script_glm_prob.R

sergio

2020-05-27

```
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_prob'

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"
```

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

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

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

df.fn_summary <- fn_summaryUtility

df.metric <- 'utility'

df.form <- Churn ~ .

print('** GLM')
```

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

```r
df.glm.ctrl <- trainControl(method = 'none',
                            verboseIter = TRUE,
                            classProbs = 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 = 1,
                metric = df.metric)
```

```
## Fitting alpha = 0.262, lambda = 0.00464 on full training set
```

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

```
## glmnet
##
## 42130 samples
##    55 predictor
##     2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: None
```

```r
df.glm.prob <- predict(df.glm, newdata = df.part$dev, type = 'prob')

df.thr_vec <- seq(0.1, 0.9, 0.025)
df.thr_utility <- rep(0, length(df.thr_vec))
for(i in 1:length(df.thr_vec)) {
    pred <- fn_pred(df.glm.prob, thr = df.thr_vec[i])
    df.thr_utility[i] <- fn_utility(yhat = pred, y = df.part$dev$Churn)
}

print('Utilidad por umbral')
```

```
## [1] "Utilidad por umbral"
```

```r
print(df.thr_utility)
```

```
##  [1]  0.203161718  0.219611194  0.279320658  0.478316599  0.726874599
##  [6]  0.942106388  1.019013031  1.183721427  1.206686605  1.025635548
## [11]  0.680623798  0.451292459  0.285836360  0.266075625  0.153813288
## [16]  0.125080111  0.107882931  0.066011536  0.039735099  0.026490066
## [21]  0.026276437  0.017517624  0.006622517  0.001174963  0.001174963
## [26] -0.004272591 -0.004272591 -0.004272591 -0.004272591 -0.004272591
## [31] -0.004272591  0.000000000  0.000000000
```

```r
plot_thr_utility(df.thr_utility, df.thr_vec, 'glmnet')
```