K Nearest Neighbours (KNN) para regresión

Contents

1 Introduccion a la clasificacion

1.1 Lectura de datos

```
d = read.csv("datos/MichelinNY.csv")
str(d)
```

```
## 'data.frame':    164 obs. of  6 variables:
##  $ InMichelin     : int  0 0 0 1 0 0 1 1 1 0 ...
##  $ Restaurant.Name: chr  "14 Wall Street" "212" "26 Seats" "44" ...
##  $ Food           : int  19 17 23 19 23 18 24 23 27 20 ...
##  $ Decor          : int  20 17 17 23 12 17 21 22 27 17 ...
##  $ Service        : int  19 16 21 16 19 17 22 21 27 19 ...
##  $ Price          : int  50 43 35 52 24 36 51 61 179 42 ...
```

2 Algoritmo K-vecinos más próximos

- Datos: (yi, x1i, x2i,. . . ,xpi), i = 1,. . . ,n

y	x1	x2		xp
y1	x11	x21	. . .	xp1
y1	x12	x22	. . .	xp2
. . .	. . .	. . .	. . .	. . .
y1	x1n	x2n	. . .	xpn

- Se calcula la distancia euclídea del dato que se quiere clasificar (x1a, x2a,. . . ,xpa) con cada uno de los puntos de la base de datos

$$d_i = \sqrt{(x_{1i} - x_{1a})^2 + (x_{2i} - x_{2a})^2 + \cdots + (x_{pi} - x_{pa})^2}$$

- se ordenan las distancias de menor a mayor y se le asigna al nuevo dato la categoría mayoritaria dentro de los k-datos con menor distancia.
- es decir, si K = 1, se le asigna la categoría del punto más cercano.

- se suelen utilizar K impares para evitar empates.
- a menudo se utilizan regresores estandarizados para que todos los regresores tengan la misma contribución a la distancia.

3 KNN para regresion

```
library(FNN)
```

```
d = read.csv("datos/kidiq.csv")
#d$mom_hs = factor(d$mom_hs, labels = c("no", "si"))
#d$mom_work = factor(d$mom_work, labels = c("notrabaja", "trabaja23", "trabaja1_parcial", "trabaja1_com
```

```
set.seed(123)
n = nrow(d)
pos_train = sample(1:n,round(0.8*n), replace = F)
train_x = d[pos_train,c(3,5)]
test_x = d[-pos_train,c(3,5)]
train_y = d[pos_train,1]
test_y = d[-pos_train,1]
```

```
p = knn.reg(train_x, test_x, train_y, k = 1)
(mse = mean(test_y-p$pred)^2 )
```

```
## [1] 4.092482
```

3.1 Incluir variables cualitativas

```
d$mom_hs1 = ifelse(d$mom_hs == 1,1,0)
d$mom_hs0 = ifelse(d$mom_hs == 0,1,0)
```

```
train_x = d[pos_train,c(3,5,6,7)]
test_x = d[-pos_train,c(3,5,6,7)]
train_y = d[pos_train,1]
test_y = d[-pos_train,1]
```

```
p = knn.reg(train_x, test_x, train_y, k = 1)
(mse = mean(test_y-p$pred)^2 )
```

```
## [1] 0.8455542
```

3.2 Incluir dos regresores cualitativos

```
d$mom_work1 = ifelse(d$mom_work == 1,1,0)
d$mom_work2 = ifelse(d$mom_work == 2,1,0)
d$mom_work3 = ifelse(d$mom_work == 3,1,0)
d$mom_work4 = ifelse(d$mom_work == 4,1,0)
```

```
train_x = d[pos_train,c(3,5:11)]
test_x = d[-pos_train,c(3,5:11)]
train_y = d[pos_train,1]
test_y = d[-pos_train,1]
```

```
p = knn.reg(train_x, test_x, train_y, k = 1)
(mse = mean(test_y-p$pred)^2 )
```

```
## [1] 3.21522
```