Rmarkdown html test

Gabriel General

31-01-2021

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

library(kernlab) data(spam) str(spam[,1:5])

#generando subset de prueba set.seed(3435)

Distribución de variables en valores booleanos como trainIndicator

trainIndicator <- rbinom(4601, size = 1, prob = 0.5) table(trainIndicator)

#Se separan el dataset en Test y Training dataset mediante # distribución probabilística rbinom trainSpam = spam[trainIndicator == 1,] testSpam = spam[trainIndicator == 0,]

names(trainSpam) head(trainSpam) table(trainSpam$type)

#Se grafica incidencia entre correos spam que contienen mayor promedio de # letras mayúsculas en su contenido plot(trainSpam*capitalAve trainSpam*type)

#en logaritmo base 10 para mejor visualización. plot(log10(trainSpam*capitalAve* + 1) *trainSpam*type)

#Se eliminan los valores en cero para visualización plot(log10(trainSpam[,1:4] + 1))

#Cluster que identifica las variables con mayor incidencia en agrupación hCluster <- hclust(dist(t(trainSpam[,1:57])))

#Gráfico de dendograma de cluster plot(hCluster)

Log Base 10

hClusterUpdated <- hclust(dist(t(log10(trainSpam[,1:55] + 1)))) #dendograma plot(hClusterUpdated)

STATISTICAL PREDICTION MODELLING trainSpam*numtype* = *as.numeric*(*trainSpam*type) -1 costFunction = function(x,y) sum(x != (y > 0.5)) cvError = rep(NA,55) library(boot) for (i in 1:55){ lmFormula = reformulate(names(trainSpam)[i],response = "numtype") glmFit = glm(lmFormula, family = "binomial", data = trainSpam) cvError[i] = cv.glm(trainSpam, glmFit,costFunction,2)$delta[2] }

Cual predictor tiene un menor error de validación cruzada?

names(trainSpam)[which.min(cvError)]

#Modelo de regresión logística predictionModel = glm(numtype ~ charDollar, family = “binomial”, data = trainSpam)

hacer predicciones sobre el set de prueba

predictionTest = predict(predictionModel, testSpam) predictedSpam = rep(“nonspam”, dim(testSpam)[1])

#Clasificar como spam aquellos con una probabilidad mayor a 0.5 predictedSpam[predictionModel$fitted > 0.5] = “spam”

#Obtener una medida de incertidumbre table(predictedSpam, testSpam$type)

#tasa de error (61 + 458)/(1346 + 458 + 61 + 449)