

k-Vecinos más cercanos (kNN)

Ana Karen Martínez Marín

2022-05-28

Introducción

Análisis de vecinos más próximos es un método para clasificar casos basándose en su parecido a otros casos. En el aprendizaje automático, se desarrolló como una forma de reconocer patrones de datos sin la necesidad de una coincidencia exacta con patrones o casos almacenados.

Para este caso se trabajó con la matriz **penguins**.

```
# Se carga la base de datos
library(readxl)
penguins <- read_excel("/cloud/project/Estadística Multivariada/penguins.xlsx")
```

```
Z<-penguins
colnames(Z)
```

```
## [1] "ID"          "especie"      "isla"         "largo_pico_mm"
## [5] "grosor_pico_mm" "largo_aleta_mm" "masa_corporal_g" "genero"
## [9] "año"
```

```
# Se convierte la base de datos a un data.frame
Z<-data.frame(Z)
```

Se define la matriz de datos y la variable respuesta con las clasificaciones. Para este caso la clasificación será por especie.

```
x<-Z[,4:7]
y<-Z[,2]
```

Se definen las variables y las observaciones

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

Método k-vecinos más próximos

```
#Se carga la librería
library(class)
```

Se fija una “semilla” (para obtener los mismos valores).

```
set.seed(1500)
```

Creación de los ciclos

En este caso será un ciclo de $k=1$ hasta $k=30$ (el “ k ” puede variar de manera arbitraria).

```

# Inicialización de una lista vacía de tamaño 30
knn.class<-vector(mode="list",length=30)
knn.tables<-vector(mode="list", length=30)

# Clasificaciones erróneas
knn.mis<-matrix(NA, nrow=30, ncol=1)

for(k in 1:30){
  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,]  54
## [3,]  71
## [4,]  77
## [5,]  74
## [6,]  72
## [7,]  78
## [8,]  75
## [9,]  75
## [10,] 74
## [11,] 73
## [12,] 73
## [13,] 72
## [14,] 74
## [15,] 82
## [16,] 88
## [17,] 88
## [18,] 87
## [19,] 84
## [20,] 82
## [21,] 81
## [22,] 84
## [23,] 87
## [24,] 86
## [25,] 87
## [26,] 89
## [27,] 92
## [28,] 91
## [29,] 91
## [30,] 90

```

```

# Número óptimo de k-vecinos
which(knn.mis==min(knn.mis))

```

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

Se visualiza el resultado que arrojó el ciclo con el error más bajo.

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

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

En la especie Adelie 18 están clasificados como Chinstrap y 2 en Gentoo, con la especie Chinstrap hay un número elevado que no están bien clasificados dentro de esa especie, ya que son 12 los que identifica como Adelie y 4 como Gentoo. Respecto a la especie de Gentoo en total son 8 los pinguinos que no están bien clasificados que son 4 en Adelie y 4 en Chinstrap.

```
# Se señala el k mas eficiente:  
k.opt<-1
```

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

Se visualiza la tabla de contingencia con las clasificaciones buenas y malas. En este caso es el número 1 ya que en el resultado del ciclo fue el número más pequeño de las 30 iteraciones.

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

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

La cantidad de observaciones mal clasificadas:

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

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

```
# Error de clasificacion (MR)  
knn.mis[k.opt]/n
```

```
## [1] 0.127907
```

Esto quiere decir que de 100 pinguinos, entre 12 y 13 no están bien clasificados con respecto a la especie.

- Ahora se crea un gráfico identificando las clasificaciones correctas y erróneas.

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

Clasificación kNN de pingüinos por especie

