



OPENCOURSEWARE

APRENDIZAJE AUTOMÁTICO PARA EL ANÁLISIS DE DATOS

GRADO EN ESTADÍSTICA Y EMPRESA

Ricardo Aler

Ejemplo de uso directo de librerías de R para árboles y KNN

Con C5.0

```
library(C50)
set.seed(0)
indices_train <- sample(1:nrow(BC), nrow(BC)*2/3, replace=FALSE)
trainX <- BC[indices_train, -col_clase]
trainy <- BC[indices_train, col_clase]

testX <- BC[-indices_train, -col_clase]
testy <- BC[-indices_train, col_clase]

set.seed(0) # Para fijar la semilla aleatoria, en caso de que el algoritmo tenga elementos aleatorios
model <- C5.0(x=trainX, y=trainy)

predsTest <- predict(model, testX)

errorTest <- mean(predsTest != testy)
```

Con knn

```
library(class)
set.seed(0)
indices_train <- sample(1:nrow(BC), nrow(BC)*2/3, replace=FALSE)
trainX <- BC[indices_train, -col_clase]
trainy <- BC[indices_train, col_clase]

testX <- BC[-indices_train, -col_clase]
testy <- BC[-indices_train, col_clase]

set.seed(0) # Para fijar la semilla aleatoria, en caso de que el algoritmo tenga elementos aleatorios
predTest <- knn(train=trainX, test=testX, cl=trainy, k=k)

errorTest <- mean(predsTest != testy)
```

Si en lugar de usar knn, usaramos knn.reg (del paquete FNN), las predicciones están en otro sitio:

```
predTest <- knn(train=trainX, test=testX, cl=trainy, k=k)
errorTest <- mean(predsTest != testy)
```

```
predTest <- knn.reg(train=trainX, test=testX, y=trainy, k=k)
errorTest <- mae(predTest$pred, testy)
```

Con rpart

```
library(rpart)
set.seed(0) # Para fijar la semilla aleatoria
# Separamos en train (2/3) y test (1/3)
indices_train <- sample(1:nrow(BH), nrow(BH)*2/3, replace=FALSE)
# Nótese que para rpart, no tenemos que separar X (entradas) de y (salida)
train <- BH[indices_train, ]
test <- BH[-indices_train, ]

model <- rpart(medv ~ ., data=train)

predsTest <- predict(model, test)
errorTest <- mean(predsTest != test$medv)
```