

# script\_glm.R

sergio

2020-05-26

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 3.0-2
```

```
source('../utils/utils_oblig.R')
```

```
## Loading required package: caret
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
## Registered S3 methods overwritten by 'ggplot2':  
##   method      from  
##   [.quosures  rlang  
##   c.quosures  rlang  
##   print.quosures rlang
```

```
set.seed(117)
```

```
script.name <- 'glm'
```

```
script.date <- date()
```

```
script.start <- Sys.time()
```

```
print('Start')
```

```
## [1] "Start"
```

```
# leer el archivo dataset.csv de la carpeta

dataset <- read.csv('../data/dataset.csv')

# ver la estructura del dataset

# str(dataset)

# asignar el nombre del jugador como nombre de la fila

rownames(dataset) <- dataset$CustomerID

df <- na.omit(dataset[,-1])

df$ServiceArea <- NULL

print('** Distribucion a-priori de la variable a predecir')
```

```
## [1] "** Distribucion a-priori de la variable a predecir"
```

```
print(prop.table(table(df$Churn)))
```

```
##
##           No           Yes
## 0.7131871 0.2868129
```

```
df.part <- train_dev_partition(df, p = 0.9)

df.fn_summary <- fn_summaryUtility

df.metric <- 'utility'

df.form <- Churn ~ .

print('** GLM')
```

```
## [1] "** GLM"
```

```
df.glm.ctrl <- trainControl(method = 'cv',
                           number = 5,
                           verboseIter = TRUE,
                           search = 'random',
                           summaryFunction = df.fn_summary)

df.glm <- train(form = df.form,
               data = df.part$train,
               method = 'glmnet',
               family = 'binomial',
               trControl = df.glm.ctrl,
               tuneLength = 10,
               metric = df.metric)
```

```
## + Fold1: alpha=0.26189, lambda=0.382909
## - Fold1: alpha=0.26189, lambda=0.382909
## + Fold1: alpha=0.17301, lambda=0.539936
## - Fold1: alpha=0.17301, lambda=0.539936
## + Fold1: alpha=0.89094, lambda=0.018498
## - Fold1: alpha=0.89094, lambda=0.018498
## + Fold1: alpha=0.03789, lambda=0.005650
## - Fold1: alpha=0.03789, lambda=0.005650
## + Fold1: alpha=0.26030, lambda=0.003013
## - Fold1: alpha=0.26030, lambda=0.003013
## + Fold1: alpha=0.51916, lambda=0.008256
## - Fold1: alpha=0.51916, lambda=0.008256
## + Fold1: alpha=0.92198, lambda=0.005467
## - Fold1: alpha=0.92198, lambda=0.005467
## + Fold1: alpha=0.48300, lambda=0.002933
## - Fold1: alpha=0.48300, lambda=0.002933
## + Fold1: alpha=0.56814, lambda=0.014619
## - Fold1: alpha=0.56814, lambda=0.014619
## + Fold1: alpha=0.46533, lambda=1.625179
## - Fold1: alpha=0.46533, lambda=1.625179
## + Fold2: alpha=0.26189, lambda=0.382909
## - Fold2: alpha=0.26189, lambda=0.382909
## + Fold2: alpha=0.17301, lambda=0.539936
## - Fold2: alpha=0.17301, lambda=0.539936
## + Fold2: alpha=0.89094, lambda=0.018498
## - Fold2: alpha=0.89094, lambda=0.018498
## + Fold2: alpha=0.03789, lambda=0.005650
## - Fold2: alpha=0.03789, lambda=0.005650
## + Fold2: alpha=0.26030, lambda=0.003013
## - Fold2: alpha=0.26030, lambda=0.003013
## + Fold2: alpha=0.51916, lambda=0.008256
## - Fold2: alpha=0.51916, lambda=0.008256
## + Fold2: alpha=0.92198, lambda=0.005467
## - Fold2: alpha=0.92198, lambda=0.005467
## + Fold2: alpha=0.48300, lambda=0.002933
## - Fold2: alpha=0.48300, lambda=0.002933
## + Fold2: alpha=0.56814, lambda=0.014619
## - Fold2: alpha=0.56814, lambda=0.014619
## + Fold2: alpha=0.46533, lambda=1.625179
## - Fold2: alpha=0.46533, lambda=1.625179
## + Fold3: alpha=0.26189, lambda=0.382909
## - Fold3: alpha=0.26189, lambda=0.382909
## + Fold3: alpha=0.17301, lambda=0.539936
## - Fold3: alpha=0.17301, lambda=0.539936
## + Fold3: alpha=0.89094, lambda=0.018498
## - Fold3: alpha=0.89094, lambda=0.018498
## + Fold3: alpha=0.03789, lambda=0.005650
## - Fold3: alpha=0.03789, lambda=0.005650
## + Fold3: alpha=0.26030, lambda=0.003013
## - Fold3: alpha=0.26030, lambda=0.003013
## + Fold3: alpha=0.51916, lambda=0.008256
## - Fold3: alpha=0.51916, lambda=0.008256
## + Fold3: alpha=0.92198, lambda=0.005467
## - Fold3: alpha=0.92198, lambda=0.005467
## + Fold3: alpha=0.48300, lambda=0.002933
## - Fold3: alpha=0.48300, lambda=0.002933
## + Fold3: alpha=0.56814, lambda=0.014619
```

```

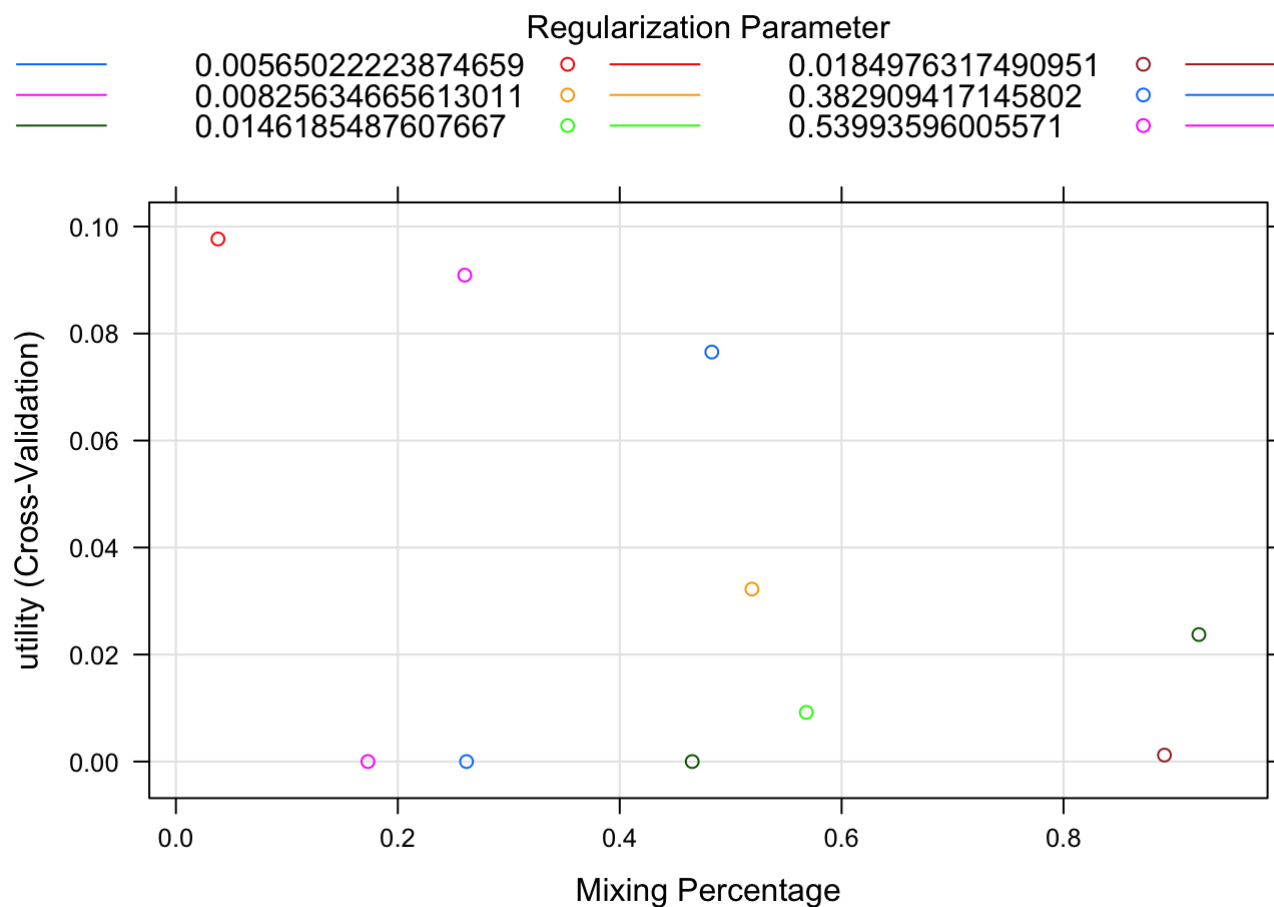
## - Fold3: alpha=0.56814, lambda=0.014619
## + Fold3: alpha=0.46533, lambda=1.625179
## - Fold3: alpha=0.46533, lambda=1.625179
## + Fold4: alpha=0.26189, lambda=0.382909
## - Fold4: alpha=0.26189, lambda=0.382909
## + Fold4: alpha=0.17301, lambda=0.539936
## - Fold4: alpha=0.17301, lambda=0.539936
## + Fold4: alpha=0.89094, lambda=0.018498
## - Fold4: alpha=0.89094, lambda=0.018498
## + Fold4: alpha=0.03789, lambda=0.005650
## - Fold4: alpha=0.03789, lambda=0.005650
## + Fold4: alpha=0.26030, lambda=0.003013
## - Fold4: alpha=0.26030, lambda=0.003013
## + Fold4: alpha=0.51916, lambda=0.008256
## - Fold4: alpha=0.51916, lambda=0.008256
## + Fold4: alpha=0.92198, lambda=0.005467
## - Fold4: alpha=0.92198, lambda=0.005467
## + Fold4: alpha=0.48300, lambda=0.002933
## - Fold4: alpha=0.48300, lambda=0.002933
## + Fold4: alpha=0.56814, lambda=0.014619
## - Fold4: alpha=0.56814, lambda=0.014619
## + Fold4: alpha=0.46533, lambda=1.625179
## - Fold4: alpha=0.46533, lambda=1.625179
## + Fold5: alpha=0.26189, lambda=0.382909
## - Fold5: alpha=0.26189, lambda=0.382909
## + Fold5: alpha=0.17301, lambda=0.539936
## - Fold5: alpha=0.17301, lambda=0.539936
## + Fold5: alpha=0.89094, lambda=0.018498
## - Fold5: alpha=0.89094, lambda=0.018498
## + Fold5: alpha=0.03789, lambda=0.005650
## - Fold5: alpha=0.03789, lambda=0.005650
## + Fold5: alpha=0.26030, lambda=0.003013
## - Fold5: alpha=0.26030, lambda=0.003013
## + Fold5: alpha=0.51916, lambda=0.008256
## - Fold5: alpha=0.51916, lambda=0.008256
## + Fold5: alpha=0.92198, lambda=0.005467
## - Fold5: alpha=0.92198, lambda=0.005467
## + Fold5: alpha=0.48300, lambda=0.002933
## - Fold5: alpha=0.48300, lambda=0.002933
## + Fold5: alpha=0.56814, lambda=0.014619
## - Fold5: alpha=0.56814, lambda=0.014619
## + Fold5: alpha=0.46533, lambda=1.625179
## - Fold5: alpha=0.46533, lambda=1.625179
## Aggregating results
## Selecting tuning parameters
## Fitting alpha = 0.0379, lambda = 0.00565 on full training set

```

```
print(df.glm)
```

```
## glmnet
##
## 42130 samples
##    55 predictor
##    2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 33704, 33705, 33703, 33704, 33704
## Resampling results across tuning parameters:
##
##  alpha          lambda          utility
##  0.03788688    0.005650222    0.097674298
##  0.17300505    0.539935960    0.000000000
##  0.26029747    0.003012596    0.090921560
##  0.26189424    0.382909417    0.000000000
##  0.46533173    1.625179138    0.000000000
##  0.48300361    0.002933316    0.076549519
##  0.51915940    0.008256347    0.032233617
##  0.56813514    0.014618549    0.009197706
##  0.89094150    0.018497632    0.001210539
##  0.92197954    0.005466809    0.023735850
##
## utility was used to select the optimal model using the largest value.
## The final values used for the model were alpha = 0.03788688 and lambda
## = 0.005650222.
```

```
plot(df.glm)
```



```
df.glm.model <- df.glm$finalModel

df.glm.model.coef <- predict(df.glm.model,
                             s = df.glm.model$lambdaOpt,
                             type = 'coefficients')

print(df.glm.model.coef)
```

```

## 84 x 1 sparse Matrix of class "dgCMatrix"
##                                     1
## (Intercept)                      -0.8829210838
## MonthlyRevenue                    0.0010826125
## MonthlyMinutes                    -0.0002257496
## TotalRecurringCharge              -0.0025611654
## DirectorAssistedCalls             -0.0007159445
## OverageMinutes                    0.0009649362
## RoamingCalls                      0.0033912257
## PercChangeMinutes                 -0.0005591307
## PercChangeRevenues                0.0029435017
## DroppedCalls                      0.0043611299
## BlockedCalls                      0.0003205355
## UnansweredCalls                   0.0007190014
## CustomerCareCalls                 -0.0074604498
## ThreewayCalls                     -0.0404098991
## ReceivedCalls                     0.0001398066
## OutboundCalls                     0.0005254314
## InboundCalls                      -0.0020231501
## PeakCallsInOut                    -0.0005009393
## OffPeakCallsInOut                 -0.0001306336
## DroppedBlockedCalls               0.0022226071
## CallForwardingCalls               .
## CallWaitingCalls                  -0.0014542356
## MonthsInService                   -0.0172326611
## UniqueSubs                        0.1412746540
## ActiveSubs                        -0.1330361683
## Handsets                          0.0407398497
## HandsetModels                     .
## CurrentEquipmentDays              0.0012538727
## AgeHH1                            -0.0039599640
## AgeHH2                            -0.0009520021
## ChildrenInHHYes                   0.1480784932
## HandsetRefurbishedYes              0.2681911387
## HandsetWebCapableYes              -0.1563485591
## TruckOwnerYes                     -0.0178485298
## RVOwnerYes                       -0.0013373587
## HomeownershipUnknown              0.0011110364
## BuysViaMailOrderYes               .
## RespondsToMailOffersYes           -0.1254472639
## OptOutMailingsYes                 0.0124384389
## NonUSTravelYes                    0.0242472342
## OwnsComputerYes                   0.0169603282
## HasCreditCardYes                  0.1049973805
## RetentionCalls                    0.2356429124
## RetentionOffersAccepted            -0.1150853300
## NewCellphoneUserYes               -0.0241923965
## NotNewCellphoneUserYes             0.0403813673
## ReferralsMadeBySubscriber          -0.0255081148
## IncomeGroup                       -0.0064153998
## OwnsMotorcycleYes                 0.1349784694
## AdjustmentsToCreditRating         -0.0789824124
## HandsetPrice100                    0.0768162668
## HandsetPrice130                    .
## HandsetPrice150                    0.0830590357
## HandsetPrice180                    -0.4809950609
## HandsetPrice200                    0.1283548104

```

```
## HandsetPrice240 -0.2971212060
## HandsetPrice250 0.7939818873
## HandsetPrice30 0.0322633196
## HandsetPrice300 -0.7050677632
## HandsetPrice40 0.1675022607
## HandsetPrice400 -0.1003287939
## HandsetPrice500 -0.3946057348
## HandsetPrice60 0.0666827123
## HandsetPrice80 0.0553621835
## HandsetPriceUnknown -0.0726065884
## MadeCallToRetentionTeamYes 0.4839465529
## CreditRating2-High 0.0706089182
## CreditRating3-Good 0.0644040350
## CreditRating4-Medium -0.1260415415
## CreditRating5-Low -0.3405897304
## CreditRating6-VeryLow -0.1452777077
## CreditRating7-Lowest -0.0358780585
## PrizmCodeRural 0.1228721286
## PrizmCodeSuburban -0.0329015433
## PrizmCodeTown 0.0465967328
## OccupationCrafts -0.0436313599
## OccupationHomemaker 0.1364964741
## OccupationOther -0.0155563981
## OccupationProfessional -0.0612521373
## OccupationRetired -0.1532845257
## OccupationSelf -0.0461373490
## OccupationStudent 0.1954486424
## MaritalStatusUnknown 0.1057728096
## MaritalStatusYes 0.0596588492
```

```
df.glm.pred <- predict(df.glm, newdata = df.part$dev)

print('Utilidad')
```

```
## [1] "Utilidad"
```

```
df.glm.utility <- fn_utility(df.glm.pred, df.part$dev$Churn)

print(df.glm.utility)
```

```
## [1] 0.1398206
```

```
print('Matriz de confusion')
```

```
## [1] "Matriz de confusion"
```

```
df.glm.cm <- conf_matrix(df.glm.pred, df.part$dev$Churn)

print(df.glm.cm)
```

```
##           Reference
## Prediction    No   Yes
##           No  3302 1306
##           Yes   34   39
```

```
print('** Generacion de la prediccion sobre test sample')
```

```
## [1] "** Generacion de la prediccion sobre test sample"
```

```
test_sample <- read.csv('../data/test_sample.csv')
rownames(test_sample) <- test_sample$CustomerID
test_sample$CustomerID <- NULL
test_sample$ServiceArea <- NULL

file_id <- paste0(c(script.name, script.date), collapse = ' ')

gen_prediction(df.glm, test_sample, id = file_id)

print('Done')
```

```
## [1] "Done"
```

```
script.done <- Sys.time()

print(script.done - script.start)
```

```
## Time difference of 3.220879 mins
```