Tarea 1 ANALITICA

Jhonatan Smith Garcia

14/10/2021

```
a = read.csv(file.choose(), sep = ";")
datos = data.frame(a)
```

Suponga que llega la observacion con X1=X2=X3 = 0

(A)

Compute la distancia euclidiana entre cada punto de la observacion dada.

- $SeaX_1$ un vector dado por $X_1 = [x_1, x_2, x_3, .., x_n]$
- $SeaY_1$ un vector dado por $Y_2 = [y_1, y_2, y_3, .., y_n]$

Entonces, se define la distancia euclidiana como

$$D = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2 + (y_3 - x3) + ... + (y_n - x_n)^2)}$$

para vectores n dimensionales.

En este caso, las observaciones que se tienen son las siguientes:

```
datos
```

```
##   X1 X2 X3     Y
## 1  0  3  0   Red
## 2  2  0  0   Red
## 3  0  1  3   Red
## 4  0  1  2 Green
## 5 -1  0  1 Green
## 6  1  1  1   Red
```

De esta manera, si la nueva entrada es con todos sus componentes nulos, $X_7 = (0, 0, 0)$ entonces:

- Observación 1 = $\sqrt{(0-0)^2 + (3-0)^2 + (0-0)^2} = 3$
- Observación 2 = $\sqrt{(2-0)^2 + (0-0)^2 + (0-0)^2} = 2$
- Observación 3 = $\sqrt{(0-0)^2 + (1-0)^2 + (3-0)^2} = 3.16$
- Observación 4 = $\sqrt{(0-0)^2 + (1-0)^2 + (2-0)^2} = 2.24$
- Observación 5 = $\sqrt{(-1-0)^2 + (0-0)^2 + (1-0)^2} = 1.41$
- Observación 6 = $\sqrt{(1-0)^2 + (1-0)^2 + (1-0)^2} = 1.73$

Son las distancias de cada observacion respecto al nuevo punto asignado.

#(B)

Si K = 1 implica que el metodo de los K vecinos mas cercanos tomará los K-esimos vecinos del punto observado.

En este caso, K = 1 implica que:

- Dada la observacion $X_7 = [0, 0, 0]$ se tiene que la observacion mas cercana será $X_5 = [-1, 0, 1]$ pues su distancia es la menor al punto.

*Probabilidad de clasificar al punto mas cercano como red dado el metodo Knn

$P(Y = "Red"|x = x_0) = I(Y = "Red") = 0$

*Probabilidad de clasificar al punto mas cercano como green dado el metodo Knn

$P(Y = "Green"|x = x_0) = I(Y = "Green") = 1$

Donde, la Indicadora "I" toma el valor 1 o 0 dependiendo de la probabilidad del evento dada la categoria. En este caso, el punto mas cercano es el dado que toma el valor "Green"; anulando la indicadora en el segundo caso y asignandole el calor de 1 en el segundo.

*En R:

Recuerde que siempre se deben normalizar los datos, para ello;

```
x_train = normilize(datos[,1:3])
Test = c(0,0,0)
cl= datos[,4] # Selecciona la Y categorica de respuesta
fit.knn_train = knn(train=x_train,test = Test,cl =cl, k=1, prob = TRUE)
fit.knn_train
```

```
## [1] Green
## attr(,"prob")
## [1] 1
## Levels: Green Red
```

En este caso, la primera fila nos asigna el valor a la categoria del nuevo dato. "Green". Que es consecuente con lo ya calculado.

(C)

*Para K= 3 se tiene el siguiente proceso:

Se han de seleccionar los 3 vecinos mas cercanos (k=3) del total de los datos obtenidos. Para este caso, dichos vecinos son X_2, X_5, X_6

Asi, se tiene a las observaciones:

- $X_2 = [2, 0, 0]$
- $X_5 = [-1, 0, 1]$
- $X_6 = [1, 1, 1]$

Y la probabilidad de clasificacion del evento como "Red" será:

$P(Y_0 = "Red"|x_0) = \frac{1}{3}I * (Y_2 = "Red") + \frac{1}{3}I * (Y_5 = "Red") + \frac{1}{3}I * (Y_6 = "Red")$

$= \frac{1}{3} * 1 + \frac{1}{3} * (0) + \frac{1}{3} * 1 = 0.667$

Y para que la clasificacion del nuevo evento sea "Green" será:

$P(Y_0 = "Green"|x_0) = \frac{1}{3}I * (Y_2 = "Green") + \frac{1}{3}I * (Y_5 = "Green") + \frac{1}{3}I * (Y_6 = "Green")$

$= \frac{1}{3} * (0) + \frac{1}{3} * (1) + \frac{1}{3} * (0) = 0.337$

*Utilizando R:

```
library(class)
x_train = normilize(datos[,1:3])
Test = c(0,0,0)
cl= datos[,4] # Selecciona la Y categorica de respuesta
fit.knn_train = knn(train=x_train,test = Test,cl =cl, k=3, prob = TRUE)
fit.knn_train
```

```
## [1] Red
## attr(,"prob")
## [1] 0.6666667
## Levels: Green Red
```

Nuevamente, con una probabilidad del 66.667% R clasifica a la nueva observacion como “Red” y esto, nuevamente; es consistente con los calculos realizados.

#(D)

Se debe de recordar que el metodo Knn tiene una tendencia a ser no lineal a valores mas pequeños de K y una tendencia a ser mas rigido (lineal) a valores de K mas grandes, por consiguiente, si se sabe que la frontera es altamente no lineal, la eleccion de un K pequeño seria la mas optima puesto que al incluir menos vecinos toma grupos reducidos de clasificacion y estos se ven menos influenciados por la mayoria del ruido de los datos.