**Caso Practico Final**

Alfred Burquier Reyes

22 de marzo de 2019

**PREGUNTAS**

Tomaremos el dataset de aprobación de crédito bancario en <https://archive.ics.uci.edu/ml/datasets/Credit+Approval> .

1.- Carga los datos. Realiza una inspección por variables de la distribución de aprobación de crédito en función de cada atributo visualmente. Realiza las observaciones pertinentes. ¿ Qué variables son mejores para separar los datos?.

2.- Prepara el dataset convenientemente e imputa los valores faltantes usando la librería missForest.

3.- Divide el dataset tomando las primeras 590 instancias como train y las últimas 100 como test.

4.- Entrena un modelo de regresión logística con regularización Ridge y Lasso en train. seleccionando el que mejor AUC tenga. Da las métricas en test.

5.- Aporta los log odds de las variables predictoras sobre la variable objetivo.

6.- Si por cada verdadero positivo ganamos 100e y por cada falso positivo perdemos 20e. ¿ Qué rentabilidad aporta aplicar este modelo?.

**DESARROLLO**

**1.- Carga los datos. Realiza una inspección por variables de la distribución de aprobación de crédito en función de cada atributo visualmente. Realiza las observaciones pertinentes. ¿ Qué variables son mejores para separar los datos?.**

url <- "<https://archive.ics.uci.edu/ml/machine-learning-databases/credit-screening/crx.data>"  
crx <- **read.csv**(url,na.strings = "?", header = FALSE)  
**write.csv**(crx, file='crx.csv')  
**dim**(crx)

## [1] 690 16

**str**(crx)

## 'data.frame': 690 obs. of 16 variables:  
## $ V1 : Factor w/ 2 levels "a","b": 2 1 1 2 2 2 2 1 2 2 ...  
## $ V2 : num 30.8 58.7 24.5 27.8 20.2 ...  
## $ V3 : num 0 4.46 0.5 1.54 5.62 ...  
## $ V4 : Factor w/ 3 levels "l","u","y": 2 2 2 2 2 2 2 2 3 3 ...  
## $ V5 : Factor w/ 3 levels "g","gg","p": 1 1 1 1 1 1 1 1 3 3 ...  
## $ V6 : Factor w/ 14 levels "aa","c","cc",..: 13 11 11 13 13 10 12 3 9 13 ...  
## $ V7 : Factor w/ 9 levels "bb","dd","ff",..: 8 4 4 8 8 8 4 8 4 8 ...  
## $ V8 : num 1.25 3.04 1.5 3.75 1.71 ...  
## $ V9 : Factor w/ 2 levels "f","t": 2 2 2 2 2 2 2 2 2 2 ...  
## $ V10: Factor w/ 2 levels "f","t": 2 2 1 2 1 1 1 1 1 1 ...  
## $ V11: int 1 6 0 5 0 0 0 0 0 0 ...  
## $ V12: Factor w/ 2 levels "f","t": 1 1 1 2 1 2 2 1 1 2 ...  
## $ V13: Factor w/ 3 levels "g","p","s": 1 1 1 1 3 1 1 1 1 1 ...  
## $ V14: int 202 43 280 100 120 360 164 80 180 52 ...  
## $ V15: int 0 560 824 3 0 0 31285 1349 314 1442 ...  
## $ V16: Factor w/ 2 levels "-","+": 2 2 2 2 2 2 2 2 2 2 ...

**summary**(crx)

## V1 V2 V3 V4 V5   
## a :210 Min. :13.75 Min. : 0.000 l : 2 g :519   
## b :468 1st Qu.:22.60 1st Qu.: 1.000 u :519 gg : 2   
## NA's: 12 Median :28.46 Median : 2.750 y :163 p :163   
## Mean :31.57 Mean : 4.759 NA's: 6 NA's: 6   
## 3rd Qu.:38.23 3rd Qu.: 7.207   
## Max. :80.25 Max. :28.000   
## NA's :12   
## V6 V7 V8 V9 V10   
## c :137 v :399 Min. : 0.000 f:329 f:395   
## q : 78 h :138 1st Qu.: 0.165 t:361 t:295   
## w : 64 bb : 59 Median : 1.000   
## i : 59 ff : 57 Mean : 2.223   
## aa : 54 j : 8 3rd Qu.: 2.625   
## (Other):289 (Other): 20 Max. :28.500   
## NA's : 9 NA's : 9   
## V11 V12 V13 V14 V15 V16   
## Min. : 0.0 f:374 g:625 Min. : 0 Min. : 0.0 -:383   
## 1st Qu.: 0.0 t:316 p: 8 1st Qu.: 75 1st Qu.: 0.0 +:307   
## Median : 0.0 s: 57 Median : 160 Median : 5.0   
## Mean : 2.4 Mean : 184 Mean : 1017.4   
## 3rd Qu.: 3.0 3rd Qu.: 276 3rd Qu.: 395.5   
## Max. :67.0 Max. :2000 Max. :100000.0   
## NA's :13

La mejor variable para separar los datos es el factor V16, porque separa los datos en………

**2.- Prepara el dataset convenientemente e imputa los valores faltantes usando la librería missForest.**

*#install.packages("missForest", dependencies = TRUE)*  
**library**(missForest)

## Warning: package 'missForest' was built under R version 3.5.3

## Loading required package: randomForest

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

## Loading required package: foreach

## Loading required package: itertools

## Loading required package: iterators

crx.imp <- **missForest**(crx) *# Cargo el dataset crx en missForest*

## missForest iteration 1 in progress...done!  
## missForest iteration 2 in progress...done!  
## missForest iteration 3 in progress...done!  
## missForest iteration 4 in progress...done!

*#crx.imp$ximp*  
crx.imp**$**OOBerror *#OOB indicar el error, NRMSE es el error cuadratico medio normalizado (entre mas cercano a 0 este valor mejor el rendimiento) y PFC es la proporcion de entradas faltantes clasificadas*

## NRMSE PFC   
## 0.03181175 0.13596418

crx2 <- crx.imp**$**ximp *# crx2 es el nuevo dataset sin valores nulos.*  
**summary**(crx2)

## V1 V2 V3 V4 V5 V6   
## a:213 Min. :13.75 Min. : 0.000 l: 2 g :525 c :137   
## b:477 1st Qu.:22.67 1st Qu.: 1.000 u:525 gg: 2 q : 78   
## Median :28.62 Median : 2.750 y:163 p :163 w : 64   
## Mean :31.60 Mean : 4.759 i : 63   
## 3rd Qu.:38.06 3rd Qu.: 7.207 aa : 55   
## Max. :80.25 Max. :28.000 ff : 54   
## (Other):239   
## V7 V8 V9 V10 V11 V12   
## v :401 Min. : 0.000 f:329 f:395 Min. : 0.0 f:374   
## h :138 1st Qu.: 0.165 t:361 t:295 1st Qu.: 0.0 t:316   
## bb : 63 Median : 1.000 Median : 0.0   
## ff : 58 Mean : 2.223 Mean : 2.4   
## j : 9 3rd Qu.: 2.625 3rd Qu.: 3.0   
## z : 8 Max. :28.500 Max. :67.0   
## (Other): 13   
## V13 V14 V15 V16   
## g:625 Min. : 0.0 Min. : 0.0 -:383   
## p: 8 1st Qu.: 80.0 1st Qu.: 0.0 +:307   
## s: 57 Median : 160.0 Median : 5.0   
## Mean : 183.4 Mean : 1017.4   
## 3rd Qu.: 272.0 3rd Qu.: 395.5   
## Max. :2000.0 Max. :100000.0   
##

**dim**(crx2)

## [1] 690 16

**3.- Divide el dataset tomando las primeras 590 instancias como train y las últimas 100 como test.**

crx2**$**V16 <- **as.numeric**(crx2**$**V16) **-**1   
crx2\_train <- crx2[1**:**590, 1**:**16]  
crx2\_test <- crx2[591**:**690, 1**:**16]  
X <- crx2[,1**:**15]  
y <- crx2**$**V16   
**unique**(y)

## [1] 1 0

X\_train <- X[1**:**590,]  
y\_train <- y[1**:**590]  
X\_test <- X[591**:**690,]  
y\_test <- y[591**:**690]

Se dividio el dataset crx2 en primeras 590 instancias como X\_train y y\_train, las últimas 100 como X\_test y y\_test.

**4.- Entrena un modelo de regresión logística con regularización Ridge y Lasso en train. seleccionando el que mejor AUC tenga. Da las métricas en test.**

Primero realizare una regresion logistica mediante AIC.

*#help("glm")*  
fit1 <- **glm**(V16**~**., data=crx2, family=binomial)

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

**summary**(fit1)

##   
## Call:  
## glm(formula = V16 ~ ., family = binomial, data = crx2)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.5981 -0.3029 -0.1347 0.3987 3.3718   
##   
## Coefficients: (2 not defined because of singularities)  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.056e+01 7.537e+02 0.014 0.98882   
## V1b -1.629e-01 3.146e-01 -0.518 0.60469   
## V2 6.533e-03 1.289e-02 0.507 0.61222   
## V3 -2.197e-02 2.884e-02 -0.762 0.44620   
## V4u -1.424e+01 7.537e+02 -0.019 0.98492   
## V4y -1.507e+01 7.537e+02 -0.020 0.98405   
## V5gg NA NA NA NA   
## V5p NA NA NA NA   
## V6c 4.504e-01 5.078e-01 0.887 0.37516   
## V6cc 1.638e+00 7.672e-01 2.136 0.03270 \*   
## V6d 3.349e-01 7.832e-01 0.428 0.66895   
## V6e 2.199e+00 1.156e+00 1.903 0.05708 .   
## V6ff -3.176e+00 2.116e+00 -1.501 0.13334   
## V6i -2.784e-02 6.966e-01 -0.040 0.96813   
## V6j -3.768e+00 2.205e+00 -1.709 0.08750 .   
## V6k -9.600e-02 6.216e-01 -0.154 0.87726   
## V6m 4.334e-01 6.814e-01 0.636 0.52475   
## V6q 6.082e-01 5.580e-01 1.090 0.27572   
## V6r -1.488e+00 4.275e+00 -0.348 0.72776   
## V6w 1.119e+00 5.744e-01 1.949 0.05134 .   
## V6x 2.983e+00 9.150e-01 3.260 0.00112 \*\*   
## V7dd -8.703e-01 1.803e+00 -0.483 0.62935   
## V7ff 2.330e+00 2.036e+00 1.144 0.25248   
## V7h 6.935e-01 5.659e-01 1.226 0.22037   
## V7j 4.615e+00 2.111e+00 2.186 0.02878 \*   
## V7n 3.194e+00 1.584e+00 2.016 0.04376 \*   
## V7o -3.120e+01 4.745e+07 0.000 1.00000   
## V7v 4.078e-01 5.214e-01 0.782 0.43419   
## V7z -2.933e+00 1.752e+00 -1.674 0.09404 .   
## V8 6.598e-02 5.055e-02 1.305 0.19180   
## V9t 3.778e+00 3.512e-01 10.758 < 2e-16 \*\*\*  
## V10t 5.797e-01 3.731e-01 1.554 0.12020   
## V11 1.223e-01 5.858e-02 2.087 0.03689 \*   
## V12t -2.569e-01 2.817e-01 -0.912 0.36173   
## V13p 4.160e+00 9.323e-01 4.462 8.11e-06 \*\*\*  
## V13s -1.372e-02 4.902e-01 -0.028 0.97768   
## V14 -2.597e-03 9.132e-04 -2.844 0.00445 \*\*   
## V15 5.614e-04 1.859e-04 3.019 0.00253 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 948.16 on 689 degrees of freedom  
## Residual deviance: 392.54 on 654 degrees of freedom  
## AIC: 464.54  
##   
## Number of Fisher Scoring iterations: 14

fit0 <- **glm**(V16**~**1, data=crx2, family=binomial)  
**summary**(fit0)

##   
## Call:  
## glm(formula = V16 ~ 1, family = binomial, data = crx2)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.085 -1.085 -1.085 1.273 1.273   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.2212 0.0766 -2.887 0.00388 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 948.16 on 689 degrees of freedom  
## Residual deviance: 948.16 on 689 degrees of freedom  
## AIC: 950.16  
##   
## Number of Fisher Scoring iterations: 3

**library**(MASS)  
step <-**stepAIC**(fit0,direction="forward",scope=**list**(upper=fit1,lower=fit0))

## Start: AIC=950.16  
## V16 ~ 1

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

## Df Deviance AIC  
## + V9 1 540.95 544.95  
## + V11 1 762.74 766.74  
## + V10 1 798.66 802.66  
## + V15 1 862.80 866.80  
## + V8 1 863.32 867.32  
## + V6 13 847.77 875.77  
## + V7 8 901.94 919.94  
## + V3 1 918.36 922.36  
## + V4 2 919.84 925.84  
## + V5 2 919.84 925.84  
## + V2 1 930.23 934.23  
## + V13 2 938.55 944.55  
## + V14 1 940.61 944.61  
## <none> 948.16 950.16  
## + V1 1 947.40 951.40  
## + V12 1 947.47 951.47  
##   
## Step: AIC=544.95  
## V16 ~ V9

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

## Df Deviance AIC  
## + V11 1 503.21 509.21  
## + V15 1 505.09 511.09  
## + V10 1 507.67 513.67  
## + V4 2 518.66 526.66  
## + V5 2 518.66 526.66  
## + V13 2 521.04 529.04  
## + V6 13 505.32 535.32  
## + V8 1 532.97 538.97  
## + V14 1 536.27 542.27  
## <none> 540.95 544.95  
## + V12 1 539.26 545.26  
## + V3 1 539.69 545.69  
## + V7 8 526.46 546.46  
## + V1 1 540.54 546.54  
## + V2 1 540.72 546.72  
##   
## Step: AIC=509.21  
## V16 ~ V9 + V11

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

## Df Deviance AIC  
## + V15 1 477.87 485.87  
## + V13 2 484.00 494.00  
## + V4 2 484.15 494.15  
## + V5 2 484.15 494.15  
## + V6 13 470.98 502.98  
## + V10 1 498.33 506.33  
## + V8 1 499.68 507.68  
## <none> 503.21 509.21  
## + V14 1 501.32 509.32  
## + V7 8 487.34 509.34  
## + V12 1 501.99 509.99  
## + V1 1 503.11 511.11  
## + V2 1 503.16 511.16  
## + V3 1 503.19 511.19

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

##   
## Step: AIC=485.87  
## V16 ~ V9 + V11 + V15

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

## Df Deviance AIC  
## + V13 2 462.37 474.37  
## + V5 2 464.59 476.59  
## + V4 2 464.59 476.59  
## + V6 13 446.80 480.80  
## + V10 1 473.35 483.35  
## + V8 1 473.93 483.93  
## + V14 1 475.45 485.45  
## <none> 477.87 485.87  
## + V12 1 476.52 486.52  
## + V3 1 477.73 487.73  
## + V1 1 477.77 487.77  
## + V2 1 477.81 487.81  
## + V7 8 464.61 488.61

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

##   
## Step: AIC=474.37  
## V16 ~ V9 + V11 + V15 + V13

## 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  
## + V5 2 449.66 465.66  
## + V4 2 449.66 465.66  
## + V6 13 428.00 466.00  
## + V10 1 456.49 470.49  
## + V8 1 457.91 471.91  
## + V14 1 460.22 474.22  
## <none> 462.37 474.37  
## + V7 8 447.52 475.52  
## + V12 1 461.69 475.69  
## + V2 1 462.11 476.11  
## + V1 1 462.25 476.25  
## + V3 1 462.36 476.36

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

##   
## Step: AIC=465.66  
## V16 ~ V9 + V11 + V15 + V13 + V5

## 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  
## + V6 13 417.81 459.81  
## + V10 1 444.44 462.44  
## + V7 8 432.19 464.19  
## + V8 1 446.69 464.69  
## + V14 1 446.85 464.85  
## <none> 449.66 465.66  
## + V12 1 448.79 466.79  
## + V2 1 449.21 467.21  
## + V3 1 449.63 467.63  
## + V1 1 449.65 467.65

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

##   
## Step: AIC=459.81  
## V16 ~ V9 + V11 + V15 + V13 + V5 + V6

## 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  
## + V14 1 409.82 453.82  
## + V10 1 415.37 459.37  
## <none> 417.81 459.81  
## + V12 1 416.35 460.35  
## + V8 1 416.66 460.66  
## + V1 1 417.59 461.59  
## + V3 1 417.70 461.70  
## + V2 1 417.78 461.78  
## + V7 8 407.40 465.40

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

##   
## Step: AIC=453.82  
## V16 ~ V9 + V11 + V15 + V13 + V5 + V6 + V14

## 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  
## + V10 1 407.47 453.47  
## <none> 409.82 453.82  
## + V8 1 408.75 454.75  
## + V3 1 409.00 455.00  
## + V12 1 409.12 455.12  
## + V1 1 409.69 455.69  
## + V2 1 409.80 455.80  
## + V7 8 398.67 458.67

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

##   
## Step: AIC=453.47  
## V16 ~ V9 + V11 + V15 + V13 + V5 + V6 + V14 + V10

## 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> 407.47 453.47  
## + V8 1 406.17 454.17  
## + V3 1 406.82 454.82  
## + V12 1 407.02 455.02  
## + V2 1 407.37 455.37  
## + V1 1 407.40 455.40  
## + V7 8 396.10 458.10

**summary**(step)

##   
## Call:  
## glm(formula = V16 ~ V9 + V11 + V15 + V13 + V5 + V6 + V14 + V10,   
## family = binomial, data = crx2)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.8369 -0.3277 -0.1547 0.4170 3.0894   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.192e+00 4.976e-01 -6.415 1.41e-10 \*\*\*  
## V9t 3.658e+00 3.145e-01 11.632 < 2e-16 \*\*\*  
## V11 1.156e-01 5.704e-02 2.026 0.042755 \*   
## V15 5.254e-04 1.739e-04 3.022 0.002514 \*\*   
## V13p 4.041e+00 8.944e-01 4.518 6.25e-06 \*\*\*  
## V13s 1.400e-01 4.674e-01 0.299 0.764619   
## V5gg 1.606e+01 6.758e+02 0.024 0.981041   
## V5p -8.132e-01 3.189e-01 -2.550 0.010773 \*   
## V6c 5.354e-01 4.761e-01 1.125 0.260720   
## V6cc 1.723e+00 7.206e-01 2.390 0.016830 \*   
## V6d 2.465e-01 7.469e-01 0.330 0.741382   
## V6e 9.557e-01 8.440e-01 1.132 0.257509   
## V6ff -1.133e+00 7.116e-01 -1.592 0.111363   
## V6i -2.501e-01 5.938e-01 -0.421 0.673618   
## V6j 2.435e-01 1.106e+00 0.220 0.825713   
## V6k -1.323e-02 5.927e-01 -0.022 0.982193   
## V6m 3.824e-01 6.505e-01 0.588 0.556641   
## V6q 7.574e-01 5.324e-01 1.423 0.154810   
## V6r 1.088e+00 3.712e+00 0.293 0.769382   
## V6w 1.016e+00 5.509e-01 1.845 0.065107 .   
## V6x 2.951e+00 8.584e-01 3.438 0.000586 \*\*\*  
## V14 -2.332e-03 8.431e-04 -2.766 0.005680 \*\*   
## V10t 5.594e-01 3.639e-01 1.537 0.124201   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 948.16 on 689 degrees of freedom  
## Residual deviance: 407.47 on 667 degrees of freedom  
## AIC: 453.47  
##   
## Number of Fisher Scoring iterations: 14

El mejor modelo de regresión logística con AIC es V16 ~ V9 + V11 + V15 + V13 + V5 + V6 + V14 + V10

y\_pred <- **as.numeric**(**predict**(step, X\_test)**>**.5)  
*#install.packages(c("e1071", "caret", "ggplot2","lattice","class","gmodels"))*  
**library**(caret)

## Warning: package 'caret' was built under R version 3.5.3

## Loading required package: lattice

## Warning: package 'lattice' was built under R version 3.5.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.5.3

##   
## Attaching package: 'ggplot2'

## The following object is masked from 'package:randomForest':  
##   
## margin

**library**(ggplot2)  
**library**(lattice)  
**library**(e1071)

## Warning: package 'e1071' was built under R version 3.5.3

**library**(class)  
**library**(gmodels)

## Warning: package 'gmodels' was built under R version 3.5.3

*#help("confusionMatrix") #La matriz de confusión y las métricas de erorr se pueden calcular. Para obtener todas las que se han dado incluir la opción mode="everything"*  
**confusionMatrix**(**table**(y\_test,y\_pred), mode="everything")

## Confusion Matrix and Statistics  
##   
## y\_pred  
## y\_test 0 1  
## 0 85 1  
## 1 7 7  
##   
## Accuracy : 0.92   
## 95% CI : (0.8484, 0.9648)  
## No Information Rate : 0.92   
## P-Value [Acc > NIR] : 0.5926   
##   
## Kappa : 0.5951   
## Mcnemar's Test P-Value : 0.0771   
##   
## Sensitivity : 0.9239   
## Specificity : 0.8750   
## Pos Pred Value : 0.9884   
## Neg Pred Value : 0.5000   
## Precision : 0.9884   
## Recall : 0.9239   
## F1 : 0.9551   
## Prevalence : 0.9200   
## Detection Rate : 0.8500   
## Detection Prevalence : 0.8600   
## Balanced Accuracy : 0.8995   
##   
## 'Positive' Class : 0   
##

Efectuaremos las regresiones logísticas con regularización Ridge y Lasso.

Regresión logística con regularización Ridge

X\_train2 <- **data.matrix**(X\_train)  
X\_test2 <- **data.matrix**(X\_test)  
*#install.packages("glmnet")*  
**library**(glmnet)

## Warning: package 'glmnet' was built under R version 3.5.3

## Loading required package: Matrix

## Loaded glmnet 2.0-16

**set.seed**(999)  
cv.ridge <- **cv.glmnet**(X\_train2, y\_train, family='binomial', alpha=0, parallel=TRUE, standardize=TRUE, type.measure='auc')

## Warning: executing %dopar% sequentially: no parallel backend registered

**plot**(cv.ridge) *# Resultados*



cv.ridge**$**lambda.min *#este es el mejor valor de lambda*

## [1] 0.09810657

**max**(cv.ridge**$**cvm) *#este es el valor del error que se estima para ese valor lambda mínimo dado en MSE*

## [1] 0.9082055

**coef**(cv.ridge, s=cv.ridge**$**lambda.min) *#coeficientes*

## 16 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) -3.5484434628  
## V1 -0.0531014119  
## V2 0.0016396833  
## V3 0.0135595185  
## V4 -0.2871069354  
## V5 -0.0857979809  
## V6 0.0313370801  
## V7 0.0101809734  
## V8 0.0651199381  
## V9 1.8286573840  
## V10 0.6578820357  
## V11 0.0596169179  
## V12 -0.0195708074  
## V13 -0.0846654566  
## V14 -0.0006502294  
## V15 0.0000405046

*#métricas en el test*   
y\_pred2 <- **as.numeric**(**predict.glmnet**(cv.ridge**$**glmnet.fit, newx=X\_test2, s=cv.ridge**$**lambda.min)**>**.5)  
**confusionMatrix**(**table**(y\_test,y\_pred2), mode="everything")

## Confusion Matrix and Statistics  
##   
## y\_pred2  
## y\_test 0 1  
## 0 86 0  
## 1 8 6  
##   
## Accuracy : 0.92   
## 95% CI : (0.8484, 0.9648)  
## No Information Rate : 0.94   
## P-Value [Acc > NIR] : 0.85371   
##   
## Kappa : 0.5633   
## Mcnemar's Test P-Value : 0.01333   
##   
## Sensitivity : 0.9149   
## Specificity : 1.0000   
## Pos Pred Value : 1.0000   
## Neg Pred Value : 0.4286   
## Precision : 1.0000   
## Recall : 0.9149   
## F1 : 0.9556   
## Prevalence : 0.9400   
## Detection Rate : 0.8600   
## Detection Prevalence : 0.8600   
## Balanced Accuracy : 0.9574   
##   
## 'Positive' Class : 0   
##

Regresión logística con regularización Lasso

**set.seed**(999)  
cv.lasso <- **cv.glmnet**(X\_train2, y\_train, family='binomial', alpha=1, parallel=TRUE, standardize=TRUE, type.measure='auc')  
**plot**(cv.lasso)*# Resultados*



cv.lasso**$**lambda.min *#este es el mejor valor de lambda*

## [1] 0.01796074

**max**(cv.lasso**$**cvm)*#este es el valor del error que se estima para ese valor lambda mínimo dado en MSE*

## [1] 0.9121825

**coef**(cv.lasso, s=cv.lasso**$**lambda.min) *#coeficientes*

## 16 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) -5.5227206921  
## V1 .   
## V2 .   
## V3 .   
## V4 -0.3812652724  
## V5 .   
## V6 0.0199752966  
## V7 .   
## V8 0.0362709630  
## V9 3.0274603985  
## V10 0.7109072156  
## V11 0.0624529393  
## V12 .   
## V13 .   
## V14 -0.0004539130  
## V15 0.0000467045

y\_pred3 <- **as.numeric**(**predict.glmnet**(cv.lasso**$**glmnet.fit, newx=X\_test2, s=cv.lasso**$**lambda.min)**>**.5)  
**confusionMatrix**(**table**(y\_test,y\_pred3), mode="everything")

## Confusion Matrix and Statistics  
##   
## y\_pred3  
## y\_test 0 1  
## 0 85 1  
## 1 7 7  
##   
## Accuracy : 0.92   
## 95% CI : (0.8484, 0.9648)  
## No Information Rate : 0.92   
## P-Value [Acc > NIR] : 0.5926   
##   
## Kappa : 0.5951   
## Mcnemar's Test P-Value : 0.0771   
##   
## Sensitivity : 0.9239   
## Specificity : 0.8750   
## Pos Pred Value : 0.9884   
## Neg Pred Value : 0.5000   
## Precision : 0.9884   
## Recall : 0.9239   
## F1 : 0.9551   
## Prevalence : 0.9200   
## Detection Rate : 0.8500   
## Detection Prevalence : 0.8600   
## Balanced Accuracy : 0.8995   
##   
## 'Positive' Class : 0   
##

El modelo de Ridge tiene mayor precision y mejor AUC.

**5.- Aporta los log odds de las variables predictoras sobre la variable objetivo.**

**help**("coef")

## starting httpd help server ... done

logistic\_ridge<- **glm**(V16**~**., data=crx2, family=binomial)

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

logistic\_AIC<- **glm**(y **~** V9 **+** V11 **+** V15 **+** V13 **+** V5 **+** V6 **+** V14 **+** V10, data=crx2, family=binomial)*# Este es usando el modelo por AIC*

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

**summary**(logistic\_ridge)

##   
## Call:  
## glm(formula = V16 ~ ., family = binomial, data = crx2)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.5981 -0.3029 -0.1347 0.3987 3.3718   
##   
## Coefficients: (2 not defined because of singularities)  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.056e+01 7.537e+02 0.014 0.98882   
## V1b -1.629e-01 3.146e-01 -0.518 0.60469   
## V2 6.533e-03 1.289e-02 0.507 0.61222   
## V3 -2.197e-02 2.884e-02 -0.762 0.44620   
## V4u -1.424e+01 7.537e+02 -0.019 0.98492   
## V4y -1.507e+01 7.537e+02 -0.020 0.98405   
## V5gg NA NA NA NA   
## V5p NA NA NA NA   
## V6c 4.504e-01 5.078e-01 0.887 0.37516   
## V6cc 1.638e+00 7.672e-01 2.136 0.03270 \*   
## V6d 3.349e-01 7.832e-01 0.428 0.66895   
## V6e 2.199e+00 1.156e+00 1.903 0.05708 .   
## V6ff -3.176e+00 2.116e+00 -1.501 0.13334   
## V6i -2.784e-02 6.966e-01 -0.040 0.96813   
## V6j -3.768e+00 2.205e+00 -1.709 0.08750 .   
## V6k -9.600e-02 6.216e-01 -0.154 0.87726   
## V6m 4.334e-01 6.814e-01 0.636 0.52475   
## V6q 6.082e-01 5.580e-01 1.090 0.27572   
## V6r -1.488e+00 4.275e+00 -0.348 0.72776   
## V6w 1.119e+00 5.744e-01 1.949 0.05134 .   
## V6x 2.983e+00 9.150e-01 3.260 0.00112 \*\*   
## V7dd -8.703e-01 1.803e+00 -0.483 0.62935   
## V7ff 2.330e+00 2.036e+00 1.144 0.25248   
## V7h 6.935e-01 5.659e-01 1.226 0.22037   
## V7j 4.615e+00 2.111e+00 2.186 0.02878 \*   
## V7n 3.194e+00 1.584e+00 2.016 0.04376 \*   
## V7o -3.120e+01 4.745e+07 0.000 1.00000   
## V7v 4.078e-01 5.214e-01 0.782 0.43419   
## V7z -2.933e+00 1.752e+00 -1.674 0.09404 .   
## V8 6.598e-02 5.055e-02 1.305 0.19180   
## V9t 3.778e+00 3.512e-01 10.758 < 2e-16 \*\*\*  
## V10t 5.797e-01 3.731e-01 1.554 0.12020   
## V11 1.223e-01 5.858e-02 2.087 0.03689 \*   
## V12t -2.569e-01 2.817e-01 -0.912 0.36173   
## V13p 4.160e+00 9.323e-01 4.462 8.11e-06 \*\*\*  
## V13s -1.372e-02 4.902e-01 -0.028 0.97768   
## V14 -2.597e-03 9.132e-04 -2.844 0.00445 \*\*   
## V15 5.614e-04 1.859e-04 3.019 0.00253 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 948.16 on 689 degrees of freedom  
## Residual deviance: 392.54 on 654 degrees of freedom  
## AIC: 464.54  
##   
## Number of Fisher Scoring iterations: 14

**summary**(logistic\_AIC)

##   
## Call:  
## glm(formula = y ~ V9 + V11 + V15 + V13 + V5 + V6 + V14 + V10,   
## family = binomial, data = crx2)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.8369 -0.3277 -0.1547 0.4170 3.0894   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.192e+00 4.976e-01 -6.415 1.41e-10 \*\*\*  
## V9t 3.658e+00 3.145e-01 11.632 < 2e-16 \*\*\*  
## V11 1.156e-01 5.704e-02 2.026 0.042755 \*   
## V15 5.254e-04 1.739e-04 3.022 0.002514 \*\*   
## V13p 4.041e+00 8.944e-01 4.518 6.25e-06 \*\*\*  
## V13s 1.400e-01 4.674e-01 0.299 0.764619   
## V5gg 1.606e+01 6.758e+02 0.024 0.981041   
## V5p -8.132e-01 3.189e-01 -2.550 0.010773 \*   
## V6c 5.354e-01 4.761e-01 1.125 0.260720   
## V6cc 1.723e+00 7.206e-01 2.390 0.016830 \*   
## V6d 2.465e-01 7.469e-01 0.330 0.741382   
## V6e 9.557e-01 8.440e-01 1.132 0.257509   
## V6ff -1.133e+00 7.116e-01 -1.592 0.111363   
## V6i -2.501e-01 5.938e-01 -0.421 0.673618   
## V6j 2.435e-01 1.106e+00 0.220 0.825713   
## V6k -1.323e-02 5.927e-01 -0.022 0.982193   
## V6m 3.824e-01 6.505e-01 0.588 0.556641   
## V6q 7.574e-01 5.324e-01 1.423 0.154810   
## V6r 1.088e+00 3.712e+00 0.293 0.769382   
## V6w 1.016e+00 5.509e-01 1.845 0.065107 .   
## V6x 2.951e+00 8.584e-01 3.438 0.000586 \*\*\*  
## V14 -2.332e-03 8.431e-04 -2.766 0.005680 \*\*   
## V10t 5.594e-01 3.639e-01 1.537 0.124201   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 948.16 on 689 degrees of freedom  
## Residual deviance: 407.47 on 667 degrees of freedom  
## AIC: 453.47  
##   
## Number of Fisher Scoring iterations: 14

Para el modelo de ridge como el valor de p-values es superior a 0.05 en todas las variables el log odds no aporta en las variables, en cambio se usaramos el modelo de regresión logística con AIC si hay variables que aportan los log odds de las variables predictoras sobre la variable objetivo debido que hay el p-values es superior a 0.05 en algunas varaibles.

**6.- Si por cada verdadero positivo ganamos 100e y por cada falso positivo perdemos 20e. ¿ Qué rentabilidad aporta aplicar este modelo?.**