k-Vecinos más cercanos (kNN)

Lino Oswaldo Sanchez

25/6/2022

Introducción

Con este análisis de K-Vecinos(vecino más próximo) es un método para clasificar observaciones "casos" basando en su parecido a las otras observaciones.Es un método de clasificación no paramétrico. Es decir, no requiere asumir ninguna distribución para variable aleatoria. La idea es buscar, para una nueva observación que se requiere clasificar, sus k vecinos más cercanos. Es decir, las observaciones más cercanas respecto a una medida de distancia.

Para este Anáslisis usaremos la matriz de **penguins** donde tenemos las caracteristicas de tres distintas especies de pingÜinos.

Base de datos

```
library(readxl)
penguins <- read_excel("Analisis canonico/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"
```

Llamamos la base y vemos los nombres de las variables para recordar las varibles que en otros ejercicios hemos trabajado.

```
Z<-data.frame(Z)
```

Convertimos la base a un data frame para trabajar y la renombramos Z

X sera de la variable 4 a 7

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

Y sera la especie

```
y<-Z[,2]
```

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

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

Definimos las variables y las observaciones

Algoritmo k-vecinos más próximos

Librería necesaria.

Se fija una “semilla” *para obtener los mismos valores al replicar el ejercicio.*

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

Creación de los ciclos

para este caso usaremos un ciclo de k=1 hasta k=30 “el”k" puede variar de manera arbitraria".

```
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]])
}
```

Se crea una función ## 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 y en este caso es **1**

```
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]]
```

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

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

Lo visualizamos en una tabla de contingencia con las clasificaciones buenas y malas, para este caso es el numero uno ya que en el resultado del ciclo fue el numero más pequeño de las treinta iteraciones.

La cantidad de observaciones mal clasificadas:

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

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

Error de clasificacion (MR)

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

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

Gráfico identificando las clasificaciones correctas y erróneas.

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