K-VECINOS MAS CERCANOS (KNN)

Saraí Campos Varela

2022-06-03

Introducción

El algoritmo de k vecinos más cercanos, también conocido como KNN o k-NN, es un clasificador de aprendizaje supervisado no paramétrico, que utiliza la proximidad para hacer clasificaciones o predicciones sobre la agrupación de un punto de datos individual.

Librerías

```
library(MASS)
library(class)
```

Matriz

Se trabajará con la base de datos de iris precargada en R.

```
Z<-as.data.frame(iris)
colnames(Z)
```

```
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
```

Se define la matriz de datos y la variable respuesta, con las clasificaciones.

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

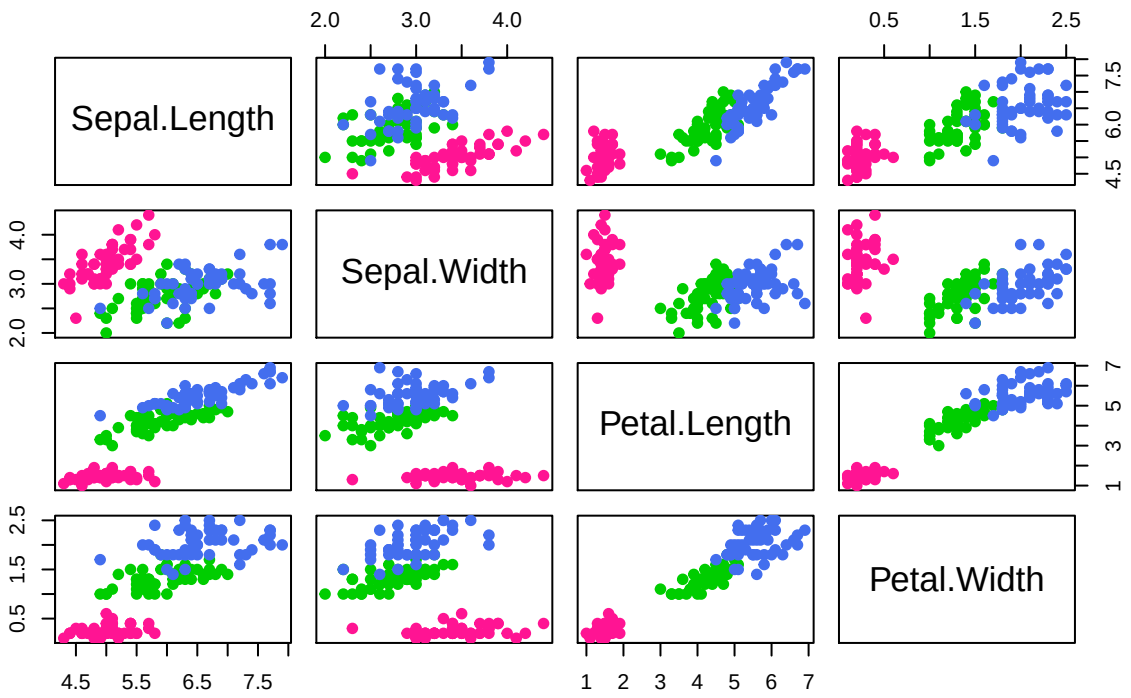
Se definen las variables y las observaciones.

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

Se realiza el gráfico scatter plot.

```
col.iris<-c("deeppink","green3","royalblue2")[y]
```

```
pairs(x, main="Data set Iris, Setosa(rosa), Versicolor(verde), Virginica(azul)", pch=19,col=col.iris)
```



Método k-vecinos más próximos

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

```
set.seed(1000)
```

Creación de los ciclos

En este caso será un ciclo de k=1 hasta k=20 (el "k" puede variar de manera arbitraria). Inicialización de una lista vacia de tamaño 20

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

Clasificaciones erróneas

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

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

Número óptimo de k-vecinos

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

```
## [1] 14 18 19
```

Se visualizan los resultados que nos arrojó el ciclo con el error más bajo.

```
knn.tables[[14]]
```

```
##
## y            setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         48         2
##   virginica       0          1        49
```

```
knn.tables[[18]]
```

```
##
## y            setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         48         2
##   virginica       0          1        49
```

```
knn.tables[[19]]
```

```
##
## y            setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         48         2
##   virginica       0          1        49
```

El resultado en los tres casos es el mismo, todas las setosa están bien clasificadas,y en versicolor 48 flores están bien clasificadas y dos de ellas se identifican como virginica de las cuales sólo una es clasificada como versicolor.

Se señala el k mas eficiente

```
k.opt<-14
knn.cv.opt<-knn.class[[k.opt]]
```

Se visualiza la tabla de contingencia con las clasificaciones buenas y malas:

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

```
##
## y            setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         48         2
##   virginica       0          1        49
```

La cantidad de observaciones mal clasificadas:

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

```
## [1] 3
```

Esto quiere decir que de 100 flores, 2 no están bien clasificadas.

Error de clasificacion (MR)

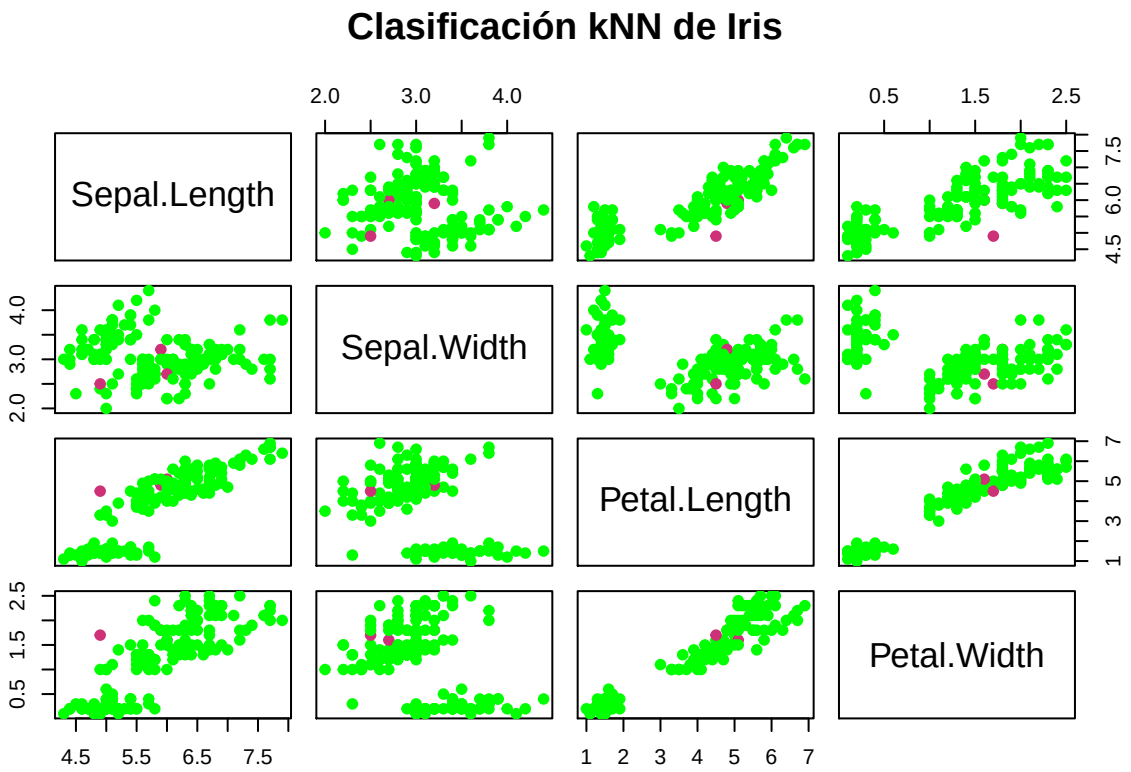
```
knn.mis[k.opt]/n
```

```
## [1] 0.02
```

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

Grafico de clasificaciones

```
col.knn.iris<-c("violetred3","green")[1*(y==knn.cv.opt)+1]
pairs(x, main="Clasificación kNN de Iris",
      pch=19, col=col.knn.iris)
```



PRACTICA PENGUINS

En esta practica se realizará el mismo ejercicio pero con una matriz diferente, en este caso se trabajó con la matriz **penguins** la cualse encuentra en la libreria datos ya precargada en R, pero en mi caso la extraeré de excel.

Libreria

```
library(readxl)
```

Obtenemos y previsualizamos la matriz

```
X <- read_excel("C:/Users/USUARIO/Documents/MULTIVARIADA/penguins.xlsx")
X[1:10,]
```

```
## # A tibble: 10 x 9
##    ID    especie isla      largo_pico_mm grosor_pico_mm largo_aleta_mm
##    <chr> <chr>   <chr>             <dbl>          <dbl>          <dbl>
```

```
##  1 i1    Adelie  Torgersen          39.1           18.7           181
##  2 i2    Adelie  Torgersen          39.5           17.4           186
##  3 i3    Adelie  Torgersen          40.3           18             195
##  4 i4    Adelie  Torgersen          37.8           18.1           190
##  5 i5    Adelie  Torgersen          36.7           19.3           193
##  6 i6    Adelie  Torgersen          39.3           20.6           190
##  7 i7    Adelie  Torgersen          38.9           17.8           181
##  8 i8    Adelie  Torgersen          39.2           19.6           195
##  9 i9    Adelie  Torgersen          34.1           18.1           193
## 10 i10   Adelie  Torgersen          42             20.2           190
## # ... with 3 more variables: masa_corporal_g <dbl>, genero <chr>, año <dbl>
```

Exploración de la matriz

```
colnames(X)
```

```
## [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

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

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

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

Se definen las variables y las observaciones

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

Método k-vecinos más próximos

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 vacia 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]])
  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
```

La especie Adelie 18 están clasificados como Chinstrap y 2 en Gentoo, con la especie Chinstrap, existe un número elevado que no está bien clasificados dentro de la especie, ya que se identifican 12 como Adelie y 4 como Gentoo. Respecto a la especie de Gentoo en total nos encontramos 8 pinguinos, de los cuales todos están bien clasificados, 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
```

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

Error de clasificacion (MR)

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

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

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

Grafico de clasificaciones

```
col.knn.iris<-c("mediumpurple4","turquoise")[1*(y==knn.cv.opt)+1]
pairs(x, main="Clasificación kNN de pinguinos por género",
      pch=19, col=col.knn.iris)
```