K-Vecinos más cercanos (KNN) de penguins

Omar Sanchez Hernandez

5/6/2022

Introducción

K-vecinos más cercanos es un método para clasificar casos basándose en sus similitudes a otros casos.

Librerias necesarias

```
library(MASS)
library(readxl)
```

Cargar los datos de los pengüins

```
ruta = "penguins.xlsx"
Z<-as.data.frame(read_excel(ruta))[,c(4,5,6,7,2)]
Z$especie = as.factor(Z$especie)
colnames(Z)
```

```
## [1] "largo_pico_mm"   "grosor_pico_mm"  "largo_aleta_mm"  "masa_corporal_g"
## [5] "especie"
```

Definir la matriz de datos y la variable respuesta con las clasificaciones

```
x<-Z[,1:4]
y<-Z[,5]
dim(Z)
```

```
## [1] 344   5
```

La matriz de datos tiene 344 observaciones y 5 variables.

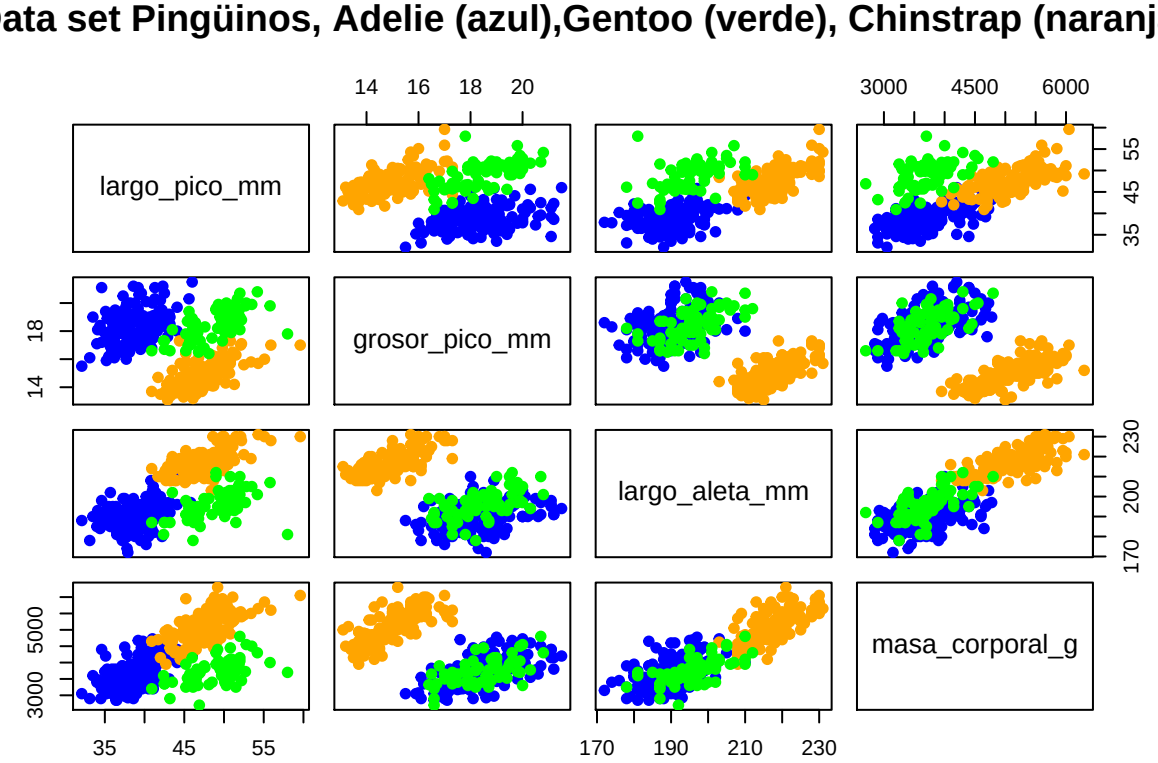
Se definen las variables y observaciones

```
n<-nrow(x)
p<-ncol(x)
dim(Z)
```

```
## [1] 344   5
```

Grafico scatter plot

```
col.iris<-c("blue","green","orange")[y]
pairs(x, main="Data set Pingüinos, Adelie (azul),Gentoo (verde), Chinstrap (naranja)",
      pch=19,col=col.iris)
```



Se fija una “semilla” para tener valores iguales

```
set.seed(64)
library(class)
```

Creacion de los ciclos para k=1 hasta k=20

Se selecciona el valor de k que tenga el error mas bajo.

Inicialización de una lista vacia de tamaño 20

```
knn.class<-vector(mode="list",length=20)
knn.tables<-vector(mode="list", length=20)
```

Clasificaciones erroneas

```
knn.mis<-matrix(NA, nrow=20, ncol=1)
knn.mis
```

```
##       [,1]
##  [1,]   NA
##  [2,]   NA
##  [3,]   NA
##  [4,]   NA
##  [5,]   NA
##  [6,]   NA
##  [7,]   NA
##  [8,]   NA
##  [9,]   NA
## [10,]   NA
## [11,]   NA
## [12,]   NA
## [13,]   NA
## [14,]   NA
## [15,]   NA
## [16,]   NA
## [17,]   NA
## [18,]   NA
## [19,]   NA
## [20,]   NA
```

```
# Se hace el ciclo
for(k in 1:20){
  knn.class[[k]]<-knn.cv(x,y,k=k)
  knn.tables[[k]]<-table(y,knn.class[[k]])
  # la suma de las clasificaciones menos las correctas
  knn.mis[k]<- n-sum(y==knn.class[[k]])
}
knn.mis
```

```
##       [,1]
##  [1,]   44
##  [2,]   60
##  [3,]   72
##  [4,]   69
##  [5,]   69
##  [6,]   76
##  [7,]   78
##  [8,]   75
##  [9,]   75
## [10,]   73
## [11,]   73
## [12,]   71
## [13,]   74
## [14,]   78
## [15,]   81
```

```
## [16,]   87
## [17,]   88
## [18,]   88
## [19,]   83
## [20,]   79
```

Numero optimo de k-vecinos

```
which(knn.mis==min(knn.mis))
```

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

```
knn.tables[[1]]
```

```
##
## y           Adelie Chinstrap Gentoo
##   Adelie       136        12      4
##   Chinstrap     18        46      4
##   Gentoo         2         4    118
```

El numero optimo de k-vecinos es 1.

El mas eficiente es k=1 se señala el k mas eficiente

```
k.opt<-1
knn.cv.opt<-knn.class[[k.opt]]
knn.cv.opt
```

```
##   [1] Adelie    Adelie    Chinstrap Adelie    Adelie    Adelie    Adelie
##   [8] Adelie    Adelie    Adelie    Adelie    Adelie    Adelie    Adelie
##  [15] Adelie    Adelie    Adelie    Chinstrap Adelie    Adelie    Adelie
##  [22] Chinstrap Adelie    Adelie    Adelie    Adelie    Adelie    Chinstrap
##  [29] Adelie    Adelie    Chinstrap Adelie    Adelie    Adelie    Adelie
##  [36] Adelie    Adelie    Adelie    Adelie    Adelie    Adelie    Adelie
##  [43] Adelie    Adelie    Adelie    Adelie    Adelie    Adelie    Adelie
##  [50] Adelie    Adelie    Adelie    Adelie    Chinstrap Adelie    Adelie
##  [57] Adelie    Adelie    Adelie    Adelie    Adelie    Adelie    Adelie
##  [64] Adelie    Adelie    Adelie    Adelie    Adelie    Adelie    Chinstrap
##  [71] Adelie    Adelie    Adelie    Chinstrap Adelie    Adelie    Adelie
##  [78] Adelie    Adelie    Adelie    Adelie    Gentoo    Adelie    Adelie
##  [85] Adelie    Adelie    Adelie    Adelie    Adelie    Adelie    Adelie
##  [92] Adelie    Adelie    Adelie    Adelie    Adelie    Adelie    Gentoo
##  [99] Adelie    Adelie    Adelie    Gentoo    Adelie    Adelie    Adelie
## [106] Adelie    Adelie    Adelie    Adelie    Gentoo    Adelie    Adelie
## [113] Adelie    Adelie    Adelie    Adelie    Adelie    Chinstrap Adelie
## [120] Adelie    Adelie    Adelie    Adelie    Chinstrap Adelie    Adelie
## [127] Adelie    Adelie    Adelie    Adelie    Adelie    Chinstrap Adelie
## [134] Adelie    Adelie    Adelie    Adelie    Adelie    Adelie    Adelie
## [141] Chinstrap Adelie    Adelie    Adelie    Adelie    Adelie    Adelie
## [148] Adelie    Adelie    Adelie    Adelie    Adelie    Chinstrap Gentoo
## [155] Gentoo    Gentoo    Gentoo    Chinstrap Gentoo    Gentoo    Gentoo
## [162] Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo
## [169] Adelie    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo
## [176] Gentoo    Gentoo    Gentoo    Chinstrap Gentoo    Gentoo    Gentoo
## [183] Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo
## [190] Gentoo    Adelie    Gentoo    Chinstrap Gentoo    Gentoo    Gentoo
## [197] Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo
## [204] Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo
## [211] Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo
## [218] Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo
## [225] Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo
## [232] Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo
## [239] Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo
## [246] Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo
## [253] Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo
## [260] Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo
## [267] Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo
## [274] Gentoo    Gentoo    Gentoo    Chinstrap Chinstrap Chinstrap Chinstrap
## [281] Adelie    Chinstrap Adelie    Adelie    Adelie    Chinstrap Chinstrap
## [288] Chinstrap Chinstrap Chinstrap Chinstrap Chinstrap Chinstrap Chinstrap
## [295] Chinstrap Adelie    Adelie    Chinstrap Adelie    Chinstrap Chinstrap
## [302] Adelie    Chinstrap Chinstrap Adelie    Gentoo    Adelie    Chinstrap
## [309] Chinstrap Chinstrap Chinstrap Chinstrap Adelie    Gentoo    Adelie
## [316] Gentoo    Chinstrap Chinstrap Adelie    Adelie    Chinstrap Adelie
```

```
## [323] Chinstrap Gentoo    Chinstrap Chinstrap Adelie    Chinstrap Chinstrap
## [330] Chinstrap Chinstrap Chinstrap Adelie    Chinstrap Chinstrap Chinstrap
## [337] Chinstrap Chinstrap Chinstrap Adelie    Chinstrap Chinstrap Chinstrap
## [344] Chinstrap
## Levels: Adelie Chinstrap Gentoo
```

Tabla de contingencia con las clasificaciones buenas y malas

```
knn.tables[[k.opt]]
```

```
##
## y           Adelie Chinstrap Gentoo
##   Adelie       136        12      4
##   Chinstrap     18        46      4
##   Gentoo         2         4    118
```

Cantidad de observaciones mal clasificadas

```
knn.mis[k.opt]
```

```
## [1] 44
```

El modelo se equivoca clasificando a 44 sujetos.

Error de clasificacion (MR)

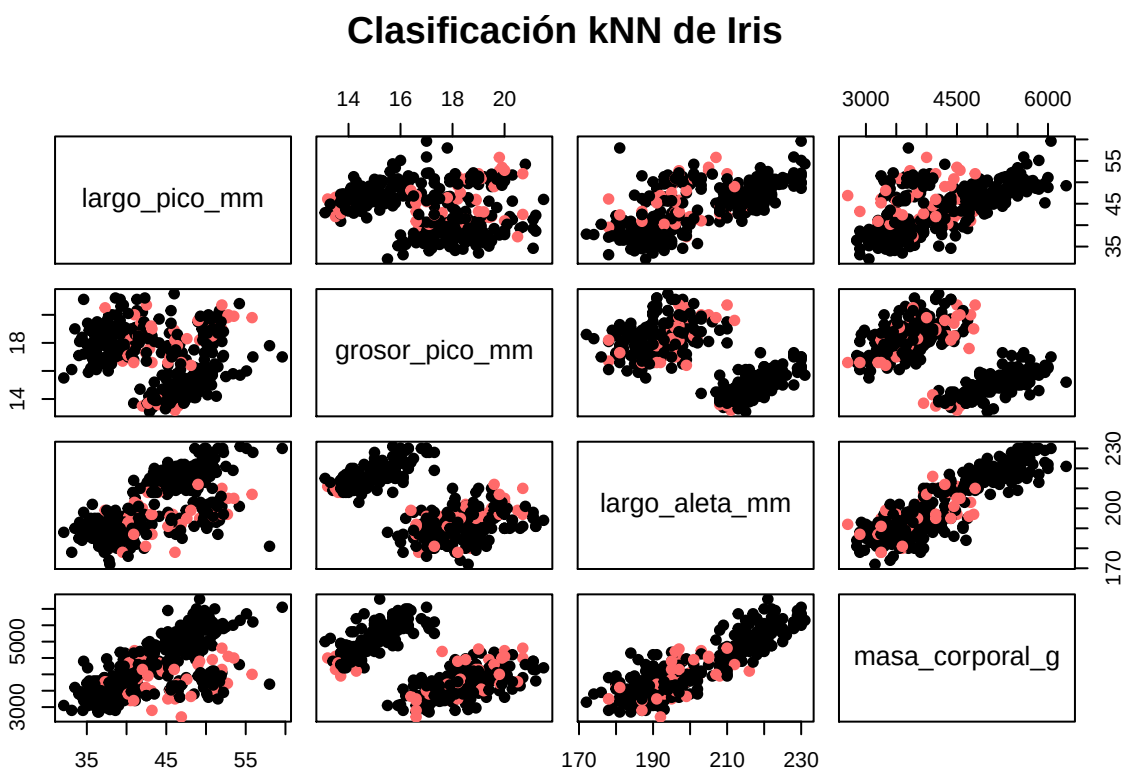
```
(knn.mis[k.opt]/n)*100
```

```
## [1] 12.7907
```

El modelo presenta un error de mala clasificación del 12.79%

Grafico de clasificaciones correctas y erroneas

```
# Grafico de clasificaciones correctas y erroneas
col.knn.iris<-c("indianred1","black")[1*(y==knn.cv.opt)+1]
pairs(x, main="Clasificación kNN de Iris",
      pch=19, col=col.knn.iris)
```



Se puede observar que existen muchos valores mal clasificados que se traslapan con otros.