#1 --- cargando librer?as #-----

library(foreign)

library(dplyr)

library(caret)

library(e1071) #maquinas de soporte svm

library(rpart) #CART

library(rpart.plot) # grafico

library(party) # No param?trico

library(partykit) #toolkit para pasar de party a partykit

library(ROCR) #validaci?n

###################################### ######################################

##################################### ######################################

##################################### SECCION A ######################################

##################################### ######################################

###2 iniciando el proceso de modelamiento ###

datos <- read.spss("C:\\Users\\lprodriguez\\Desktop\\EXPERTO EN CIENCIA DE ATOS\\MODULO 6\\machine learning\\ENCUESTA NACIDOS VIVOS\\ENV\_2017.sav"

,

use.value.labels = F,

to.data.frame = T)

# depurar la informaci?n

table(datos$prov\_nac)

str(datos$prov\_nac)

##CONVERTIR A NUMERICO PROV NACIMIENTO

datos$prov\_nac <- as.numeric(as.character(datos$prov\_nac))

### FILTRAR EL CODIGO 13 O MANABI Y LIMPIAR DATA NO FUNCIONAL CON VALORES ATIPICOS O SIN INFORMACION

nuevadata <- datos %>% filter(prov\_nac==13)%>%

select(peso,

talla,

sem\_gest,

sexo,

edad\_mad,

sabe\_leer,

con\_pren)%>%

filter(

peso!=99,

talla!=99,

sem\_gest!=99,

con\_pren!=99,

sabe\_leer!=9)%>%

mutate(peso=if\_else(peso>2500,1,0),

sexo=if\_else(sexo==1,0,1),

sabe\_leer=if\_else(sabe\_leer==1,1,0),

con\_pren=if\_else(con\_pren>=7,1,0),

edad2=edad\_mad^2)

###VER DATA DEPURADA

nuevadata

##VER SI YA ES FACTOR LA VAR DE INTERES

str(nuevadata$peso)

##CONVERTIR A FACTOR VARIABLE PESO QUE ES DE INTERES

nuevadata$peso <- factor(nuevadata$peso)

##CONVERTIR A NIVELES LA VAR DE INTERES EN VEZ DE BINARIOS 1,0

nuevadata <- nuevadata %>%

mutate(peso=recode\_factor(

peso,

`0`="no.adecuado",

`1`="adecuado"))

## OBSERVAR LOS RESULTADOS DE LOS NUEVOS NIVELES

table(nuevadata$peso)

"no.adecuado adecuado

2299 21676 "

# fijar una semilla

set.seed(1234)

#crear una muestra de entrenamiento

#10% Entrenamiento normalmente

#90% validaci?n

## solo 30% por capacidad computacional TENGO UNA CORE I 5 16 DE RAM POR ESO LE SUBI UN POCO

entrenamiento <- createDataPartition(nuevadata$peso,

p=0.10,list=F)

#realizamos el modelo SVM con la muestra de entrenamiento

modelo <- svm(peso ~talla+sem\_gest+sexo+

edad\_mad+edad2+sabe\_leer,

data=nuevadata[entrenamiento,],

kernel="linear",

cost=10,scale=T,probability=TRUE)

summary(modelo)

# VALIDACION CRUZADA CON LA FUNCION TUNE

#----- BUSCAR EL MEJOR COSTO PAL KERNEL

## ESTO CAMBIA GRAFICA Y MATRIZ DE CONFUSION Y ACCURACY

#, VERA POR VALIDACION CRUZADA QUE COSTO DE LOS DESCRITOS ES EL MEJOR

modelo.tuneado <- tune(svm,

peso ~.,

data=nuevadata[entrenamiento,],

ranges = list(cost=c(0.001,0.01,0.1,1,5,10,50,75,85,95,100)),

kernel="linear",

scale=T,

probability=TRUE)

summary(modelo.tuneado)

### EL MEJOR MODELO ES EL QUE INCLUYE EL COSTO 0.10 A CONTINUACION LA PRUEBA:

"best parameters:

cost

0.1

- best performance: 0.08757322

- Detailed performance results:

cost error dispersion

1 0.001 0.09591353 0.010018462

2 0.010 0.08840656 0.008282115

3 0.100 0.08757322 0.008102012"

#### GRAFICO MEJOR COSTO

ggplot(data=modelo.tuneado$performances,

aes(x=cost,y=error))+

geom\_line()+

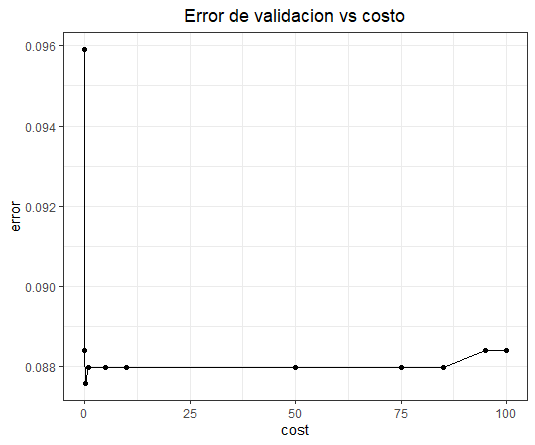
geom\_point()+

labs(title="Error de validacion vs costo")+

theme\_bw()+

theme(plot.title = element\_text(hjust = 0.5))

#A CONTINUACION LA GRAFICA

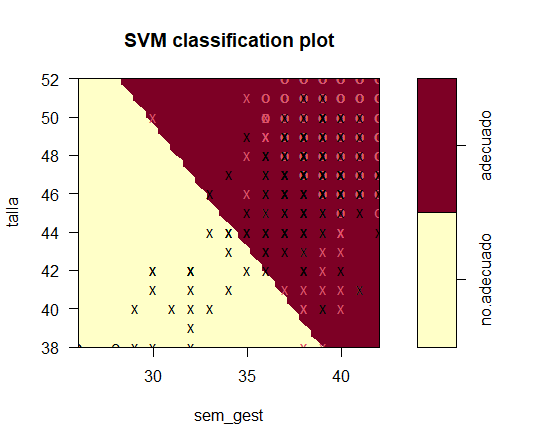


######## GRAFICA MEJOR MODELO

plot(mejor.modelo,

data=nuevadata[entrenamiento,],

talla~sem\_gest)



###################################### ######################################

##################################### ######################################

##################################### SECCION B ######################################

##################################### ######################################

#----- validando el mejor modelo #------

ajustados.mejor.modelo <- predict(mejor.modelo,

nuevadata[entrenamiento,],

type="prob",

probability = T)

#verificar como captura las probabilidades

str(ajustados.mejor.modelo)

head(attr(ajustados.mejor.modelo,"probabilities"),5)

"adecuado no.adecuado

24 0.9410649 0.05893512

37 0.9180822 0.08191778

40 0.8120200 0.18797999

43 0.8633588 0.13664117

55 0.9877387 0.01226128"

#matriz de confusi?n o clasificaci?n

confusionMatrix(ajustados.mejor.modelo,

nuevadata$peso[entrenamiento],

positive = levels(nuevadata$peso)[2])

"Reference

Prediction no.adecuado adecuado

no.adecuado 47 15

adecuado 183 2153

Accuracy : 0.9174

95% CI : (0.9057, 0.9281)

No Information Rate : 0.9041

P-Value [Acc > NIR] : 0.01313

Kappa : 0.2931

Mcnemar's Test P-Value : < 2e-16

Sensitivity : 0.9931

Specificity : 0.2043

Pos Pred Value : 0.9217

Neg Pred Value : 0.7581

Prevalence : 0.9041

Detection Rate : 0.8978

Detection Prevalence : 0.9741

Balanced Accuracy : 0.5987 "

## en este proceso sale todo mas facil

#ANALIZAR LAS CURAS ROC

pred <- prediction(attr(ajustados.mejor.modelo,

"probabilities")[,2],

nuevadata$peso[entrenamiento])

perf <- performance(pred,"tpr","fpr")

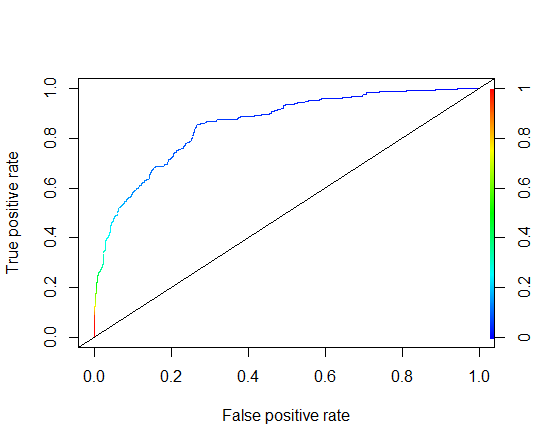
plot(perf,colorize=T,lty=3)

abline(0,1,col="black")

#area bajo la curva, esta alejada de la linea tangente 45 grados, es un buen modelo, mejor

## que el azar en definitiva. Sin embargo la curvatura no muestra

## A CONTINUACION LA CURVA



###AUC

aucmodelo1 <- performance(pred,measure = "auc")

aucmodelo1 <- aucmodelo1@y.values[[1]]

aucmodelo1

"aucmodelo1

[1] 0.8548, dio esto cercano a uno o area bajo la curva

puede concebirse como un r cuadrado, mejor se clasifica en el

modelo los 1 y los ceros"

#PUNTO DE CORTE OPTIMO

#determinar el cut -off que maximiza

#el accuracy de mi modelo

max.accuracy <- performance(pred,measure = "acc")

plot(max.accuracy)

indice <- which.max(slot(max.accuracy,"y.values")[[1]])

acc <- slot(max.accuracy,"y.values")[[1]][indice]

cutoff <- slot(max.accuracy,"x.values")[[1]][indice]

##impresion el punto de corte

print(c(accuracy=acc,

cutoff=cutoff))

## EL RESULTADO ES accuracy cutoff.10430

# MEJOR SCORE 0.9195163 MEJOR UMBRAL 0.4589198

#--- Evaluando punto de corte sugerido #-----

#definamosel punto de corte

umbral <- as.numeric(cutoff)

##cutoff es el punto de corte que calcule antes que maximiza el accuracy

table(attr(ajustados.mejor.modelo,

"probabilities")[,1]>umbral,

nuevadata$peso[entrenamiento])

#echar un vistazo de las probabilidades devueltas

head(attr(ajustados.mejor.modelo,

"probabilities"))

#seleccionamos la probabilidad objetivo

prediccionescutoff <- attr(ajustados.mejor.modelo,

"probabilities")[,1]

str(prediccionescutoff)

prediccionescutoff <- as.numeric(prediccionescutoff)

predCut <- factor(ifelse(prediccionescutoff>umbral,1,0))

matrizpuntocorte <- data.frame(real=nuevadata$peso[entrenamiento],

predicho=predCut)

matrizpuntocorte <- matrizpuntocorte %>% mutate(predicho=recode\_factor(predicho,

`0`="no.adecuado",

`1`="adecuado"))

###MATRIZ DE CONFUSION

confusionMatrix(matrizpuntocorte$predicho,

matrizpuntocorte$real,

positive = "adecuado")

##resultado

" Reference

Prediction no.adecuado adecuado

no.adecuado 44 14

adecuado 186 2154

Accuracy : 0.9166

95% CI : (0.9048, 0.9274)

No Information Rate : 0.9041

P-Value [Acc > NIR] : 0.01887

Kappa : 0.2776

Mcnemar's Test P-Value : < 2e-16

Sensitivity : 0.9935

Specificity : 0.1913

Pos Pred Value : 0.9205

Neg Pred Value : 0.7586

Prevalence : 0.9041

Detection Rate : 0.8982

Detection Prevalence : 0.9758

Balanced Accuracy : 0.5924"

#especificidad muy baja podria requerir un remuestreo

# ---- prediciendo con SVM #----

########### 5 predicciones

newdata <- head(nuevadata,5)

str(newdata)

#predecir dentro de la muestra

#considerar que para la predicci?n

### El punto de corte por defecto es de 0.5

predict(mejor.modelo,newdata)

pronostico1 <- predict(mejor.modelo,newdata)

p.probabilidades <- predict(mejor.modelo,

newdata,

probability = TRUE)

p.probabilidades

#resultdo

#1 2 3 4 5

#adecuado adecuado adecuado adecuado adecuado

"adecuado no.adecuado

1 0.9892709 0.01072911

2 0.9896688 0.01033123

3 0.9754416 0.02455837

4 0.9686308 0.03136924

5 0.9645059 0.03549407"

#PRONOSTICO FUERA DE LA MUESTRA

names(newdata)

newdata2 <- data.frame(talla=40,

sem\_gest=40,

sexo=1,

edad\_mad=23,

sabe\_leer=0,

con\_pren=0,

edad2=860)

pronostico2 <- predict(mejor.modelo,newdata2, probability = T)

pronostico2

predict(mejor.modelo,newdata2)

#resultado

"1

adecuado

Levels: no.adecuado adecuado"

###################################### ######################################

##################################### ######################################

##################################### SECCION C ######################################

##################################### ######################################

library(ROSE)

table(nuevadata$peso[entrenamiento])

##ver si son dispares y requiere remuestreo

"no.adecuado adecuado

230 2168 "

"hay que equiparar las categorias"

#traemos del arreglo de datos entrenamiento

#y lo convertimos en dataframe

train\_data <- nuevadata[entrenamiento, ]

#ROSE: metodo sintetico

roses <- ROSE(peso ~.,

data = train\_data,seed = 1)$data

table(roses$peso)

"adecuado no.adecuado

1249 1149

ES EL METODO MAS ROBUSTO HA EQUIPARADO LA MUESTRA EN LOS DOS NIVELES"

###ROSE

modelo.rose <- tune(svm, peso ~ .,

data=roses,

ranges = list(cost = c(0.001, 0.01, 0.1, 1, 5, 10, 50)),

kernel = "linear",

scale=T,

probability = TRUE)

summary(modelo.rose)

"best parameters:

cost

1

- best performance: 0.2585146

- Detailed performance results:

cost error dispersion

1 1e-03 0.2939679 0.02422588

2 1e-02 0.2601813 0.03097851

3 1e-01 0.2597646 0.03266779

4 1e+00 0.2585146 0.03279042

5 5e+00 0.2585146 0.03225661

6 1e+01 0.2585146 0.03225661

7 5e+01 0.2585146 0.03225661"

## mejor costo es 1

mejor.modelo.rose <- modelo.rose$best.model

mejor.modelo.rose

"Parameters:

SVM-Type: C-classification

SVM-Kernel: linear

cost: 1

Number of Support Vectors: 1451"

#----- EVALUACION DEL MODELO #------

# me interesa saber c?mo ha clasificado mi modelo

# necesito los valores predichos y los valores

######### ROSE

ajustadosrose <- predict(mejor.modelo.rose,

roses, type="prob",probability=TRUE)

# MATRIZ DE CONFUSION

confusionMatrix(roses$peso,ajustadosrose,

dnn = c("Actuales", "Predichos"),

levels(ajustadosrose)[1])

"Predichos

Actuales adecuado no.adecuado

adecuado 977 272

no.adecuado 352 797

Accuracy : 0.7398

95% CI : (0.7217, 0.7573)

No Information Rate : 0.5542

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.4772

Mcnemar's Test P-Value : 0.001564

Sensitivity : 0.7351

Specificity : 0.7456

Pos Pred Value : 0.7822

Neg Pred Value : 0.6936

Prevalence : 0.5542

Detection Rate : 0.4074

Detection Prevalence : 0.5209

Balanced Accuracy : 0.7403

'Positive' Class : adecuado"

## MUY BAJO EL ACCURACY DESPUES DE ESTAR BALANCEADO

#s?lo para comparar con el modelo sin remuestreo

confusionMatrix(ajustados.mejor.modelo,

nuevadata$peso[entrenamiento],

positive = levels(nuevadata$peso)[2])

#RESULTADO

" Reference

Prediction no.adecuado adecuado

no.adecuado 47 15

adecuado 183 2153

Accuracy : 0.9174

95% CI : (0.9057, 0.9281)

No Information Rate : 0.9041

P-Value [Acc > NIR] : 0.01313

Kappa : 0.2931

Mcnemar's Test P-Value : < 2e-16

Sensitivity : 0.9931

Specificity : 0.2043

Pos Pred Value : 0.9217

Neg Pred Value : 0.7581

Prevalence : 0.9041

Detection Rate : 0.8978

Detection Prevalence : 0.9741

Balanced Accuracy : 0.5987 "

##ROC

predrose <- prediction(attr(ajustadosrose,

"probabilities")[,2],

roses$peso)

roc.curve(roses$peso, attr(ajustadosrose,

"probabilities")[,2], col="red",

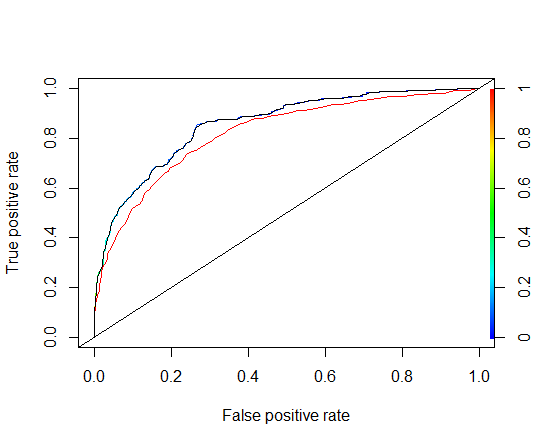
add.roc = T)

#Modelo original

roc.curve(nuevadata$peso[entrenamiento], attr(ajustados.mejor.modelo,

"probabilities")[,2], col="black",

add.roc = T)



"TOCA ELEGIR EL MODELO BALANCEADO, DADO QUE EL DESBALANCEADO MUESTRA RESULTADOS CON OVERFITTING

Y ESO HACE DESCONFIAR DEL MODELO ORIGINAL POR LO QUE SE DEBERA TOMAR LOS RESULTADOS

DEL MODELO BALANCEADO CON ROSE AUNQUE EL RESULTADO NO ES SATISFACTORIO.

La desicion de elegir el modelo balanceado a pesar de un menor accuracy se justifica por la necesidad de

evitar el overfitting y asegurar que el modelo clasifique de manera equitativa ambas clases.

Un modelo con alta sensibilidad pero baja especificidad, como en el caso del modelo sin remuestreo,

puede no ser práctico si clasifica incorrectamente una gran cantidad de casos negativos como positivos.

Importancia del Equilibrio: La elección del modelo debe considerar el equilibrio entre sensibilidad y especificidad,

así como el contexto práctico de aplicación del modelo. Un modelo con un alto valor predictivo positivo

pero bajo valor predictivo negativo (o viceversa) puede no ser útil en ciertos contextos."