script\_rpart\_thr\_grid.R

Poitou

2020-07-09

library(rpart)  
library(rpart.plot)  
  
#ojo va el utils modificado  
source('../utils/utils\_oblig.R')

## Loading required package: caret

## Loading required package: lattice

## Loading required package: ggplot2

set.seed(117)  
  
script.name <- 'rpart\_thr'  
  
script.date <- 'el\_mejor'  
  
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 = 0.8)  
  
df.thr\_vec <- seq(0.1, 0.3, 0.025)  
  
df.fn\_summary <- function(data, lev = NULL, model = NULL) {  
 fn\_summaryUtilityThr(data, df.thr\_vec)  
}  
  
df.metric <- 'utility'  
  
df.form <- Churn ~ .  
  
print('\*\* RPART')

## [1] "\*\* RPART"

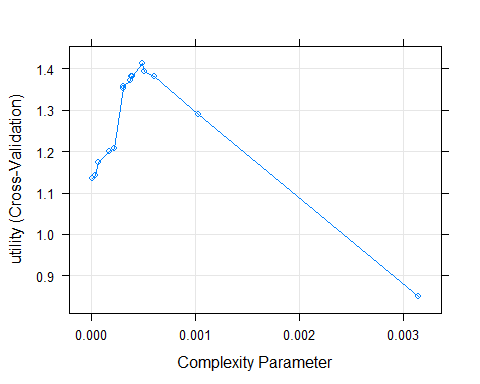
df.rpart.ctrl <- trainControl(method = 'cv',  
 number = 5,  
 verboseIter = TRUE,  
 classProbs = TRUE,  
 search = 'random',  
 summaryFunction = df.fn\_summary)  
  
#df.rpart.grid <- expand.grid(cp = seq(0.0001, 0.001, 0.0001))  
  
df.rpart <- train(form = df.form,   
 data = df.part$train,   
 method = 'rpart',   
 trControl = df.rpart.ctrl,  
 #tuneGrid = df.rpart.grid,  
 tuneLength = 20,  
 metric = df.metric)

## + Fold1: cp=1.036e-05   
## - Fold1: cp=1.036e-05   
## + Fold2: cp=1.036e-05   
## - Fold2: cp=1.036e-05   
## + Fold3: cp=1.036e-05   
## - Fold3: cp=1.036e-05   
## + Fold4: cp=1.036e-05   
## - Fold4: cp=1.036e-05   
## + Fold5: cp=1.036e-05   
## - Fold5: cp=1.036e-05   
## Aggregating results  
## Selecting tuning parameters  
## Fitting cp = 0.000489 on full training set

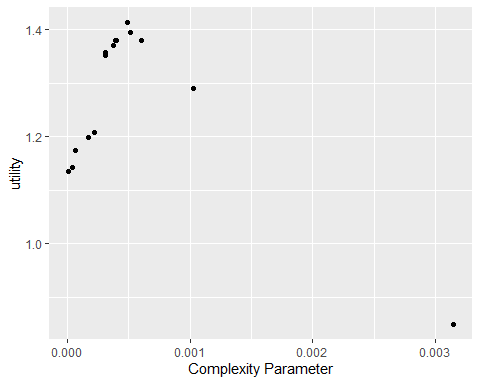
print(df.rpart)

## CART   
##   
## 37449 samples  
## 55 predictor  
## 2 classes: 'No', 'Yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 29960, 29960, 29959, 29958, 29959   
## Resampling results across tuning parameters:  
##   
## cp utility prob\_thr  
## 1.035904e-05 1.1358308 0.160   
## 3.729256e-05 1.1433482 0.165   
## 6.659386e-05 1.1747221 0.165   
## 1.709242e-04 1.1994752 0.170   
## 2.175399e-04 1.2075789 0.180   
## 3.063317e-04 1.3526427 0.210   
## 3.107713e-04 1.3582372 0.220   
## 3.729256e-04 1.3717618 0.205   
## 3.884642e-04 1.3807880 0.205   
## 3.962335e-04 1.3810951 0.205   
## 4.894649e-04 1.4140863 0.230   
## 5.127727e-04 1.3948596 0.230   
## 6.060041e-04 1.3809188 0.250   
## 1.025545e-03 1.2907187 0.225   
## 3.146560e-03 0.8489949 0.175   
##   
## utility was used to select the optimal model using the largest value.  
## The final value used for the model was cp = 0.0004894649.

df.rpart.results <- fn\_results(df.rpart)  
  
plot(df.rpart)

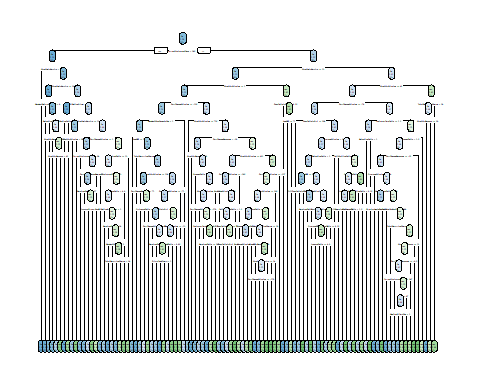


ggplot(df.rpart)



df.rpart.model <- df.rpart$finalModel  
   
rpart.plot(df.rpart.model)

## Warning: labs do not fit even at cex 0.15, there may be some overplotting



print('Umbral')

## [1] "Umbral"

print(df.rpart.results$prob\_thr)

## [1] 0.23

print('Utilidad en cv')

## [1] "Utilidad en cv"

print(df.rpart.results$utility)

## [1] 1.414086

print('Utilidad en dev')

## [1] "Utilidad en dev"

df.rpart.dev.prob <- predict(df.rpart, newdata = df.part$dev, type = 'prob')  
df.rpart.dev.pred <- fn\_pred(df.rpart.dev.prob, thr = df.rpart.results$prob\_thr)  
  
df.rpart.dev.utility <- fn\_utility(df.rpart.dev.pred, df.part$dev$Churn)  
  
print(df.rpart.dev.utility)

## [1] 1.493965

print('Utilidad en train')

## [1] "Utilidad en train"

df.rpart.train.prob <- predict(df.rpart, newdata = df.part$train, type = 'prob')  
df.rpart.train.pred <- fn\_pred(df.rpart.train.prob, thr = df.rpart.results$prob\_thr)  
  
df.rpart.train.utility <- fn\_utility(df.rpart.train.pred, df.part$train$Churn)  
  
print(df.rpart.train.utility)

## [1] 1.552244

#################  
# ploteo de las utilidades de CV y train vs el umbral  
  
df.rpart.train.prob <- predict(df.rpart, newdata = df.part$train, type = 'prob')  
  
df.rpart.dev.utility\_vec <- fn\_utility\_thr(y\_prob = df.rpart.dev.prob,   
 y = df.part$dev$Churn,   
 thr\_vec = df.thr\_vec)  
  
  
df.rpart.train.utility\_vec <- fn\_utility\_thr(y\_prob = df.rpart.train.prob,   
 y = df.part$train$Churn,   
 thr\_vec = df.thr\_vec)  
  
print('Utilidad por umbral')

## [1] "Utilidad por umbral"

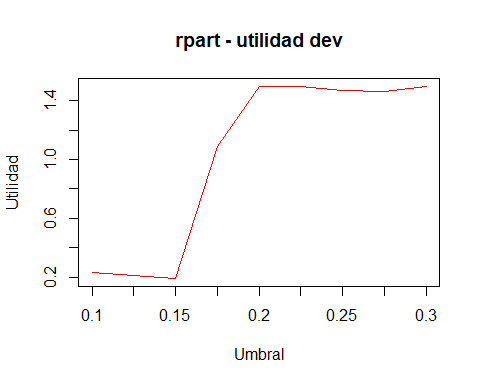
print(df.rpart.dev.utility\_vec)

## utility1 utility2 utility3 utility4 utility5 utility6 utility7 utility8   
## 0.2294382 0.2135762 0.1922666 1.0914335 1.4969024 1.4939650 1.4689703 1.4596774   
## utility9   
## 1.4957808

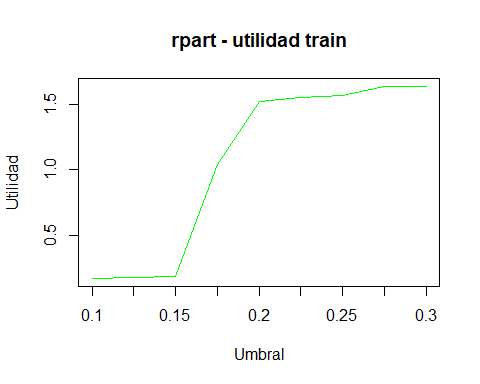
print(df.rpart.train.utility\_vec)

## utility1 utility2 utility3 utility4 utility5 utility6 utility7 utility8   
## 0.1746909 0.1826217 0.1896045 1.0414163 1.5172768 1.5522444 1.5653556 1.6334081   
## utility9   
## 1.6346364

#par(mfrow=c(1,1))  
plot\_thr\_utility(df.rpart.dev.utility\_vec, df.thr\_vec, 'rpart - utilidad dev')



#abline(h = df.rpart.results$prob\_thr)  
#par(new=TRUE)  
  
plot\_thr\_utility\_train(df.rpart.train.utility\_vec, df.thr\_vec, 'rpart - utilidad train')



#abline(v = df.rpart.results$prob\_thr, col="blue")  
#########################################  
  
print('Matriz de confusion en dev')

## [1] "Matriz de confusion en dev"

df.rpart.dev.cm <- conf\_matrix(df.rpart.dev.pred, df.part$dev$Churn)  
  
print(df.rpart.dev.cm)

## Reference  
## Prediction No Yes  
## No 2749 617  
## Yes 3913 2083

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.rpart, test\_sample, prob\_thr = df.rpart.results$prob\_thr, id = file\_id)  
  
print('Done')

## [1] "Done"

script.done <- Sys.time()  
  
print(script.done - script.start)

## Time difference of 1.260617 mins