KNN para el análisis de variables cuantitativas con R

Contents

1 Algoritmo KNN con R **1**

2 Lectura y preparación de los datos **1**

3 Calculando el valor de K: todos los datos **2**
3.1 Datos de entrenamiento y datos test 2
3.2 Validacion cruzada 3

1 Algoritmo KNN con R

Se quiere predecir el peso de los pingüinos con las siguientes características:

```
xp = data.frame(
  especie = factor(c("Adelie","Gentoo"), levels = c("Adelie","Chinstrap","Gentoo")),
  isla = factor(c("Dream","Biscoe"), levels = c("Biscoe","Dream","Torgersen")),
  long_pico = c(39.8,46.1),
  prof_pico = c(18.4,15.5),
  long_aleta = c(192,202),
  genero = factor(c("hembra","macho"))
)
xp
```

```
##   especie   isla long_pico prof_pico long_aleta genero
## 1  Adelie  Dream      39.8      18.4        192 hembra
## 2  Gentoo Biscoe      46.1      15.5        202  macho
```

utilizando los datos del archivo pinguinos.csv y el algoritmo KNN. Se va a utilizar el paquete FNN de R:

```
# se instala el paquete
# install.packages("FNN")
# se activa el paquete
library(FNN)
```

2 Lectura y preparación de los datos

En primer lugar se leen los datos y se crean los factores correspondientes

```
## se leen los datos
d = read.csv("datos/pinguinos.csv", sep = ";")
```

```
# factores
d$especie = factor(d$especie)
d$isla = factor(d$isla)
d$genero = factor(d$genero)
```

- Se normalizan los regresores numéricos (descargar knn_funciones.R):

```r
# se lee la funcion que normaliza
source("funciones/knn_funciones.R")
```

```r
d1 = data.frame(
  long_pico = knn_normaliza(d$long_pico),
  prof_pico = knn_normaliza(d$prof_pico),
  long_aleta = knn_normaliza(d$long_aleta)
)
```

- Se definen las variables auxiliares:

```r
d1$especie_Ade = ifelse(d$especie == "Adelie", 1, 0)
d1$especie_Chi = ifelse(d$especie == "Chinstrap", 1, 0)
d1$especie_Gen = ifelse(d$especie == "Gentoo", 1, 0)
#
d1$isla_Bis = ifelse(d$isla == "Biscoe", 1, 0)
d1$isla_Dre = ifelse(d$isla == "Dream", 1, 0)
d1$isla_Tor = ifelse(d$isla == "Torgersen", 1, 0)
#
d1$genero_H = ifelse(d$genero == "hembra", 1, 0)
d1$genero_M = ifelse(d$genero == "macho", 1, 0)
```

- Se preparan los valores que se quieren predecir

```r
xp1 = data.frame(
  long_pico = knn_normaliza(xp$long_pico, min(d$long_pico), max(d$long_pico)),
  prof_pico = knn_normaliza(xp$prof_pico, min(d$prof_pico), max(d$prof_pico)),
  long_aleta = knn_normaliza(xp$long_aleta, min(d$long_aleta), max(d$long_aleta))
)
#
xp1$especie_Ade = ifelse(xp$especie == "Adelie", 1, 0)
xp1$especie_Chi = ifelse(xp$especie == "Chinstrap", 1, 0)
xp1$especie_Gen = ifelse(xp$especie == "Gentoo", 1, 0)
#
xp1$isla_Bis = ifelse(xp$isla == "Biscoe", 1, 0)
xp1$isla_Dre = ifelse(xp$isla == "Dream", 1, 0)
xp1$isla_Tor = ifelse(xp$isla == "Torgersen", 1, 0)
#
xp1$genero_H = ifelse(xp$genero == "hembra", 1, 0)
xp1$genero_M = ifelse(xp1$genero == "macho", 1, 0)
```

3 Calculando el valor de K: todos los datos

3.1 Datos de entrenamiento y datos test

```r
set.seed(123)
n = nrow(d)
pos_train = sample(1:n,round(0.8*n), replace = F)
train_x = d1[pos_train,]
test_x = d1[-pos_train,]
train_y = d$peso[pos_train]
test_y = d$peso[-pos_train]
```

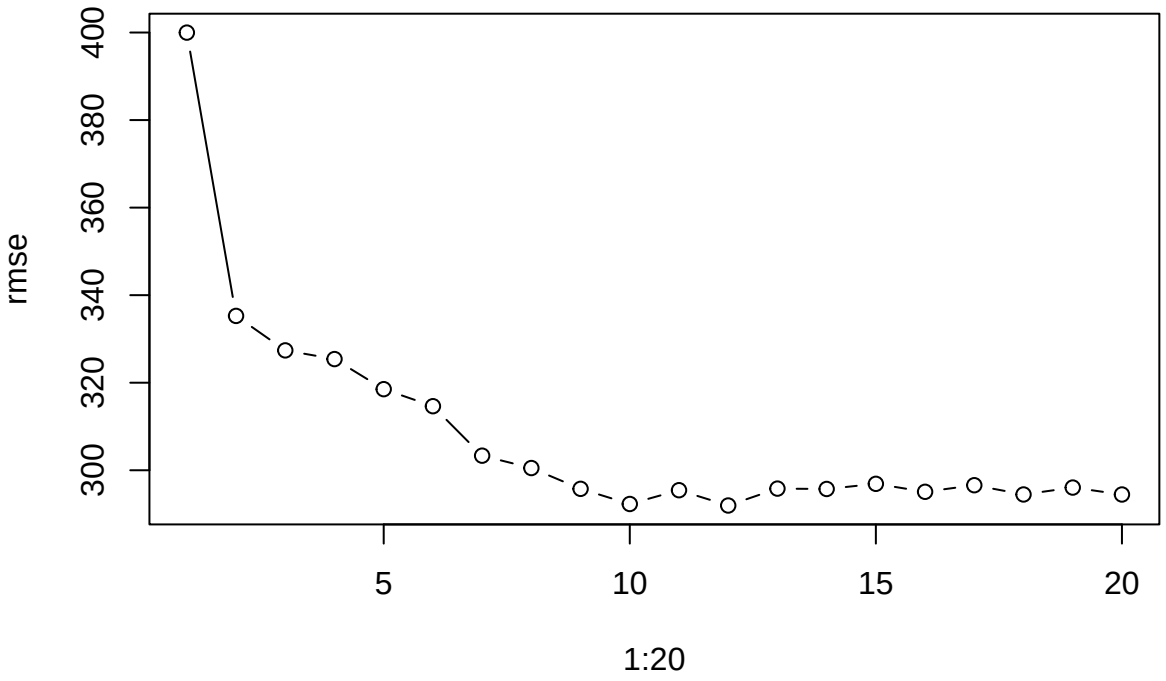
```r
yp = FNN::knn.reg(train_x, test = test_x, y = train_y, k = 1)
```

```r
# RMSE
error = test_y - yp$pred
sqrt(mean(error^2))
```

```
## [1] 399.9767
```

```r
rmse = rep(0,20)
for (ii in 1:20){
  yp = FNN::knn.reg(train_x, test = test_x, y = train_y, k = ii)
  error = test_y - yp$pred
  rmse[ii] = sqrt(mean(error^2))
}
```

```r
plot(1:20,rmse, type = "b")
```



- Calculo de la prediccion

```r
# se utilizan todos los datos
yp = FNN::knn.reg(d1, test = xp1, y = d$peso, k = 10)
yp$pred
```

```
## [1] 3467.5 4837.5
```

3.2 Validacion cruzada

Según la ayuda, Leave one out cross-validation

```r
yp = FNN::knn.reg(d1, y = d$peso, k = 1)
str(yp)
```

```
## List of 7
##  $ call     : language FNN::knn.reg(train = d1, y = d$peso, k = 1)
##  $ k        : num 1
##  $ n        : int 333
##  $ pred     : num [1:333] 3900 2900 3275 3450 3800 ...
```

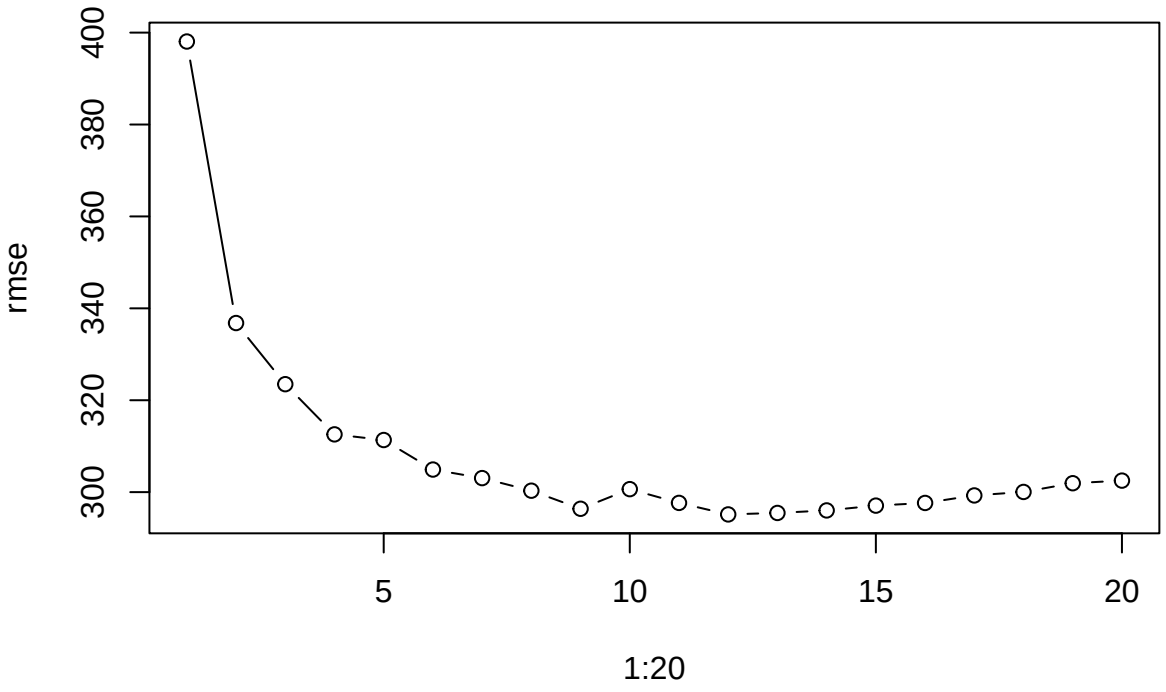
```
##  $ residuals: num [1:333] -150 900 -25 0 -150 ...
##  $ PRESS    : num 52765000
##  $ R2Pred   : num 0.755
##  - attr(*, "class")= chr "knnRegCV"
```

```
# RMSE
sqrt(mean(yp$residuals^2))
```

```
## [1] 398.0621
```

```
rmse = rep(0,20)
for (ii in 1:20){
  yp = FNN::knn.reg(d1, y = d$peso, k = ii)
  rmse[ii] = sqrt(mean(yp$residuals^2))
}
```

```
plot(1:20,rmse, type = "b")
```



- Calculo de la prediccion

```
# se utilizan todos los datos
yp = FNN::knn.reg(d1, test = xp1, y = d$peso, k = 9)
yp$pred
```

```
## [1] 3469.444 4886.111
```