# Entrega Final

rm(list=ls())

setwd("C:/Users/USER/OneDrive - Universidad de los Andes/Escritorio/BigData/PS1/Entrega-Final/stores/")

library(tidyverse)

library(table1)

births <- read\_csv("microdata\_nacimientos.csv")

# Limpieza de variables:

# Bajo peso al nacer var dependiente

births <- births %>% mutate(PESO\_NAC = ifelse(PESO\_NAC == 9, NA, PESO\_NAC))

births <- births %>% mutate(bajo\_peso = ifelse(PESO\_NAC == 1 |

PESO\_NAC == 2 |

PESO\_NAC == 3 |

PESO\_NAC == 4,'SI','NO'))

table(is.na(births$IDPERTET)) #12,6K NAS

# Nacio en hospital

births <- births %>% mutate(SIT\_PARTO = ifelse(SIT\_PARTO == 9, NA, SIT\_PARTO))

births <- births %>% mutate(hospital = ifelse(SIT\_PARTO == 1,1,0))

# Prematuro

births <- births %>% mutate(T\_GES = ifelse(T\_GES == 9 | T\_GES == 6, NA, T\_GES))

births <- births %>% mutate(prematuro = ifelse(T\_GES == 1 |

T\_GES == 2 |

T\_GES == 3,1,0))

# Num consultas prenatales

births <- births %>% mutate(n\_consultas = ifelse(NUMCONSUL == 99, NA, NUMCONSUL))

# APGAR1 APGAR2

births <- births %>% mutate(APGAR1 = ifelse(APGAR1 == 99, NA, APGAR1))

births <- births %>% mutate(APGAR2 = ifelse(APGAR2 == 99, NA, APGAR2))

# Identidad etnica (factor)

births <- births %>% mutate(id\_etnica = ifelse(IDPERTET == 9, NA, IDPERTET))

#births$id\_etnica <- as.factor(births$IDPERTET)

# Edad Madre (factor)

births <- births %>% mutate(EDAD\_MADRE = ifelse(EDAD\_MADRE == 99, NA, EDAD\_MADRE))

# Madre casada

births <- births %>% mutate(EST\_CIVM = ifelse(EST\_CIVM == 9, NA, EST\_CIVM))

births <- births %>% mutate(m\_casada = ifelse(EST\_CIVM == 6, 1, 0))

# Ultimo anio escolar alcanzado por la madre y padre

births <- births %>% mutate(ULTCURMAD = ifelse(ULTCURMAD == 99, NA, ULTCURMAD))

p1 = "[:digit:]+"

births <- births %>%

mutate(m\_anios\_colegio = str\_extract(string = births$ULTCURMAD,pattern = p1))

births$m\_anios\_colegio <- as.numeric(births$m\_anios\_colegio)

births <- births %>% mutate(ULTCURPAD = ifelse(ULTCURPAD == 99, NA, ULTCURPAD))

births <- births %>%

mutate(p\_anios\_colegio = str\_extract(string = births$ULTCURPAD,pattern = p1))

births$p\_anios\_colegio <- as.numeric(births$p\_anios\_colegio)

# Num hijos

births <- births %>%

mutate(n\_hijos = str\_extract(string = births$N\_HIJOSV,pattern = p1))

births$n\_hijos <- as.numeric(births$n\_hijos)

# Nun embarazos

births <- births %>% mutate(n\_emb = N\_EMB)

# La madre esta en el regimen contributivo

births <- births %>% mutate(m\_reg\_contribu = ifelse(SEG\_SOCIAL == 1, 1, 0))

# crear identificados

births <- births %>% select(bajo\_peso,

hospital,

prematuro,

id\_etnica,

m\_casada,

m\_anios\_colegio,

p\_anios\_colegio,

n\_hijos,

n\_emb,

m\_reg\_contribu)

# Eliminar NAs en varianle dependiente

births <- births %>% drop\_na()

# Crear identificador

births <- births %>% mutate(id = 1:1119514)

# Estadisticas descriptivas

descriptivas <- births

y <- c("bajo\_peso", "hospital", "prematuro", "id\_etnica", "m\_casada", "m\_reg\_contribu")

descriptivas[y] <- lapply(descriptivas[y], factor)

descriptiva\_1 <- table1(~ bajo\_peso+hospital+prematuro+id\_etnica+m\_casada+

m\_anios\_colegio+p\_anios\_colegio+n\_hijos+

n\_emb +m\_reg\_contribu,

data=descriptivas, overall="Total")

descriptiva\_1

# Separar Test y Train set: 80/20 (895,611 / 221,902)

set.seed(10101)

train\_sample <- sample(1119514, 895611)

train\_set <- births[train\_sample, ]

test\_set <- births[-train\_sample, ]

#test\_set <- test\_set %>% select(-bajo\_peso)

# SMOTE como técnica de balanceo

library("ROSE")

set.seed(10101)

smote <- ovun.sample(bajo\_peso ~ ., train\_set, method='both')

train\_set = smote$data

prop.table(table(train\_set$bajo\_peso)) #49% no, 50% si

# XG BOOST

library(xgboost)

library(e1071)

grid\_default <- expand.grid(nrounds = c(50,100),

max\_depth = c(4,6,8),

eta = c(0.01,0.3),

gamma = c(0,1),

min\_child\_weight = c(10, 25,50),

colsample\_bytree = c(0.7),

subsample = c(0.6))

set.seed(10101)

ctrl <- trainControl(method = "cv",

number = 5,

summaryFunction = twoClassSummary,

classProbs = TRUE,

verbose=FALSE,

savePredictions = T)

xgboost <- train(

as.factor(bajo\_peso) ~.,

data = train\_set,

method = "xgbTree",

trControl = ctrl,

metric = "Sens",

tuneGrid = grid\_default,

preProcess = c("center", "scale")

)

xgboost

# Importancia de variables

var\_Imp <- varImp(xgboost, scale = FALSE)

var\_Imp

# El modelo que optimiza Sens es:

#nrounds = 50,

#max\_depth = 6,

#eta = 0.01,

#gamma = 1,

#min\_child\_weight = 50,

#colsample\_bytree = 0.7,

#subsample = 0.6

pred\_xgb <- predict(xgboost,test\_set)

confusionMatrix(as.factor(test\_set$bajo\_peso),pred\_xgb)