Manuel Carmona Cabello de Alba
manuel.carmona@cunef.edu

# Practica2 - credit risk and loan performance

El objetivo de esta practica es establecer un modelo que permita predecir si un prestamo hara default o no a partir de los datos de Lending Club (2007-2011). Tenemos que depurar los datos y trabajar con ellos para estimar un modelo que permita responder a preguntas como, por ejemplo, las siguientes:

¿Cuáles son las características de los prestatarios que permiten determinar el riesgo de impago? ¿A partir de qué punto se debe denegar un crédito al cliente?

## Carga de datos

El primer paso es importar los datos desde el fichero. Después, tras un estudio exhaustivo de las variables a partir del archivo LCDataDictionary, filtraremos las variables en función de la información que contengan. Eliminaremos aquellas que tengan la informacion muy incompleta. Posteriormente eliminaremos las variables que, desde un punto de vista de negocio, no consideremos relevantes o que solapen la información de otra variable.

```
setwd("C:/Users/Manuel/Desktop/CUNEF/Prediccion/Practica2")
loandata<-
read.csv("C:/Users/Manuel/Desktop/CUNEF/Prediccion/Practica2/LoanStats3aS
EP.csv", sep="|", dec=".", fill=T, header=T)
#View(loandata)
```

Primer Filtro

```
loandata<-loandata[,c(2:47)]
loandata_orig<-loandata
```

Segundo Filtro

```
loandata<-loandata[,c(-18,-28,-29)]
```

Vamos a dejar las variables que podrian interesarnos para explicar loan_status (ademas de loan_status): el tipo de interés, el pago mensual (installment), el grado de calidad crediticia del credito, el subgrado de calidad, renta anual, detb-to-income ratio, los incididentes de morosidad en los ultimos 2 años,los incididentes de morosidad en los ultimos 6 meses, el numero de lineas de credito abiertas por el prestatario, balance de credito, credito utilizado respecto al total disponible, el numero total de lineas de credito actualmente asignadas al perfil de credito del prestatario y el pago total recibido hasta la fecha por el total financiado.

```
loandata2<-loandata[,c(6:9,13,16,23,24,26,27,29,30,31,35)]
#loandata2<-na.omit(loandata2)
str(loandata2)

## 'data.frame':    42585 obs. of  14 variables:
##  $ int_rate      : Factor w/ 396 levels "","  5.42%","  5.79%",..: 80
223 241 162 137 30 241 324 372 137 ...
##  $ installment   : Factor w/ 16447 levels "","100","100.04",..: 2239
13098 15550 7900 14072 2008 2518 371 1862 822 ...
##  $ grade         : Factor w/ 11 levels "","0","1","2",..: 6 7 7 7 6 5
7 9 10 6 ...
##  $ sub_grade     : Factor w/ 39 levels "","41","44","64",..: 11 18 19
15 14 8 19 25 31 14 ...
##  $ annual_inc    : Factor w/ 5592 levels "","10000","100000",..: 1233
1610 428 3049 4911 2029 2878 2960 2356 690 ...
##  $ loan_status   : Factor w/ 16 levels "","Aug-2010",..: 10 3 10 10 10
10 10 10 3 3 ...
##  $ dti           : Factor w/ 2916 levels ""," Approachable<br/>
Borrower added on 02/18/10 > Hi!  I am a segment and field producer for
Viacom. Thanks for y"| __truncated__,..: 2021 103 2772 1303 997 323 1654
2434 2454 1011 ...
##  $ delinq_2yrs   : Factor w/ 62 levels "","0","0.0","0.6",..: 2 2 2 2
2 2 2 2 2 2 ...
##  $ inq_last_6mths: Factor w/ 80 levels "","0","1","10",..: 3 32 17 3 2
24 3 17 17 2 ...
##  $ open_acc      : Factor w/ 49 levels "","0","1","10",..: 25 25 14 4
9 48 44 36 5 14 ...
##  $ revol_bal     : Factor w/ 22663 levels "","0","1","10",..: 3361
5839 12104 17579 11442 20543 6401 20838 16979 21945 ...
##  $ revol_util    : Factor w/ 1166 levels "","0","0%","0.01%",..: 988
1057 1150 211 632 301 1007 1026 363 410 ...
##  $ total_acc     : Factor w/ 130 levels "","1","10","11",..: 125 48 3
43 44 5 4 48 6 30 ...
##  $ total_pymnt   : Factor w/ 42237 levels "","0.0","0.00",..: 32169
335 22350 4866 26659 31344 324 26263 33922 9100 ...
```

## Proceso de limpieza de la variable a explicar: loan_status

Dado que tenemos valores erróneos en algunas observaciones, vamos a limpiar el dataset para poder trabajar de forma oportuna.

```
posiciones_vacio <- NULL
for (i in (1:length(loandata2$loan_status))){
        if (loandata2$loan_status[i]==""){
                posiciones_vacio <- c(posiciones_vacio,i)
        }

}

length(posiciones_vacio)
```

```
## [1] 119

loandata2<-loandata2[-posiciones_vacio,]
```

Utilizamos la funcion revalue para cambiar los valores erroneos por NA y eliminarlos facilmente

```
library(plyr)

## Warning: package 'plyr' was built under R version 3.4.2

loandata2$loan_status<-revalue(loandata2$loan_status, c("Does not meet
the credit policy. Status:Fully Paid"="Fully Paid", "Does not meet the
credit policy. Status:Charged Off"="Charged Off", "Aug-2010"="NA", "Jul-
2010"="NA", "Mar-2011"="NA", "May-2011"="NA", "Nov-2011"="NA", "Oct-
2010"="NA", "Sep-2011"="NA", "Dec-2010"="NA", "Dec-2011"="NA", "f"="NA",
"Feb-2011"="NA"))

loandata2$loan_status<-revalue(loandata2$loan_status,c("Feb-2011"="Fully
Paid", "NA"="Fully Paid"))

## The following `from` values were not present in `x`: Feb-2011
```

Comprobamos que todo es correcto

```
summary(loandata2$loan_status)

##              Fully Paid Charged Off
##           0       36044        6422
```

## Limpiamos el resto de variables importantes

### dti

```
posiciones_vacio <- NULL
posiciones_f <- NULL

for (i in (1:length(loandata2$dti))){
        if (loandata2$dti[i]==""){
                posiciones_vacio <- c(posiciones_vacio,i)
        }

}

head(posiciones_vacio)

## [1]  4312  5118  6255  8210 13326 15265

loandata2<-loandata2[-posiciones_vacio,]
```

## delinq_2yrs

*The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years*

```
posiciones_vacio <- NULL
posiciones_f <- NULL

for (i in (1:length(loandata2$delinq_2yrs))){
        if (loandata2$delinq_2yrs[i]==""){
                posiciones_vacio <- c(posiciones_vacio,i)
        }

}

head(posiciones_vacio)

## [1] 42366 42367 42376 42389 42397 42400

loandata2<-loandata2[-posiciones_vacio,]
```

## revol_bal

*Total credit revolving balance - Revolving credit is a type of credit that can be used repeatedly up to a certain limit as long as the account is open and payments are made on time.*

```
posiciones_vacio <- NULL
posiciones_f <- NULL

for (i in (1:length(loandata2$revol_bal))){
        if (loandata2$revol_bal[i]==""){
                posiciones_vacio <- c(posiciones_vacio,i)
        }

}

head(posiciones_vacio)

## [1] 3809

loandata2<-loandata2[-posiciones_vacio,]
```

## revol_util

*Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.*

```
posiciones_vacio <- NULL
posiciones_f <- NULL
```

```
for (i in (1:length(loandata2$revol_util))){
        if (loandata2$revol_util[i]==""){
                posiciones_vacio <- c(posiciones_vacio,i)
        }

}

head(posiciones_vacio)

## [1]  3573  4723  4953 11300 12058 12163

loandata2<-loandata2[-posiciones_vacio,]
```

¿De qué tipo son mis datos?

```
str(loandata2)

## 'data.frame':    42360 obs. of  14 variables:
##  $ int_rate      : Factor w/ 396 levels "","  5.42%","  5.79%",..: 80
223 241 162 137 30 241 324 372 137 ...
##  $ installment   : Factor w/ 16447 levels "","100","100.04",..: 2239
13098 15550 7900 14072 2008 2518 371 1862 822 ...
##  $ grade         : Factor w/ 11 levels "","0","1","2",..: 6 7 7 7 6 5
7 9 10 6 ...
##  $ sub_grade     : Factor w/ 39 levels "","41","44","64",..: 11 18 19
15 14 8 19 25 31 14 ...
##  $ annual_inc    : Factor w/ 5592 levels "","10000","100000",..: 1233
1610 428 3049 4911 2029 2878 2960 2356 690 ...
##  $ loan_status   : Factor w/ 3 levels "","Fully Paid",..: 2 3 2 2 2 2
2 2 3 3 ...
##  $ dti           : Factor w/ 2916 levels ""," Approachable<br/>
Borrower added on 02/18/10 > Hi!  I am a segment and field producer for
Viacom. Thanks for y"| __truncated__,..: 2021 103 2772 1303 997 323 1654
2434 2454 1011 ...
##  $ delinq_2yrs   : Factor w/ 62 levels "","0","0.0","0.6",..: 2 2 2 2
2 2 2 2 2 2 ...
##  $ inq_last_6mths: Factor w/ 80 levels "","0","1","10",..: 3 32 17 3 2
24 3 17 17 2 ...
##  $ open_acc      : Factor w/ 49 levels "","0","1","10",..: 25 25 14 4
9 48 44 36 5 14 ...
##  $ revol_bal     : Factor w/ 22663 levels "","0","1","10",..: 3361
5839 12104 17579 11442 20543 6401 20838 16979 21945 ...
##  $ revol_util    : Factor w/ 1166 levels "","0","0%","0.01%",..: 988
1057 1150 211 632 301 1007 1026 363 410 ...
##  $ total_acc     : Factor w/ 130 levels "","1","10","11",..: 125 48 3
43 44 5 4 48 6 30 ...
##  $ total_pymnt   : Factor w/ 42237 levels "","0.0","0.00",..: 32169
335 22350 4866 26659 31344 324 26263 33922 9100 ...
```

Transformamos las variables de tipo factor a numerico, si procede.

```r
typeof(loandata2$int_rate)

## [1] "integer"

loandata2$int_rate = gsub("%", "",loandata2$int_rate)
head(loandata2$int_rate)

## [1] " 10.65" " 15.27" " 15.96" " 13.49" " 12.69" "  7.90"

loandata2$revol_util= gsub("%", "",loandata2$revol_util)
head(loandata2$revol_util)

## [1] "83.7" "9.4"  "98.5" "21"   "53.9" "28.3"

loandata2$dti<-as.numeric(paste(loandata2$dti))

## Warning: NAs introducidos por coerción

loandata2$int_rate<-as.numeric(paste(loandata2$int_rate))
loandata2$revol_util<-as.numeric(paste(loandata2$revol_util))

## Warning: NAs introducidos por coerción

loandata2$annual_inc<-as.numeric(paste(loandata2$annual_inc))

## Warning: NAs introducidos por coerción

loandata2$total_acc<-as.numeric(paste(loandata2$total_acc))

## Warning: NAs introducidos por coerción

loandata2$installment<-as.numeric(paste(loandata2$installment))

## Warning: NAs introducidos por coerción

loandata2$delinq_2yrs<-as.numeric(paste(loandata2$delinq_2yrs))

## Warning: NAs introducidos por coerción

loandata2$inq_last_6mths<-as.numeric(paste(loandata2$inq_last_6mths))

## Warning: NAs introducidos por coerción

loandata2$open_acc<-as.numeric(paste(loandata2$open_acc))
loandata2$revol_bal<-as.numeric(paste(loandata2$revol_bal))

## Warning: NAs introducidos por coerción

loandata2$total_pymnt<-as.numeric(paste(loandata2$total_pymnt))

# IMPORTANTE: SI UTILIZAMOS SOLO -as.numeric- sin -paste-, sustituye LOS
VALORES POR VALORES ALEATORIOS  DE
#TIPO NUMERICO.
```

Los variables interest rate y revol_util al estar expresadas en porcentaje y al haberlas convertido a tipo numerico las tenemos que dividir entre 100.

```
loandata2$int_rate <- (loandata2$int_rate/100)
loandata2$revol_util <- (loandata2$revol_util/100)

unique(loandata2$sub_grade)
```

```
##  [1] B2 C4 C5 C1 B5 A4 E1 F2 C3 B1 D1 A1 B3 B4 C2 D2 A3 A5 D5 A2 E4 D3
D4
## [24] F3 E3 F4 F1 E5 G4 E2 G3 G2 G1 F5 G5    64 44 41
## 39 Levels:  41 44 64 A1 A2 A3 A4 A5 B1 B2 B3 B4 B5 C1 C2 C3 C4 C5 ...
G5
```

```
unique(loandata2$grade)
```

```
##  [1] B C A E F D G 0 1 2
## Levels:  0 1 2 A B C D E F G
```

Vemos un summary de los datos que tenemos:

```
loandata2 <- loandata2[,-4]
summary(loandata2)
```

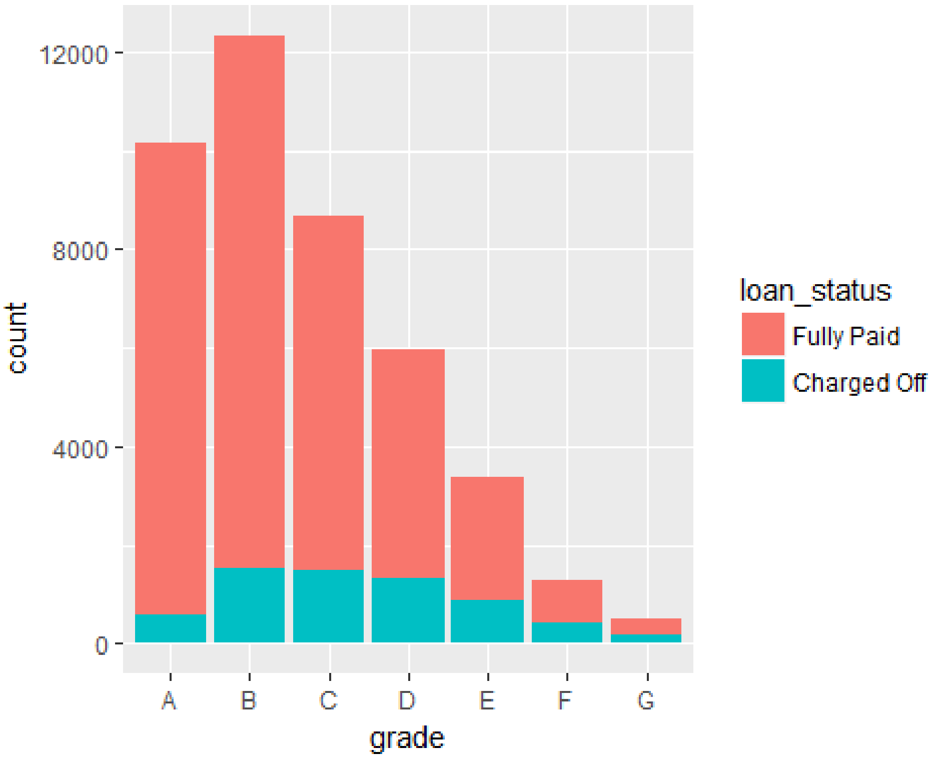
```
##     int_rate       installment          grade         annual_inc
##  Min.   :0.0000   Min.   :  15.67   B      :12349   Min.   :    406
##  1st Qu.:0.0962   1st Qu.: 165.67   A      :10154   1st Qu.:  40000
##  Median :0.1199   Median : 277.86   C      : 8703   Median :  59000
##  Mean   :0.1216   Mean   : 322.76   D      : 5982   Mean   :  69148
##  3rd Qu.:0.1472   3rd Qu.: 428.13   E      : 3367   3rd Qu.:  82500
##  Max.   :0.2459   Max.   :1305.19   F      : 1289   Max.   :6000000
##                   NA's   :7         (Other):  516   NA's   :12
##       loan_status         dti          delinq_2yrs      inq_last_6mths
##             :    0   Min.   :  0.00   Min.   : 0.0000   Min.   :
0.00
##  Fully Paid :35964   1st Qu.:  8.20   1st Qu.: 0.0000   1st Qu.:
0.00
##  Charged Off: 6396   Median : 13.48   Median : 0.0000   Median :
1.00
##                      Mean   : 13.39   Mean   : 0.1691   Mean   :
2.01
##                      3rd Qu.: 18.69   3rd Qu.: 0.0000   3rd Qu.:
2.00
##                      Max.   :476.36   Max.   :85.5162   Max.
:16055.65
##                      NA's   :45       NA's   :2         NA's   :44
##     open_acc        revol_bal        revol_util         total_acc
##  Min.   : 0.000   Min.   :     0   Min.   :  0.0000   Min.   :  1.00
##  1st Qu.: 6.000   1st Qu.:  3644   1st Qu.:  0.2570   1st Qu.: 13.00
##  Median : 9.000   Median :  8827   Median :  0.4970   Median : 20.00
##  Mean   : 9.351   Mean   : 14302   Mean   :  0.6269   Mean   : 22.15
##  3rd Qu.:12.000   3rd Qu.:  17257   3rd Qu.:  0.7280   3rd Qu.: 29.00
##  Max.   :87.000   Max.   :1207359   Max.   :409.1800   Max.   :507.00
##  NA's   :41       NA's   :1         NA's   :7         NA's   :52
##   total_pymnt
##  Min.   :    0
##  1st Qu.: 5459
##  Median : 9680
##  Mean   :12014
##  3rd Qu.:16424
##  Max.   :58886
##  NA's   :7
```

Quitamos los NAs

```
loandata2 <- na.omit(loandata2)
```

A modo ilustrativo, podemos representar la relación entre variables. Vamos a comprobar la relación entre grade y loan_status:

```
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.4.2

ggplot(loandata2, aes(grade)) + geom_bar(aes(fill=loan_status))
```



Podemos observar como, logicamente, aumenta el % de préstamos que entran en default a medida que disminuye el rating.

# Regresion

## Estimacion del modelo

Creamos un nuevo Dataframe con todas las variables menos la variable a predecir Loan Status

```
library(modelr)

## Warning: package 'modelr' was built under R version 3.4.2

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.4.2

## 
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
## 
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

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

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

library(purrr)

## Warning: package 'purrr' was built under R version 3.4.2

## 
## Attaching package: 'purrr'

## The following object is masked from 'package:plyr':
## 
##     compact

library(leaps)

## Warning: package 'leaps' was built under R version 3.4.2
```

Vamos a emplear la metolodogia mas automatica que hay disponible hasta la fecha. Seleccionaremos el modelo con step. El modelo que obtengo es el resultado de sucesivas comparaciones con base en el Cp de Mallow.

```
step(glm(loan_status~., family = "binomial",
data=loandata2),direction='backward')

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Start:  AIC=18059.7
## loan_status ~ int_rate + installment + grade + annual_inc + dti +
##     delinq_2yrs + inq_last_6mths + open_acc + revol_bal + revol_util +
##     total_acc + total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                  Df Deviance   AIC
## - inq_last_6mths  1    18024 18058
## <none>                 18024 18060
## - revol_bal       1    18031 18065
## - open_acc        1    18033 18067
## - total_acc       1    18036 18070
## - delinq_2yrs     1    18040 18074
## - revol_util      1    18043 18077
## - annual_inc      1    18048 18082
## - dti             1    18049 18083
## - grade           6    18113 18137
## - int_rate        1    18323 18357
## - installment     1    28423 28457
## - total_pymnt     1    33708 33742
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=18057.87
## loan_status ~ int_rate + installment + grade + annual_inc + dti +
##     delinq_2yrs + open_acc + revol_bal + revol_util + total_acc +
##     total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##               Df Deviance   AIC
## <none>              18024 18058
## - revol_bal    1    18032 18064
## - open_acc     1    18033 18065
## - total_acc    1    18036 18068
## - delinq_2yrs  1    18040 18072
## - revol_util   1    18043 18075
## - annual_inc   1    18048 18080
## - dti          1    18049 18081
## - grade        6    18115 18137
## - int_rate     1    18326 18358
## - installment  1    28437 28469
## - total_pymnt  1    33908 33940

##
## Call:  glm(formula = loan_status ~ int_rate + installment + grade +
##     annual_inc + dti + delinq_2yrs + open_acc + revol_bal + revol_util
+
##     total_acc + total_pymnt, family = "binomial", data = loandata2)
##
## Coefficients:
## (Intercept)     int_rate  installment       gradeB       gradeC
##  -4.922e+00    3.148e+01    2.826e-02   -2.812e-01   -7.463e-01
##       gradeD       gradeE       gradeF       gradeG   annual_inc
##  -8.846e-01   -5.662e-01   -6.124e-01   -1.472e+00   -2.858e-06
##          dti  delinq_2yrs     open_acc    revol_bal   revol_util
##   1.673e-02   -1.501e-01   -1.820e-02    3.120e-06   -3.529e-01
##    total_acc  total_pymnt
##   8.816e-03   -8.878e-04
##
## Degrees of Freedom: 42306 Total (i.e. Null);  42290 Residual
## Null Deviance:       35920
## Residual Deviance: 18020     AIC: 18060
```

## Crossvalidation

Una vez tenemos un primero modelo, vamos a hacer crossvalidation con kfolds. Utilizaremos el 90% de la muestra para train y el 10% para test. Ningun train sample es igual y, por tanto, ningun test es igual. De esta manera, consigo 10 modelos distintos para poder elegir cual es el mejor en el training y en el testing.

```
#Crossvalidation
set.seed(20171025)
folds <- crossv_kfold(loandata2, k = 10)
folds

## # A tibble: 10 x 3
##              train           test   .id
##             <list>          <list> <chr>
##  1 <S3: resample> <S3: resample>    01
##  2 <S3: resample> <S3: resample>    02
##  3 <S3: resample> <S3: resample>    03
##  4 <S3: resample> <S3: resample>    04
##  5 <S3: resample> <S3: resample>    05
##  6 <S3: resample> <S3: resample>    06
##  7 <S3: resample> <S3: resample>    07
##  8 <S3: resample> <S3: resample>    08
##  9 <S3: resample> <S3: resample>    09
## 10 <S3: resample> <S3: resample>    10

folds$test[[1]]

## <resample [4,231 x 13]> 4, 22, 45, 78, 80, 82, 105, 112, 113, 116, ...

folds$train[[1]]

## <resample [38,076 x 13]> 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, ...
```

Ejecuto el proceso para cada submuestra (step)

```
folds <- folds %>%
  mutate(model = map(train, ~ step(glm(loan_status~., family =
"binomial", data=.),direction='backward'))) %>%
  mutate(aic=map_dbl(model,AIC)) %>%
  mutate(deviance = map2_dbl(model, test, deviance))

## Warning: package 'bindrcpp' was built under R version 3.4.2

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Start:  AIC=16154.63
## loan_status ~ int_rate + installment + grade + annual_inc + dti +
##     delinq_2yrs + inq_last_6mths + open_acc + revol_bal + revol_util +
##     total_acc + total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                  Df Deviance   AIC
## - inq_last_6mths  1    16119 16153
## <none>                 16119 16155
## - revol_bal       1    16124 16158
## - open_acc        1    16128 16162
## - total_acc       1    16129 16163
## - delinq_2yrs     1    16132 16166
## - revol_util      1    16133 16167
## - annual_inc      1    16140 16174
## - dti             1    16141 16175
## - grade           6    16201 16225
## - int_rate        1    16389 16423
## - installment     1    25570 25604
## - total_pymnt     1    30350 30384
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=16152.63
## loan_status ~ int_rate + installment + grade + annual_inc + dti +
##     delinq_2yrs + open_acc + revol_bal + revol_util + total_acc +
##     total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##               Df Deviance   AIC
## <none>              16119 16153
## - revol_bal    1    16124 16156
## - open_acc     1    16128 16160
## - total_acc    1    16129 16161
## - delinq_2yrs  1    16132 16164
## - revol_util   1    16133 16165
## - annual_inc   1    16140 16172
## - dti          1    16141 16173
## - grade        6    16202 16224
## - int_rate     1    16390 16422
## - installment  1    25585 25617
## - total_pymnt  1    30535 30567

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Start:  AIC=16323.93
## loan_status ~ int_rate + installment + grade + annual_inc + dti +
##     delinq_2yrs + inq_last_6mths + open_acc + revol_bal + revol_util +
##     total_acc + total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                   Df Deviance   AIC
## - inq_last_6mths  1    16288 16322
## <none>                 16288 16324
## - revol_bal       1    16296 16330
## - open_acc        1    16297 16331
## - total_acc       1    16298 16332
## - delinq_2yrs     1    16301 16335
## - revol_util      1    16302 16336
## - annual_inc      1    16306 16340
## - dti             1    16310 16344
## - grade           6    16366 16390
## - int_rate        1    16554 16588
## - installment     1    25608 25642
## - total_pymnt     1    30323 30357

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=16322.15
## loan_status ~ int_rate + installment + grade + annual_inc + dti +
##     delinq_2yrs + open_acc + revol_bal + revol_util + total_acc +
##     total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                  Df Deviance   AIC
## <none>                 16288 16322
## - revol_bal    1    16297 16329
## - open_acc     1    16297 16329
## - total_acc    1    16298 16330
## - delinq_2yrs  1    16301 16333
## - revol_util   1    16302 16334
## - annual_inc   1    16306 16338
## - dti          1    16310 16342
## - grade        6    16368 16390
## - int_rate     1    16557 16589
## - installment  1    25620 25652
## - total_pymnt  1    30493 30525

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Start:  AIC=16173.89
## loan_status ~ int_rate + installment + grade + annual_inc + dti +
##     delinq_2yrs + inq_last_6mths + open_acc + revol_bal + revol_util +
##     total_acc + total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                   Df Deviance   AIC
## - inq_last_6mths  1    16138 16172
## <none>                 16138 16174
## - revol_bal       1    16144 16178
## - open_acc        1    16147 16181
## - total_acc       1    16149 16183
## - annual_inc      1    16153 16187
## - revol_util      1    16156 16190
## - delinq_2yrs     1    16157 16191
## - dti             1    16166 16200
## - grade           6    16214 16238
## - int_rate        1    16410 16444
## - installment     1    25535 25569
## - total_pymnt     1    30349 30383

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=16172.51
## loan_status ~ int_rate + installment + grade + annual_inc + dti +
##     delinq_2yrs + open_acc + revol_bal + revol_util + total_acc +
##     total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
##               Df Deviance   AIC
## <none>              16138 16172
## - revol_bal    1    16144 16176
## - open_acc     1    16147 16179
## - total_acc    1    16149 16181
## - annual_inc   1    16154 16186
## - revol_util   1    16156 16188
## - delinq_2yrs  1    16157 16189
## - dti          1    16166 16198
## - grade        6    16216 16238
## - int_rate     1    16414 16446
## - installment  1    25544 25576
## - total_pymnt  1    30517 30549

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Start:  AIC=16322.88
## loan_status ~ int_rate + installment + grade + annual_inc + dti + 
##     delinq_2yrs + inq_last_6mths + open_acc + revol_bal + revol_util + 
##     total_acc + total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                  Df Deviance   AIC
## - inq_last_6mths  1    16287 16321
## <none>                 16287 16323
## - revol_bal       1    16295 16329
## - open_acc        1    16297 16331
## - revol_util      1    16304 16338
## - total_acc       1    16306 16340
## - delinq_2yrs     1    16307 16341
## - annual_inc      1    16307 16341
## - dti             1    16308 16342
## - grade           6    16365 16389
## - int_rate        1    16556 16590
## - installment     1    25586 25620
## - total_pymnt     1    30299 30333

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=16321.05
## loan_status ~ int_rate + installment + grade + annual_inc + dti +
##     delinq_2yrs + open_acc + revol_bal + revol_util + total_acc +
##     total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                Df Deviance   AIC
## <none>              16287 16321
## - revol_bal    1    16295 16327
## - open_acc     1    16297 16329
## - revol_util   1    16304 16336
## - total_acc    1    16306 16338
## - delinq_2yrs  1    16307 16339
## - annual_inc   1    16308 16340
## - dti          1    16308 16340
## - grade        6    16367 16389
## - int_rate     1    16559 16591
## - installment  1    25600 25632
## - total_pymnt  1    30483 30515
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Start:  AIC=16331.15
## loan_status ~ int_rate + installment + grade + annual_inc + dti +
##     delinq_2yrs + inq_last_6mths + open_acc + revol_bal + revol_util +
##     total_acc + total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                  Df Deviance   AIC
## - inq_last_6mths  1    16295 16329
## <none>                 16295 16331
## - revol_bal       1    16301 16335
## - open_acc        1    16302 16336
## - delinq_2yrs     1    16306 16340
## - total_acc       1    16306 16340
## - revol_util      1    16310 16344
## - dti             1    16314 16348
## - annual_inc      1    16327 16361
## - grade           6    16375 16399
## - int_rate        1    16563 16597
## - installment     1    25711 25745
## - total_pymnt     1    30409 30443

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=16329.16
## loan_status ~ int_rate + installment + grade + annual_inc + dti +
##     delinq_2yrs + open_acc + revol_bal + revol_util + total_acc + 
##     total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##               Df Deviance   AIC
## <none>              16295 16329
## - revol_bal    1    16301 16333
## - open_acc     1    16302 16334
## - delinq_2yrs  1    16306 16338
## - total_acc    1    16306 16338
## - revol_util   1    16311 16343
## - dti          1    16314 16346
## - annual_inc   1    16327 16359
## - grade        6    16376 16398
## - int_rate     1    16565 16597
## - installment  1    25728 25760
## - total_pymnt  1    30596 30628
## Start:  AIC=16164.88
## loan_status ~ int_rate + installment + grade + annual_inc + dti + 
##     delinq_2yrs + inq_last_6mths + open_acc + revol_bal + revol_util + 
##     total_acc + total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                  Df Deviance   AIC
## - inq_last_6mths  1    16129 16163
## <none>                 16129 16165
## - open_acc        1    16135 16169
## - revol_bal       1    16137 16171
## - total_acc       1    16137 16171
## - revol_util      1    16144 16178
## - delinq_2yrs     1    16145 16179
## - annual_inc      1    16149 16183
## - dti             1    16152 16186
## - grade           6    16216 16240
## - int_rate        1    16404 16438
## - installment     1    25474 25508
## - total_pymnt     1    30225 30259
##
## Step:  AIC=16163.33
## loan_status ~ int_rate + installment + grade + annual_inc + dti +
##     delinq_2yrs + open_acc + revol_bal + revol_util + total_acc +
##     total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                Df Deviance   AIC
## <none>              16129 16163
## - open_acc     1    16135 16167
## - total_acc    1    16137 16169
## - revol_bal    1    16137 16169
## - revol_util   1    16144 16176
## - delinq_2yrs  1    16145 16177
## - annual_inc   1    16150 16182
## - dti          1    16152 16184
## - grade        6    16219 16241
## - int_rate     1    16408 16440
## - installment  1    25484 25516
## - total_pymnt  1    30397 30429

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Start:  AIC=16236.78
## loan_status ~ int_rate + installment + grade + annual_inc + dti +
##     delinq_2yrs + inq_last_6mths + open_acc + revol_bal + revol_util +
##     total_acc + total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                   Df Deviance   AIC
## - inq_last_6mths  1    16201 16235
## <none>                 16201 16237
## - revol_bal       1    16207 16241
## - open_acc        1    16208 16242
## - total_acc       1    16210 16244
## - delinq_2yrs     1    16216 16250
## - revol_util      1    16220 16254
## - annual_inc      1    16221 16255
## - dti             1    16221 16255
## - grade           6    16274 16298
## - int_rate        1    16455 16489
## - installment     1    25663 25697
## - total_pymnt     1    30472 30506

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=16234.82
## loan_status ~ int_rate + installment + grade + annual_inc + dti +
##     delinq_2yrs + open_acc + revol_bal + revol_util + total_acc +
##     total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
##                Df Deviance   AIC
## <none>              16201 16235
## - revol_bal    1    16207 16239
## - open_acc     1    16208 16240
## - total_acc    1    16210 16242
## - delinq_2yrs  1    16216 16248
## - revol_util   1    16220 16252
## - annual_inc   1    16221 16253
## - dti          1    16221 16253
## - grade        6    16276 16298
## - int_rate     1    16457 16489
## - installment  1    25677 25709
## - total_pymnt  1    30662 30694

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Start:  AIC=16336.5
## loan_status ~ int_rate + installment + grade + annual_inc + dti + 
##     delinq_2yrs + inq_last_6mths + open_acc + revol_bal + revol_util + 
##     total_acc + total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                   Df Deviance   AIC
## - inq_last_6mths  1    16301 16335
## <none>                 16300 16336
## - open_acc        1    16306 16340
## - revol_bal       1    16309 16343
## - delinq_2yrs     1    16313 16347
## - total_acc       1    16314 16348
## - revol_util      1    16317 16351
## - dti             1    16318 16352
## - annual_inc      1    16331 16365
## - grade           6    16390 16414
## - int_rate        1    16579 16613
## - installment     1    25568 25602
## - total_pymnt     1    30301 30335

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=16335.03
## loan_status ~ int_rate + installment + grade + annual_inc + dti +
##     delinq_2yrs + open_acc + revol_bal + revol_util + total_acc +
##     total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##               Df Deviance   AIC
## <none>              16301 16335
## - open_acc     1    16307 16339
## - revol_bal    1    16309 16341
## - delinq_2yrs  1    16313 16345
## - total_acc    1    16314 16346
## - revol_util   1    16317 16349
## - dti          1    16319 16351
## - annual_inc   1    16332 16364
## - grade        6    16392 16414
## - int_rate     1    16583 16615
## - installment  1    25583 25615
## - total_pymnt  1    30491 30523
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Start:  AIC=16174.9
## loan_status ~ int_rate + installment + grade + annual_inc + dti +
##     delinq_2yrs + inq_last_6mths + open_acc + revol_bal + revol_util +
##     total_acc + total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                  Df Deviance   AIC
## - inq_last_6mths  1    16139 16173
## <none>                 16139 16175
## - revol_bal       1    16145 16179
## - open_acc        1    16148 16182
## - total_acc       1    16151 16185
## - revol_util      1    16157 16191
## - delinq_2yrs     1    16158 16192
## - dti             1    16163 16197
## - annual_inc      1    16163 16197
## - grade           6    16216 16240
## - int_rate        1    16401 16435
## - installment     1    25607 25641
## - total_pymnt     1    30443 30477

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Step:  AIC=16173.31
## loan_status ~ int_rate + installment + grade + annual_inc + dti +
##     delinq_2yrs + open_acc + revol_bal + revol_util + total_acc + 
##     total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                Df Deviance   AIC
## <none>              16139 16173
## - revol_bal     1    16145 16177
## - open_acc      1    16148 16180
## - total_acc     1    16151 16183
## - revol_util    1    16157 16189
## - delinq_2yrs   1    16158 16190
## - dti           1    16163 16195
## - annual_inc    1    16164 16196
## - grade         6    16219 16241
## - int_rate      1    16405 16437
## - installment   1    25619 25651
## - total_pymnt   1    30624 30656

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Start:  AIC=16329.45
## loan_status ~ int_rate + installment + grade + annual_inc + dti + 
##     delinq_2yrs + inq_last_6mths + open_acc + revol_bal + revol_util + 
##     total_acc + total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                   Df Deviance   AIC
## - inq_last_6mths  1    16294 16328
## <none>                 16294 16330
## - revol_bal       1    16300 16334
## - total_acc       1    16302 16336
## - open_acc        1    16303 16337
## - delinq_2yrs     1    16304 16338
## - annual_inc      1    16309 16343
## - revol_util      1    16316 16350
## - dti             1    16323 16357
## - grade           6    16379 16403
## - int_rate        1    16576 16610
## - installment     1    25459 25493
## - total_pymnt     1    30177 30211

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## 
## Step:  AIC=16327.48
## loan_status ~ int_rate + installment + grade + annual_inc + dti + 
##     delinq_2yrs + open_acc + revol_bal + revol_util + total_acc + 
##     total_pymnt

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##               Df Deviance   AIC
## <none>              16294 16328
## - revol_bal    1    16300 16332
## - total_acc    1    16302 16334
## - open_acc     1    16303 16335
## - delinq_2yrs  1    16304 16336
## - annual_inc   1    16310 16342
## - revol_util   1    16317 16349
## - dti          1    16323 16355
## - grade        6    16380 16402
## - int_rate     1    16578 16610
## - installment  1    25472 25504
## - total_pymnt  1    30352 30384
```

```
folds %>%
  select(.id, aic, deviance)
```

```
## # A tibble: 10 x 3
##      .id      aic deviance
##    <chr>    <dbl>    <dbl>
##  1    01 16152.63 16118.63
##  2    02 16322.15 16288.15
##  3    03 16172.51 16138.51
##  4    04 16321.05 16287.05
##  5    05 16329.16 16295.16
##  6    06 16163.33 16129.33
##  7    07 16234.82 16200.82
##  8    08 16335.03 16301.03
##  9    09 16173.31 16139.31
## 10    10 16327.48 16293.48
```

```
folds$aic
```

```
##        1        2        3        4        5        6        7 
8 
## 16152.63 16322.15 16172.51 16321.05 16329.16 16163.33 16234.82 
16335.03 
##        9       10 
## 16173.31 16327.48
```

Como podemos comprobar el modelo que presenta una menor AIC es el 1 que por tanto es el que vamos a seleccionar.

```
folds$model[1]

## $`1`
## 
## Call:  glm(formula = loan_status ~ int_rate + installment + grade + 
##     annual_inc + dti + delinq_2yrs + open_acc + revol_bal + revol_util 
+ 
##     total_acc + total_pymnt, family = "binomial", data = .)
## 
## Coefficients:
## (Intercept)     int_rate  installment       gradeB       gradeC  
##  -4.919e+00    3.160e+01    2.830e-02   -2.990e-01   -7.843e-01  
##      gradeD       gradeE       gradeF       gradeG   annual_inc  
##  -9.147e-01   -6.172e-01   -6.218e-01   -1.441e+00   -2.844e-06  
##         dti  delinq_2yrs     open_acc    revol_bal   revol_util  
##   1.682e-02   -1.427e-01   -1.931e-02    2.969e-06   -3.322e-01  
##   total_acc  total_pymnt  
##   8.785e-03   -8.900e-04  
## 
## Degrees of Freedom: 38075 Total (i.e. Null);  38059 Residual
## Null Deviance:       32310 
## Residual Deviance: 16120     AIC: 16150
```

Guardamos en un Dataframe la muestra que se ha utilizado para el entrenamiento en el modelo 1.

```
df_train <- data.frame(folds$train[1])

head(df_train)

##   X1.int_rate X1.installment X1.grade X1.annual_inc X1.loan_status
X1.dti
## 1      0.1065         162.87        B         24000     Fully Paid
27.65
## 2      0.1527          59.83        C         30000    Charged Off
1.00
## 3      0.1596          84.33        C         12252     Fully Paid
8.72
## 5      0.1269          67.79        B         80000     Fully Paid
17.94
## 6      0.0790         156.46        A         36000     Fully Paid
11.20
## 7      0.1596         170.08        C         47004     Fully Paid
23.51
##   X1.delinq_2yrs X1.inq_last_6mths X1.open_acc X1.revol_bal
X1.revol_util
## 1              0                 1           3        13648
0.837
## 2              0                 5           3         1687
0.094
```

```
## 3              0                 2           2         2956
0.985
## 5              0                 0          15        27783
0.539
## 6              0                 3           9         7963
0.283
## 7              0                 1           7        17726
0.856
##   X1.total_acc X1.total_pymnt
## 1            9       5863.155
## 2            4       1014.530
## 3           10       3005.667
## 5           38       4066.908
## 6           12       5632.210
## 7           11      10137.840
```

Cambiamos el nombre de las columnas quitando X1.

```
names <- colnames(df_train)
names <- gsub('X1.','',names, fixed=TRUE)
colnames(df_train) <- names
```

Guardamos en un Dataframe la muestra que se ha utilizado para el test en el modelo 1

```
df_test <- data.frame(folds$test[1])

head(df_test)

##    X1.int_rate X1.installment X1.grade X1.annual_inc X1.loan_status
X1.dti
## 4       0.1349         339.31        C         49200     Fully Paid
20.00
## 22      0.1242         701.73        B        105000    Charged Off
13.22
## 45      0.0603         182.62        A         45600     Fully Paid
5.34
## 78      0.2167         197.51        F         75000     Fully Paid
24.82
## 80      0.1991         475.99        E         65000     Fully Paid
6.81
## 82      0.1427         343.09        C         68000     Fully Paid
15.39
##    X1.delinq_2yrs X1.inq_last_6mths X1.open_acc X1.revol_bal
X1.revol_util
## 4               0                 1          10         5598
0.210
## 22              0                 0           7        32135
0.903
## 45              0                 1           6         3378
0.325
## 78              0                 2           9        21706
```

```
0.912
## 80              0                 0          10         11745
0.778
## 82              0                 1           9         11303
0.813
##    X1.total_acc X1.total_pymnt
## 4            37      12231.890
## 22           38      14034.600
## 45           28       6065.860
## 78           19       8204.774
## 80           40      26034.660
## 82           15      10672.894
```

Cambiamos el nombre de las columnas quitando X1.

```
names <- colnames(df_test)
names <- gsub('X1.','',names, fixed=TRUE)
colnames(df_test) <- names
```

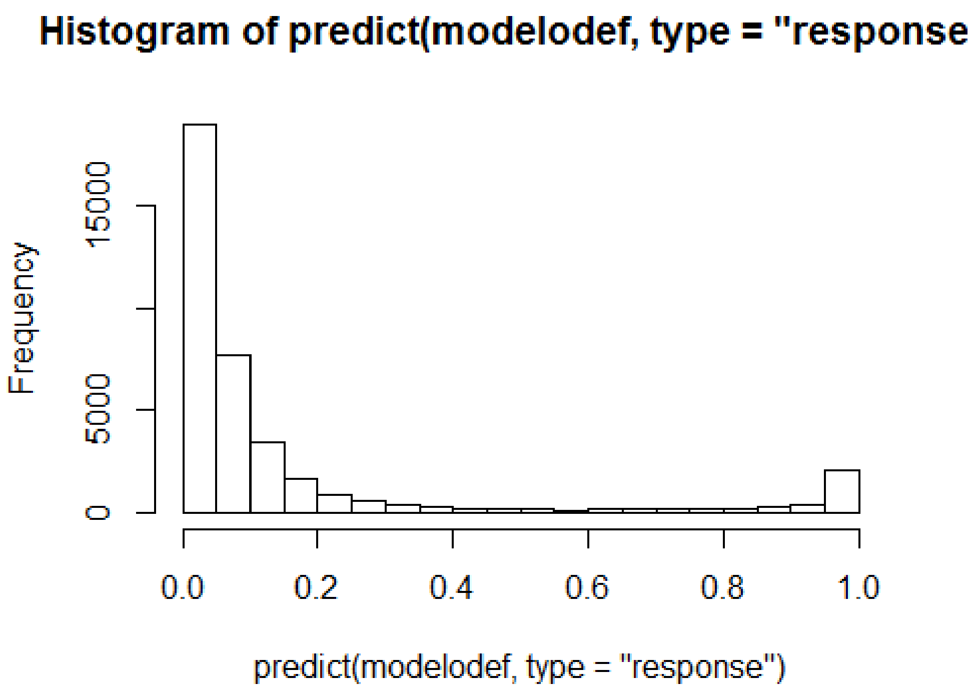
Guardamos el modelo definitivo como "modelodef"

```
modelodef <- glm(formula = loan_status ~ int_rate + installment + grade +
    annual_inc + dti + delinq_2yrs + open_acc + revol_bal + revol_util +
    total_acc + total_pymnt, family = "binomial", data = df_train)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(modelodef)

##
## Call:
## glm(formula = loan_status ~ int_rate + installment + grade +
##     annual_inc + dti + delinq_2yrs + open_acc + revol_bal + revol_util
+
##     total_acc + total_pymnt, family = "binomial", data = df_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6178  -0.3915  -0.2498  -0.0284   6.6257
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.919e+00  1.666e-01 -29.534  < 2e-16 ***
## int_rate     3.160e+01  1.947e+00  16.229  < 2e-16 ***
## installment  2.830e-02  4.794e-04  59.037  < 2e-16 ***
## gradeB      -2.990e-01  9.513e-02  -3.143 0.001672 **
## gradeC      -7.843e-01  1.346e-01  -5.828 5.61e-09 ***
## gradeD      -9.147e-01  1.689e-01  -5.415 6.13e-08 ***
## gradeE      -6.172e-01  1.983e-01  -3.112 0.001859 **
## gradeF      -6.218e-01  2.479e-01  -2.508 0.012127 *
## gradeG      -1.441e+00  3.048e-01  -4.726 2.29e-06 ***
## annual_inc  -2.844e-06  6.518e-07  -4.363 1.28e-05 ***
## dti          1.682e-02  3.536e-03   4.758 1.96e-06 ***
## delinq_2yrs -1.427e-01  4.069e-02  -3.507 0.000453 ***
## open_acc    -1.931e-02  6.492e-03  -2.975 0.002929 **
## revol_bal    2.969e-06  1.147e-06   2.589 0.009627 **
## revol_util  -3.322e-01  8.626e-02  -3.851 0.000117 ***
## total_acc    8.785e-03  2.667e-03   3.294 0.000987 ***
## total_pymnt -8.900e-04  1.404e-05 -63.398  < 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: 32306  on 38075  degrees of freedom
## Residual deviance: 16119  on 38059  degrees of freedom
## AIC: 16153
##
## Number of Fisher Scoring iterations: 7
```

## Predicción dentro y fuera de la muestra

### Dentro de la muestra(Training)

```
hist(predict(modelodef,type="response"))
```



```
table(predict(modelodef,type="response")>0.20)

##
## FALSE  TRUE
## 31798  6278
```

Vemos como, con un punto de corte de 0,20 en la muestra de entrenamiento, habría 31.798 individuos con una probabilidad de Default inferior al 20% y 6.278 con probabilidad superior.

```
prob.modelodf.insample <- predict(modelodef,type="response")
predicted_modelodf_insample <- predict(modelodef,type="response")>0.20
predicted_modelodf_insample <- as.numeric(predicted_modelodf_insample)
```

### Creamos la matriz de confusion

```
matriz_confusion_train <-
table(df_train$loan_status,predicted_modelodf_insample,dnn=c("Truth","Pre
dicted"))
matriz_confusion_train <- as.data.frame(matriz_confusion_train)
filas <- c(1,4)
matriz_confusion_train <- matriz_confusion_train[-filas,]
matriz_confusion_train

##         Truth Predicted  Freq
## 2  Fully Paid         0 30672
## 3 Charged Off         0  1126
## 5  Fully Paid         1  1659
## 6 Charged Off         1  4619
```

Tasa de error

```
Verdadero_positivo <- matriz_confusion_train$Freq[1]
Falso_positivo <- matriz_confusion_train$Freq[2]
Falso_negativo <- matriz_confusion_train$Freq[3]
Verdadero_negativo <- matriz_confusion_train$Freq[4]
Total <- sum(matriz_confusion_train$Freq)

Tasa_de_error <- (Falso_positivo+Falso_negativo)/Total
Tasa_de_error

## [1] 0.07314319
```

## Aciertos positivos (*sensitivity*) y aciertos negativos (*specificity*)

```
aciertos_positivos <- Verdadero_positivo/(Verdadero_positivo +
Falso_negativo)
aciertos_positivos

## [1] 0.948687
```

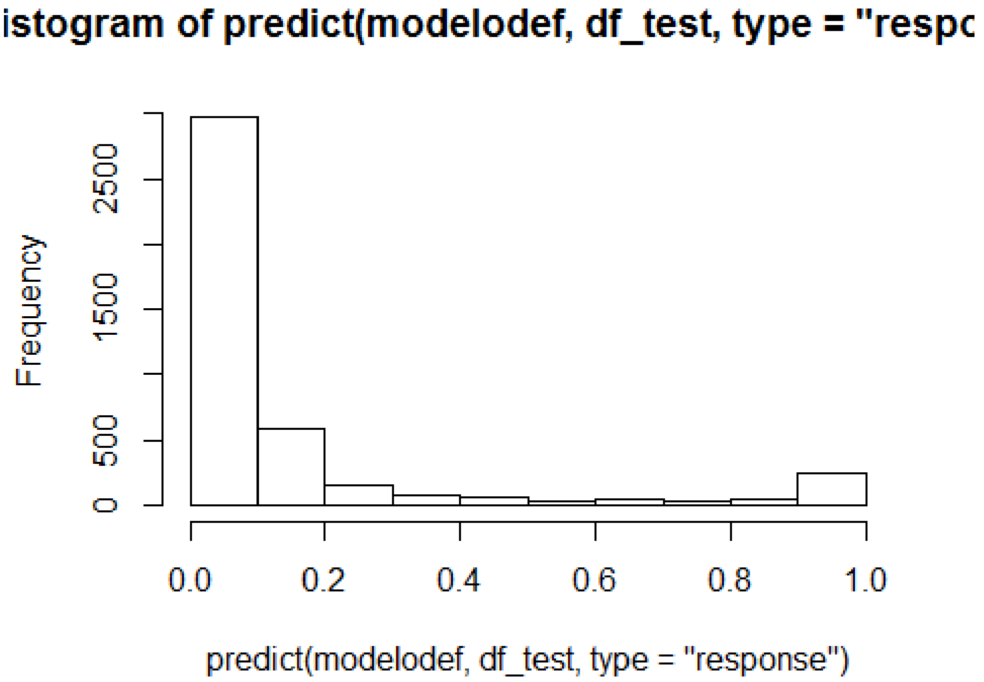
```
aciertos_negativos <-
Verdadero_negativo/(Verdadero_negativo+Falso_positivo)
aciertos_negativos

## [1] 0.8040035
```

Como podemos observar, en la muestra de entrenamiento el modelo tiene una tasa de éxito muy alta con respecto a clasificar como Fully Paid a un individuo pero presenta una mayor tasa de error a la hora de clasificar como charged Off (Default).

Por tanto nuestro modelo no es muy efectivo a la hora de predecir los casos de Default. Esto nos sugiere que debemos ser prudentes a la hora de utilizar el mismo, seleccionando un criterio de corte prudente (cortaremos con probabilidades bajas).

### Fuera de la muestra(Test)

```
hist(predict(modelodef,df_test,type="response"))
```



```
table(predict(modelodef,df_test,type="response")>0.20)

##
## FALSE  TRUE
##  3554   677
```

Con un punto de corte de 0,20 en el TEST habría 3.554 individuos con una probabilidad de Default inferior al 20% y 677 con probabilidad superior.

```
prob.modelodf.outsample <- predict(modelodef,df_test,type="response")
predicted_modelodf_outsample <-
predict(modelodef,df_test,type="response")>0.20
predicted_modelodf_outsample <- as.numeric(predicted_modelodf_outsample)
```

### Creamos la matriz de confusion

```
matriz_confusion_test <-
table(df_test$loan_status,predicted_modelodf_outsample,dnn=c("Truth","Pre
dicted"))
matriz_confusion_test <- as.data.frame(matriz_confusion_test)
filas <- c(1,4)
matriz_confusion_test <- matriz_confusion_test[-filas,]
matriz_confusion_test

##         Truth Predicted Freq
## 2  Fully Paid         0 3413
## 3 Charged Off         0  141
## 5  Fully Paid         1  173
## 6 Charged Off         1  504
```

## Tasa de error

```
Verdadero_positivo <- matriz_confusion_test$Freq[1]
Falso_positivo <- matriz_confusion_test$Freq[2]
Falso_negativo <- matriz_confusion_test$Freq[3]
Verdadero_negativo <- matriz_confusion_test$Freq[4]
Total <- sum(matriz_confusion_test$Freq)

Tasa_de_error <- (Falso_positivo+Falso_negativo)/Total
Tasa_de_error

## [1] 0.07421413
```

## Aciertos positivos (*sensitivity*) y aciertos negativos (*specificity*)

```
aciertos_positivos <- Verdadero_positivo/(Verdadero_positivo +
Falso_negativo)
aciertos_positivos

## [1] 0.9517568

aciertos_negativos <-
Verdadero_negativo/(Verdadero_negativo+Falso_positivo)
aciertos_negativos

## [1] 0.7813953
```

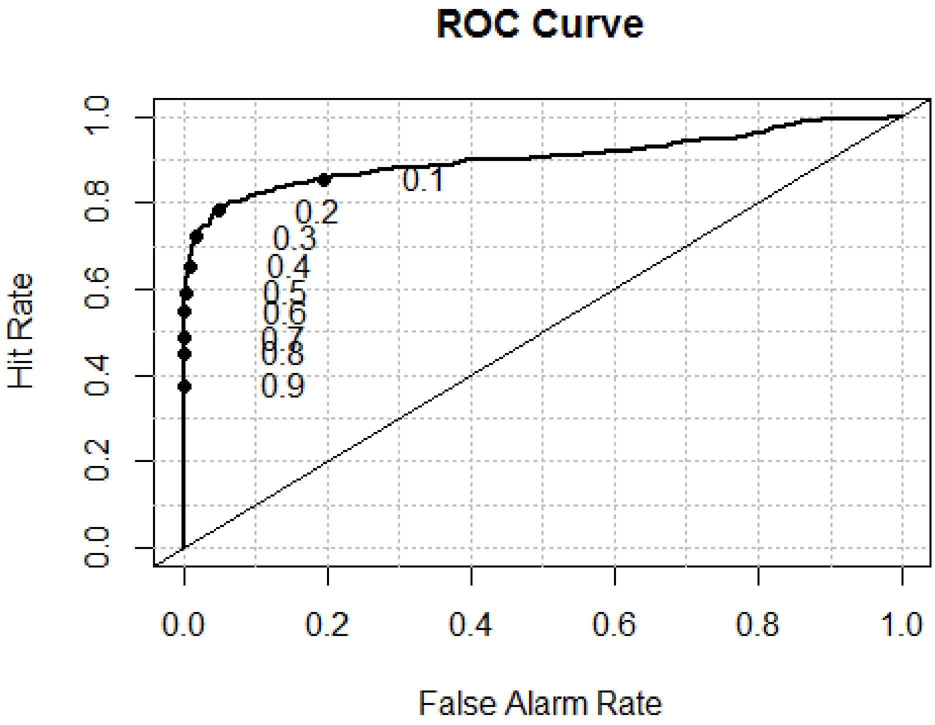
En este caso se cumplen unas condiciones prácticamente idénticas a las obtenidas dentro de la muestra, con algo mas de *sensitivity* y algo menos de *specificity.*

En el test se corrobora que nuestro modelo no es muy efectivo a la hora de predecir los casos de Default, por lo que seremos prudentes a la hora de utilizar el mismo, seleccionando un criterio de corte prudente (cortaremos con probabilidades bajas).

## Curva ROC

Representación de la curva ROC para fuera del training (TEST)

```
library(verification)

## Warning: package 'verification' was built under R version 3.4.2

## Loading required package: fields

## Warning: package 'fields' was built under R version 3.4.2

## Loading required package: spam

## Warning: package 'spam' was built under R version 3.4.2

## Loading required package: dotCall64

## Warning: package 'dotCall64' was built under R version 3.4.2

## Loading required package: grid

## Spam version 2.1-1 (2017-07-02) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: maps

## Warning: package 'maps' was built under R version 3.4.2

##
## Attaching package: 'maps'

## The following object is masked from 'package:purrr':
##
##     map

## The following object is masked from 'package:plyr':
##
##     ozone
## Loading required package: boot

## Loading required package: CircStats

## Warning: package 'CircStats' was built under R version 3.4.2

## Loading required package: MASS

## Warning: package 'MASS' was built under R version 3.4.2

## 
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
## 
##     select

## Loading required package: dtw

## Warning: package 'dtw' was built under R version 3.4.2

## Loading required package: proxy

## Warning: package 'proxy' was built under R version 3.4.2

## 
## Attaching package: 'proxy'

## The following object is masked from 'package:spam':
## 
##     as.matrix

## The following objects are masked from 'package:stats':
## 
##     as.dist, dist

## The following object is masked from 'package:base':
## 
##     as.matrix

## Loaded dtw v1.18-1. See ?dtw for help, citation("dtw") for use in
publication.

roc.plot(df_test$loan_status == "Charged Off", prob.modelodf.outsample)

## Warning in roc.plot.default(df_test$loan_status == "Charged Off",
## prob.modelodf.outsample): Large amount of unique predictions used as
## thresholds. Consider specifying thresholds.
```

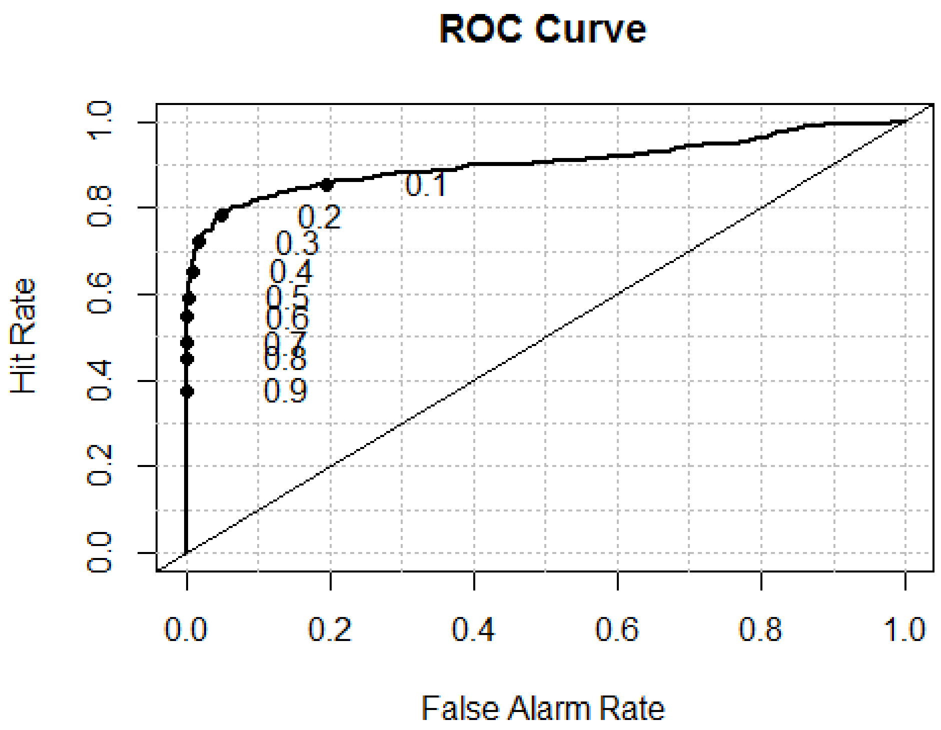


La curva de Roc crece muy lentamente a partir de un false alarm Rate de un 10% (proporcion de Falso negativo entre el total de valores negativos reales) con un respectivo Hit rate del 80% (proporcion de verdaderos positivos sobre el total de valores positivos reales).

Por tanto a priori siendo conservadores a partir de un 20% de probabilidad deberíamos considerar como Charged Off o Default al individuo.

Calculamos el área de la curva de ROC.

```
roc.plot(df_test$loan_status == "Charged Off",
prob.modelodf.outsample)$roc.vol

## Warning in roc.plot.default(df_test$loan_status == "Charged Off",
## prob.modelodf.outsample): Large amount of unique predictions used as
## thresholds. Consider specifying thresholds.
```



```
##      Model      Area       p.value binorm.area
## 1 Model   1 0.9021738 5.474014e-233          NA
```

El área de nuestro modelo es un 0,90. Al ser un número cercano a 1 nos indica la bondad de nuestro modelo.

## Función de costes:

Vamos a calcular la función para poder hacer valoraciones posteriores. Para ello, ponderaremos el coste de dar un prestamo a quien no va a pagar como 10 veces mas alto que el de no concederselo a alguien que realmente fuera a pagarlo. La razón de esto es que el coste de un default es mucho mayor que potencial beneficio que obtendría el prestamista (interés).

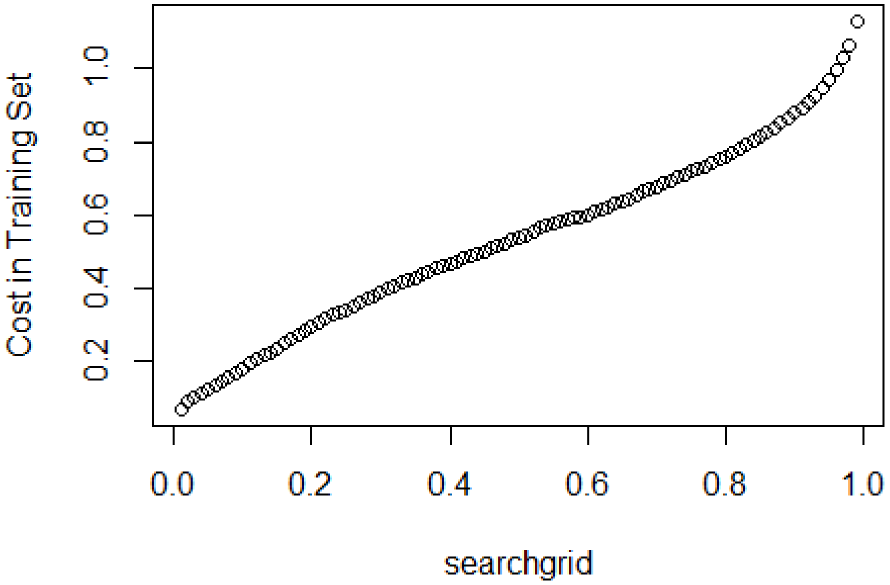
```
searchgrid = seq(0.01, 0.99, 0.01)
```

El resultado es una matriz de 99 filas y 2 columnas, la primera columna contiene la cut-off p y la segunda el coste

```
result = cbind(searchgrid, NA)

cost1 <- function(r, pi){
        weight1 = 10
        weight0 = 1
        c1 = (r=="Charged Off")&(pi<pcut) #logical vector - true if
actual 1 but predict 0
        c0 = (r=="Fully paid")&(pi>pcut) #logical vector - true if actual
0 but predict 1
        return(mean(weight1*c1+weight0*c0))
}
```

in the cost function, both r and pi are vectors, r=truth, pi=predicted probability

```
for(i in 1:length(searchgrid)) {
        pcut <- result[i,1]
        result[i,2] <- cost1(df_train$loan_status,
prob.modelodf.insample)
        }

head(result)

##      searchgrid
## [1,]       0.01 0.07012291
## [2,]       0.02 0.09008299
## [3,]       0.03 0.10163883
## [4,]       0.04 0.11319466
## [5,]       0.05 0.12396260
## [6,]       0.06 0.13341738

plot(result, ylab="Cost in Training Set")
```



## Conclusiones

Como se puede observar la función de costes sigue un comportamiento prácticamente lineal, es decir para mejorar el ratio de acierto se debe incurrir en un mayor coste en todo momento.

Por ello nos resulta complicado tomar decisiones de corte en base a este criterio. Tendremos en cuenta la curva de Roc como criterio para seleccionar nuestro cut off o punto a partir del cual asignamos a un inviduo como Default.

Tras el análisis del comportamiento de nuestra curva hemos decidio seleccionar el 20% como cutoff probability ya que en ese punto tenemos un alto porcentaje de acierto (80%) con un bajo porcentajr de falsa alarma (cercano al 10%). Adicionalmente nuestro ratio de acierto a la hora de clasificar un individuo como Default es de un 78% frente al 95% de clasificarlo correctamente como no Default. Por tanto también nos invita a ser prudentes a la hora de elegir la probabilidad de corte.

Finalmente para aumentar nuestro porcentaje de acierto tendríamos que reducer, aún mas si cabe, nuestra cutoff probabibility, lo que implicaría tener un porcentaje de falsa alarma mucho mas alto. Esto último supondría dejar de conceder préstamos a muchos individuos y por tanto reducir mucho el volumen de negocio.