (sí abandonó la empresa)
- Servicios contratados: phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies.
- Información sobre cuentas dle cliente: how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges.
- Variables demográficas: gender, age range, and if they have partners and dependents

Tras estudiar el peso predictivo de estas variables sobre la TARGET, finalmente se redujo el número de predictores a 6: Internet Service, Contract, Payment Method, tenure, Monthly Charges y Total Charges.

Proceso

1. Entorno

El primer punto tratará sobre la preparación del entorno, donde se mostrará la descarga de las librerías empleadas y la importación de datos.

2. Análisis descriptivo

Se mostrarán y explicarán las funciones empleadas en este paso, dividiéndolas en tres grupos: Análisis inicial, Tipología de datos y Análisis descriptivo (gráficos).

3. Modelización

Se preparará lo necesario para modelizar con el análisis de RL, como es:

a. Preparar funciones:

- Matriz de confusión
- Métricas
- Umbrales
- Curva ROC y AUC

b. Particiones del dataset en dos grupos: training (70%) y test (30%)

4. Modelización con regresión logística

Por motivos didácticos, se dividirá en seis pasos:

- Paso 1. Primer modelo
- Paso 2. Segundo modelo
- Paso 3. Predict
- Paso 4. Umbrales
- Paso 5. Matriz de confusión
- Paso 6. Métricas definitivas

---

1. Entorno

1.1. Instalar librerías

```
library(data.table) #para leer y escribir datos de forma rapida
library(dplyr) #para manipulación de datos
library(tidyr) #para manipulación de datos
library(ggplot2) #para gráficos
library(ROCR) #para evaluar modelos
library(DataExplorer) #para realizar el análisis descriptivo con gráficos
```

1.2. Importar datos

Como el dataset ha sido peviamente trabajado para poder modelizar directamente, si deseas seguir este tutorial, lo puedes descargar de GitHub (https://github.com/AdSan-R).

```
# Importamos los datos y los incluimos en un data frame llamado df1
df1 <- fread("TelcoChurn.csv")
```

```
options(scipen=999) #Desactivar la notación científica
```

---

2. Análisis descriptivo

2.1. Análisis inicial

```
head(df1) #con esta función podemos ver la estructura de los primeros 6 casos
```

```
##    InternetService       Contract             PaymentMethod tenure_DISC
## 1:             DSL Month-to-month          Electronic check     Grupo 1
## 2:             DSL       One year              Mailed check     Grupo 2
## 3:             DSL Month-to-month              Mailed check     Grupo 1
## 4:             DSL       One year Bank transfer (automatic)     Grupo 3
## 5:     Fiber optic Month-to-month          Electronic check     Grupo 1
## 6:     Fiber optic Month-to-month          Electronic check     Grupo 1
##    MonthlyCharges_DISC TotalCharges_DISC TARGET
## 1:             Grupo 1           Grupo 1     No
## 2:             Grupo 2           Grupo 3     No
## 3:             Grupo 2           Grupo 1     Si
## 4:             Grupo 2           Grupo 3     No
## 5:             Grupo 3           Grupo 1     Si
## 6:             Grupo 4           Grupo 2     Si
```

```
str(df1) #mostrar la estructura del dataset y los tipos de variables
```

```
## Classes 'data.table' and 'data.frame':   7032 obs. of  7 variables:
##  $ InternetService    : chr  "DSL" "DSL" "DSL" "DSL" ...
##  $ Contract           : chr  "Month-to-month" "One year" "Month-to-month" "One yea
r" ...
##  $ PaymentMethod      : chr  "Electronic check" "Mailed check" "Mailed check" "Bank
transfer (automatic)" ...
##  $ tenure_DISC        : chr  "Grupo 1" "Grupo 2" "Grupo 1" "Grupo 3" ...
##  $ MonthlyCharges_DISC: chr  "Grupo 1" "Grupo 2" "Grupo 2" "Grupo 2" ...
##  $ TotalCharges_DISC  : chr  "Grupo 1" "Grupo 3" "Grupo 1" "Grupo 3" ...
##  $ TARGET             : chr  "No" "No" "Si" "No" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Puede observarse que todas son “chr”, esto es, “character”, por tanto, vamos a pasarlas a Factor.

2.2. Tipología de datos

```
df1 <- mutate_if(df1, is.character, as.factor) #identifica todas las variables charact
er y transformarlas en factores

str(df1) #estructura de la base de datos después de la transformación
```

```
## Classes 'data.table' and 'data.frame':   7032 obs. of  7 variables:
##  $ InternetService    : Factor w/ 3 levels "DSL","Fiber optic",..: 1 1 1 1 2 2 2 1
2 1 ...
##  $ Contract           : Factor w/ 3 levels "Month-to-month",..: 1 2 1 2 1 1 1 1 1 2
...
##  $ PaymentMethod      : Factor w/ 4 levels "Bank transfer (automatic)",..: 3 4 4 1
3 3 2 4 3 1 ...
##  $ tenure_DISC        : Factor w/ 4 levels "Grupo 1","Grupo 2",..: 1 2 1 3 1 1 2 1
2 4 ...
##  $ MonthlyCharges_DISC: Factor w/ 4 levels "Grupo 1","Grupo 2",..: 1 2 2 2 3 4 3 1
4 2 ...
##  $ TotalCharges_DISC  : Factor w/ 4 levels "Grupo 1","Grupo 2",..: 1 3 1 3 1 2 3 1
3 3 ...
##  $ TARGET             : Factor w/ 2 levels "No","Si": 1 1 2 1 2 2 1 1 2 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

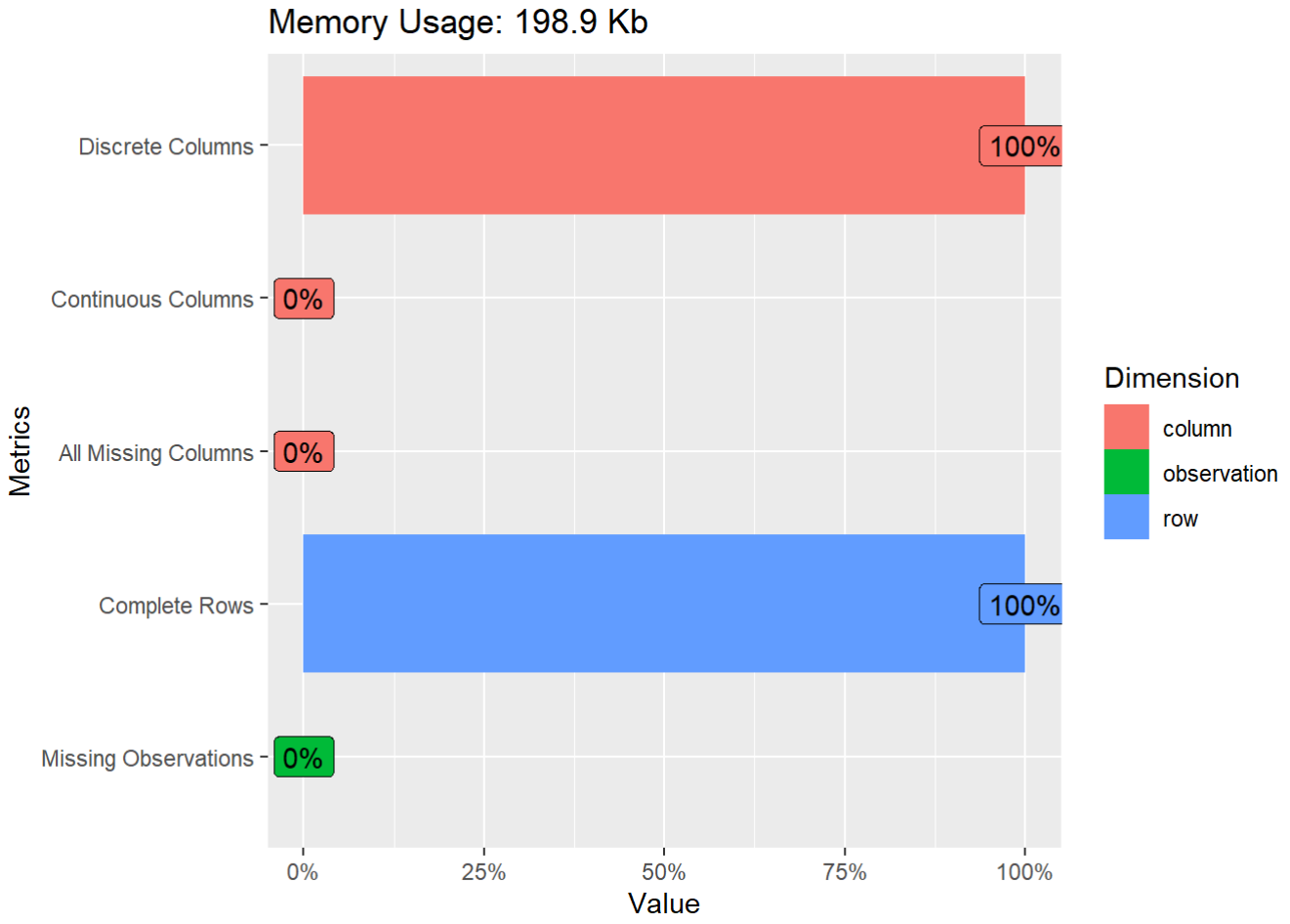
Ahora se puede observar que todas las variables son de tipo “Factor”

```
lapply(df1,summary) #mostrar la distribución de frecuencias en cada categoría de todas
las variables
```

```
## $InternetService
##         DSL Fiber optic          No
##        2416        3096        1520
##
## $Contract
## Month-to-month       One year       Two year
##           3875           1472           1685
##
## $PaymentMethod
## Bank transfer (automatic)   Credit card (automatic)         Electronic check
##                      1542                      1521                      2365
##              Mailed check
##                      1604
##
## $tenure_DISC
## Grupo 1 Grupo 2 Grupo 3 Grupo 4
##    2723    1308    1182    1819
##
## $MonthlyCharges_DISC
## Grupo 1 Grupo 2 Grupo 3 Grupo 4
##    1758    1761    1755    1758
##
## $TotalCharges_DISC
## Grupo 1 Grupo 2 Grupo 3 Grupo 4
##    1758    1758    1758    1758
##
## $TARGET
##   No   Si
## 5163 1869
```

2.3. Análisis descriptivo (gráficos)

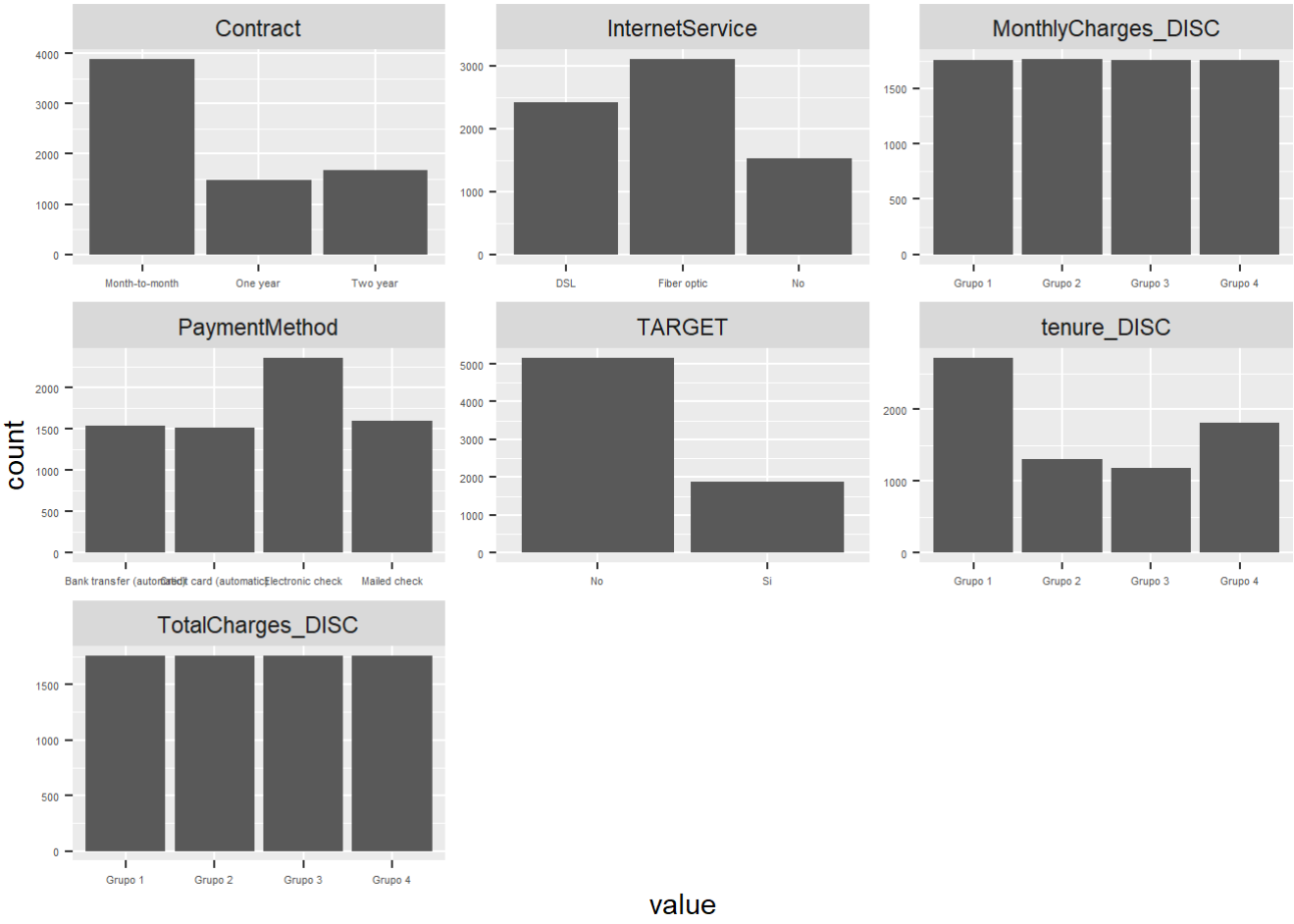
```
plot_intro(df1) #gráfico para observar la distribución de variables y los casos missin
g por columnas, observaciones y filas
```



Como se ha trabajado previamente, no existen casos missing, por lo que podemos seguir el análisis descriptivo

```
#Análisis visual de frecuencias de cada categoría por variable
df1 %>%
  gather() %>%
  ggplot(aes(value)) +
  geom_bar()+
  facet_wrap(~ key, scales = "free")+
  theme(axis.text=element_text(size=4))
```

```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```



En los gráficos anteriores pueden observarse las categorías de cada variable, algunas de ellas dicotomizadas previamente, por lo que haremos un repaso de cada una:

- Internet Service: tiene tres niveles: DSL, Fiber optic, No.
- Contract (tipo de contrato): tiene tres niveles, Month-to-month, One year, Two years.
- Payment Method: con tres niveles, Bank transfer, Credit card, Electronic check.
- Tenure: variable originalmente cuantitativa, que se discretizó en cuatro categorías por cuartiles.
- Monthly Charges: se discretizó en cuatro categorías. Grupo 1 (<= 35.59), Grupo 2 (> 35.59 & <= 70.35), Grupo 3 (> 70.35 & <= 89.86), Grupo 4 (> 89.86).
- Total Charges: se discretizó en cuatro categorías. Grupo 1 (<= 401.4), Grupo 2 (> 401.4 & <= 1397.5), Grupo 3 (> 1397.5 & <= 3794.7), Grupo 4 (> 3794.7).
- TARGET: con dos niveles Sí han abandonado (churn), No han abandonado (churn).

Parece que la distribución de frecuencias en todas las variables es aceptable, incluso en la variable TARGET, que suele dar más problemas.

---

3. Modelización

3.1. Preparar funciones

Tomadas del curso de Machine Learning Predictivo (https://www.datascience4business.com/08_mlc-salespage-b) de DS4B) :

- Matriz de confusión
- Métricas
- Umbrales
- Curva ROC y AUC

Función para la matriz de confusión

En esta función se prepara la matriz de confusión (ver en otro post), donde se observa qué casos coinciden entre la puntuación real (obtenida por cada sujeto) y la puntuación predicha ("scoring") por el modelo, estableciendo previmente un límite ("umbral") para ello.

```
confusion<-function(real,scoring,umbral){
  conf<-table(real,scoring>=umbral)
  if(ncol(conf)==2) return(conf) else return(NULL)
}
```

Funcion para métricas de los modelos

Los indicadores a observar serán:

- Acierto (accuracy) = (TRUE POSITIVE + TRUE NEGATIVE) / TODA LA POBLACIÓN
- Precisión = TRUE POSITIVE / (TRUE POSITIVE + FALSE POSITIVE)
- Cobertura (recall, sensitivity) = TRUE POSITIVE / (TRUE POSITIVE + FALSE NEGATIVE)
- F1 = 2* (precisión * cobertura) (precisión + cobertura)

```
metricas<-function(matriz_conf){
  acierto <- (matriz_conf[1,1] + matriz_conf[2,2]) / sum(matriz_conf) *100
  precision <- matriz_conf[2,2] / (matriz_conf[2,2] + matriz_conf[1,2]) *100
  cobertura <- matriz_conf[2,2] / (matriz_conf[2,2] + matriz_conf[2,1]) *100
  F1 <- 2*precision*cobertura/(precision+cobertura)
  salida<-c(acierto,precision,cobertura,F1)
  return(salida)
}
```

Función para probar distintos umbrales

Con esta función se analiza el efecto que tienen distintos umbrales sobre los indicadores de la matriz de confusión (precisión y cobertura). Lo que buscaremnos será aquél que maximice la relación entre cobertura y precisión (F1).

```
umbrales<-function(real,scoring){
  umbrales<-data.frame(umbral=rep(0,times=19),acierto=rep(0,times=19),precision=rep(0,
times=19),cobertura=rep(0,times=19),F1=rep(0,times=19))
  cont <- 1
  for (cada in seq(0.05,0.95,by = 0.05)){
    datos<-metricas(confusion(real,scoring,cada))
    registro<-c(cada,datos)
    umbrales[cont,]<-registro
    cont <- cont + 1
  }
  return(umbrales)
}
```

Funciones para calcular la curva ROC y el AUC

Por último, se preapra una función para calcular la curva ROC y el AUC.

- Curva ROC (Relative Operating Characteristic): resentación gráfica de la relación entre la cobertura (proporción de verdaderos positivos) y la especificidad (razón de falsos positivos). Muestra el rendimiento del modelo en todos los umbrales de clasificación.
- AUC (Area Under The Curve): mide el área que queda debajo de la curva. Indica en qué medida el modelo será capaz de clasificar adecuadamente. La AUC tiene un rango entre 0 y 1. Si es igual o cercano a 0.5, no tiene capacidad discriminativa.

```
roc<-function(prediction){
  r<-performance(prediction,'tpr','fpr')
  plot(r)
}

auc<-function(prediction){
  a<-performance(prediction,'auc')
  return(a@y.values[[1]])
}
```

3.2. Particiones de training (70%) y test (30%)

Se divide la muestra en dos partes:

1. Training o entrenamiento (70% de la muestra): servirá para entrenar al modelo de clasificación.
2. Test (30%): servirá para validar el modelo. La característica fundamental es que esta muestra no debe haber tenido contacto previamente con el funcionamiento del modelo.

```
# Lanzamos una semilla para que salgan siempre los mismos datos
set.seed(12345)

# Creamos los dataframes

# Generamos una variable aleatoria con una distribución 70-30
df1$random<-sample(0:1,size = nrow(df1),replace = T,prob = c(0.3,0.7))

train<-filter(df1,random==1)
test<-filter(df1,random==0)
#Eliminamos ya la random para que no moleste
df1$random <- NULL
```

---

4. Modelización con regresión logística

Paso 1. Primer modelo

Primero vamos a hacer un modelo con todas las variables seleccionadas y lo incluimos en un objeto llamado “rl”

```
rl<- glm(TARGET ~ InternetService + Contract + PaymentMethod + tenure_DISC + MonthlyCharges_DISC + TotalCharges_DISC, train, family=binomial(link='logit'))
# glm: Lanzamos la función glm para entre un modelo de RL, de la familia "binomial, "logit".
# TARGET ~ InternetService + Contract + PaymentMethod + tenure_DISC + MonthlyCharges_DISC + TotalCharges_DISC : es el modelo a entrenar, con VD la TARGET, y el resto son los predictores.
# train: solo lo lanzamos con el df "train", para entrenar el modelo.

# summary(rl): función para ver los resultados del primer modelo
summary(rl)
```

```
##
## Call:
## glm(formula = TARGET ~ InternetService + Contract + PaymentMethod +
##     tenure_DISC + MonthlyCharges_DISC + TotalCharges_DISC, family = binomial(link =
"logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9207  -0.7133  -0.3302   0.6864   3.1109
##
## Coefficients:
##                                     Estimate Std. Error z value
## (Intercept)                         -0.11606    0.19318  -0.601
## InternetServiceFiber optic           0.92305    0.15812   5.838
## InternetServiceNo                   -1.27963    0.20948  -6.109
## ContractOne year                    -0.79367    0.12188  -6.512
## ContractTwo year                    -1.88411    0.20780  -9.067
## PaymentMethodCredit card (automatic) -0.11222    0.13196  -0.850
## PaymentMethodElectronic check        0.38233    0.10924   3.500
## PaymentMethodMailed check           -0.21997    0.13222  -1.664
## tenure_DISCGrupo 2                  -0.47277    0.17012  -2.779
## tenure_DISCGrupo 3                  -0.38953    0.22206  -1.754
## tenure_DISCGrupo 4                  -0.80592    0.26953  -2.990
## MonthlyCharges_DISCGrupo 2          -0.04093    0.18620  -0.220
## MonthlyCharges_DISCGrupo 3           0.13620    0.23974   0.568
## MonthlyCharges_DISCGrupo 4           0.48312    0.26604   1.816
## TotalCharges_DISCGrupo 2            -0.85822    0.11929  -7.195
## TotalCharges_DISCGrupo 3            -1.07485    0.21233  -5.062
## TotalCharges_DISCGrupo 4            -1.28545    0.30393  -4.229
##                                             Pr(>|z|)
## (Intercept)                                 0.547983
## InternetServiceFiber optic             0.000000005294636 ***
## InternetServiceNo                      0.000000001005495 ***
## ContractOne year                       0.000000000074195 ***
## ContractTwo year                      < 0.0000000000000002 ***
## PaymentMethodCredit card (automatic)        0.395094
## PaymentMethodElectronic check               0.000466 ***
## PaymentMethodMailed check                   0.096183 .
## tenure_DISCGrupo 2                          0.005452 **
## tenure_DISCGrupo 3                          0.079398 .
## tenure_DISCGrupo 4                          0.002789 **
## MonthlyCharges_DISCGrupo 2                  0.826008
## MonthlyCharges_DISCGrupo 3                  0.569962
## MonthlyCharges_DISCGrupo 4                  0.069379 .
## TotalCharges_DISCGrupo 2               0.000000000000627 ***
## TotalCharges_DISCGrupo 3               0.000000414738210 ***
## TotalCharges_DISCGrupo 4               0.000023433479202 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5796.6  on 4933  degrees of freedom
## Residual deviance: 4246.8  on 4917  degrees of freedom
## AIC: 4280.8
##
## Number of Fisher Scoring iterations: 6
```

Revisamos la significatividad DE RL y mantenemos todas las variables que tengan tres estrellas en alguna categoría, Entran todas menos menos MonthlyCharges. Por tanto, lanzamos un segundo modelo con RL eliminando del modelo a MonthlyCharges.

Paso 2. Segundo modelo

```
rl2<- glm(TARGET ~ InternetService + Contract + PaymentMethod + tenure_DISC + TotalCha
rges_DISC, train, family=binomial(link='logit'))

summary(rl2)
```

```
##
## Call:
## glm(formula = TARGET ~ InternetService + Contract + PaymentMethod +
##     tenure_DISC + TotalCharges_DISC, family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7925  -0.7066  -0.3371   0.6692   3.1010
##
## Coefficients:
##                                    Estimate Std. Error z value
## (Intercept)                        -0.17549    0.12619  -1.391
## InternetServiceFiber optic          1.15839    0.09916  11.683
## InternetServiceNo                  -1.22651    0.14865  -8.251
## ContractOne year                   -0.76548    0.12110  -6.321
## ContractTwo year                   -1.84412    0.20665  -8.924
## PaymentMethodCredit card (automatic) -0.10060    0.13166  -0.764
## PaymentMethodElectronic check       0.39969    0.10897   3.668
## PaymentMethodMailed check          -0.22048    0.13204  -1.670
## tenure_DISCGrupo 2                 -0.51795    0.16855  -3.073
## tenure_DISCGrupo 3                 -0.50104    0.21804  -2.298
## tenure_DISCGrupo 4                 -0.93193    0.26513  -3.515
## TotalCharges_DISCGrupo 2           -0.81542    0.11631  -7.011
## TotalCharges_DISCGrupo 3           -0.92292    0.20149  -4.581
## TotalCharges_DISCGrupo 4           -0.94535    0.28054  -3.370
##                                             Pr(>|z|)
## (Intercept)                                 0.164325
## InternetServiceFiber optic          < 0.0000000000000002 ***
## InternetServiceNo                   < 0.0000000000000002 ***
## ContractOne year                        0.00000000026010 ***
## ContractTwo year                    < 0.0000000000000002 ***
## PaymentMethodCredit card (automatic)            0.444802
## PaymentMethodElectronic check                   0.000245 ***
## PaymentMethodMailed check                       0.094971 .
## tenure_DISCGrupo 2                              0.002119 **
## tenure_DISCGrupo 3                              0.021562 *
## tenure_DISCGrupo 4                              0.000440 ***
## TotalCharges_DISCGrupo 2                0.00000000000237 ***
## TotalCharges_DISCGrupo 3                0.00000463771260 ***
## TotalCharges_DISCGrupo 4                        0.000752 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5796.6  on 4933  degrees of freedom
## Residual deviance: 4258.0  on 4920  degrees of freedom
## AIC: 4286
##
## Number of Fisher Scoring iterations: 6
```

Vemos que ahora ya todas las variables tienen al menos una categoría con 3 estrellas de significación

Calculamos el pseudo R cuadrado de McFadden (“residual deviance” / “null deviance”): los resultados entre 0,2 y 0,4 indican un excelente ajuste del modelo.

```
pr2_rl <- 1 -(rl2$deviance / rl2$null.deviance)
pr2_rl
```

```
## [1] 0.2654273
```

Paso 3. Predict

Aplicamos el modelo entrenado al conjunto de test (30%), generando un vector con las probabilidades en cada caso de ser 0 o 1.

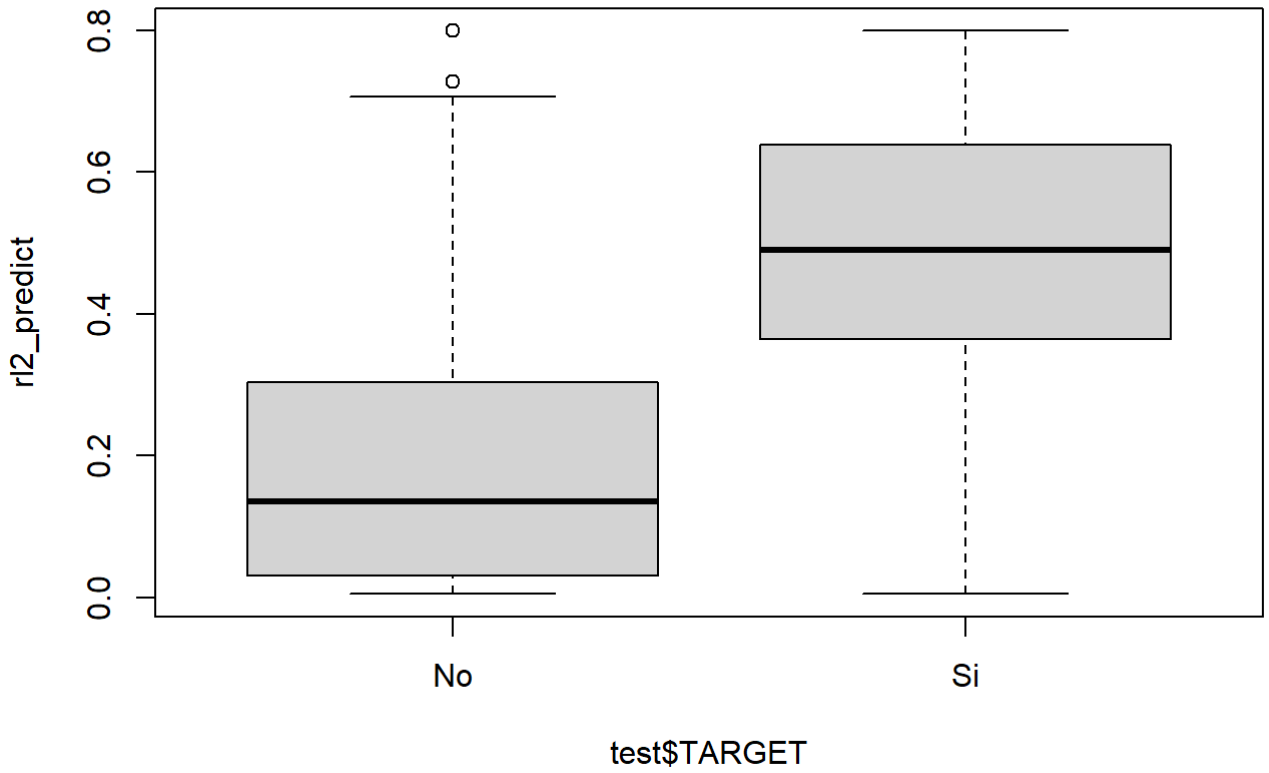
```
rl2_predict<-predict(rl2,test,type = 'response')
#type= 'response'. obtener de cada caso la probabilidqad de churn, lo que me permitirá posteriormente trabajar con umbrales
head(rl2_predict)
```

```
##          1          2          3          4          5          6
## 0.55581715 0.06898971 0.40228287 0.08588590 0.48543365 0.05755050
```

Lanzamos un “head” para ver los 6 primeros. Lo que quiere decir que: el sujeto 1 tendrá una probabilidad de clasificarse como 1 (Sí Churn) del 55,58%. El segundo de 6,90%, etc.

A continuación lanzamos un plot de caja y bigotes, para ver si discrimina bien entre las dos categorías, esto es, si la media de rl_predict de los clientes que sí contratan con la media de los que no contratan es diferente.

```
plot(rl2_predict~test$TARGET)
```



Se observa en la gráfica que la media de los que Sí y los que No es muy diferente, incluso discrimina bien entre los cuartiles.

Paso 4. Umbrales

Ahora tenemos que transformar la probabilidad obtenida en una decisión de si el cliente va a abandonar o no.

Con la función umbrales probamos diferentes cortes

```
umb_rl2<-umbrales(test$TARGET,rl2_predict)
umb_rl2
```

```
##    umbral  acierto precision cobertura       F1
## 1    0.05 46.18684  31.13846  98.06202 47.26763
## 2    0.10 58.86559  36.96469  95.34884 53.27558
## 3    0.15 62.77407  39.28860  94.18605 55.44780
## 4    0.20 69.01811  43.50775  87.01550 58.01034
## 5    0.25 73.30791  47.51131  81.39535 60.00000
## 6    0.30 75.50048  50.12255  79.26357 61.41141
## 7    0.35 77.07340  52.28162  77.71318 62.50974
## 8    0.40 79.55195  57.09625  67.82946 62.00177
## 9    0.45 80.31459  59.51941  62.40310 60.92715
## 10   0.50 80.79123  64.16040  49.61240 55.95628
## 11   0.55 80.64824  65.44944  45.15504 53.44037
## 12   0.60 80.07626  67.01389  37.40310 48.00995
## 13   0.65 78.59867  71.06918  21.89922 33.48148
## 14   0.70 78.36034  73.13433  18.99225 30.15385
## 15   0.75 77.64538  71.55963  15.11628 24.96000
## 16   0.80  0.80000   0.80000   0.80000  0.80000
## 17   0.85  0.85000   0.85000   0.85000  0.85000
## 18   0.90  0.90000   0.90000   0.90000  0.90000
## 19   0.95  0.95000   0.95000   0.95000  0.95000
```

Seleccionamos el umbral que maximiza la F1 (cuando empieza a decaer)

```
umbral_final_rl2<-umb_rl2[which.max(umb_rl2$F1),1]
umbral_final_rl2
```

```
## [1] 0.35
```

Como puede observarse en la tabla anterior, el indicador F1 crece a medida que los umbrales aumentan (esto es, se maximiza progresivamente la F1), pero llega a un punto que empieza a decrecer: umbral de 0.35

Paso 5. Matriz de confusión

Evaluamos la matriz de confusión y las métricas con el umbral optimizado

```
confusion(test$TARGET,rl2_predict,umbral_final_rl2)
```

```
##
## real FALSE TRUE
##   No  1216  366
##   Si   115  401
```

```
rl2_metricas<-filter(umb_rl2,umbral==umbral_final_rl2)
rl2_metricas
```

```
##   umbral acierto precision cobertura       F1
## 1   0.35 77.0734  52.28162  77.71318 62.50974
```

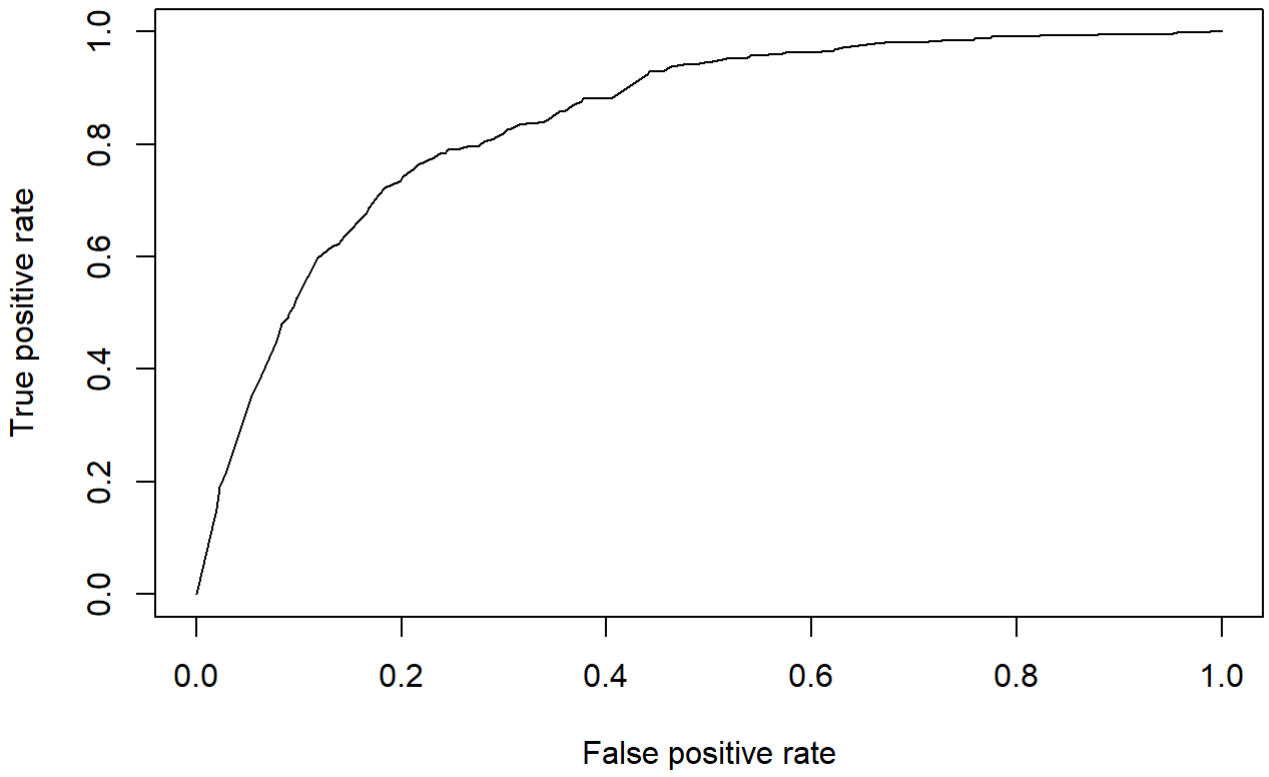
Observamos que para el umbral 0.35, tenemos un modelo con las métricas:

- acierto = 77.0734
- precision = 52.28162
- cobertura = 77.71318
- F1 = 62.50974

Paso 6. Métricas definitivas

Evaluamos la ROC

```
#creamos el objeto prediction
rl2_prediction<-prediction(rl2_predict,test$TARGET)
#visualizamos la ROC
roc(rl2_prediction)
```



En la curva ROC, la línea diagonal que divide el gráfico en dos partes iguales indica que el modelo no tiene ninguna capacidad predictiva. Todo el área que está por encima de esa diagonal hasta la curva, indica la capacidad predictiva del modelo.

Métricas definitivas

```
rl2_metricas<-cbind(rl2_metricas,AUC=round(auc(rl2_prediction),2)*100)
print(t(rl2_metricas))
```

```
##               [,1]
## umbral     0.35000
## acierto   77.07340
## precision 52.28162
## cobertura 77.71318
## F1        62.50974
## AUC       84.00000
```

Obtenemos las métricas definitivas añadiendo la métrica AUC, que indica el porcentaje de predicción del modelo, un 84%, lo que indica que es un buen modelo.