Clasificación con K-NN

Javier

25/10/2020

```
# Limpiamos
rm(list = ls())

#Librerias
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.0.3

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.0.3

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(ISLR)

## Warning: package 'ISLR' was built under R version 4.0.3

library(class)

## Warning: package 'class' was built under R version 4.0.3

library(caret)

## Warning: package 'caret' was built under R version 4.0.3

## Loading required package: lattice

library(ROCR)

## Warning: package 'ROCR' was built under R version 4.0.3

library(pROC)

## Warning: package 'pROC' was built under R version 4.0.3

## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var

#Datos
dim(Caravan)

## [1] 5822   86

attach(Caravan)
summary(Purchase)

##   No  Yes
## 5474  348

#Estandarizamos la libería
standardized.X=scale(Caravan[,-86])

# Creamos test y train
test =1:1000
train.X=standardized.X[-test ,]
# Observaciones de 1-1000
test.X=standardized.X[test ,]
train.Y=Purchase[-test]
test.Y=Purchase[test]

# Agregando los componentes con KNN
set.seed(1)
# Creamos la matriz de confusion (K=1)
knn.pred=knn(train.X,test.X, train.Y,k=1)
mean(test.Y!= knn.pred)

## [1] 0.118

mean(test.Y!=" No")

## [1] 1

# Matriz de confusion (K=1)
table(knn.pred,test.Y)

##         test.Y
## knn.pred  No Yes
##      No  873  50
##      Yes  68   9

temporal<- table(knn.pred, test.Y)

# Porcentaje de acierto
```

```r
cat("Porcentaje de acierto: ", (temporal[4]/(temporal[2] + temporal[4]) *
100), "%")
```

```
## Porcentaje de acierto:  11.68831 %
```

```r
# Creamos la matriz de confusion (K=3)
knn.pred= knn(train.X, test.X, train.Y,k=3)
table(knn.pred, test.Y)
```

```
##         test.Y
## knn.pred  No Yes
##      No  920  54
##      Yes  21   5
```

```r
temporal<- table(knn.pred, test.Y)
# Porcentaje de acierto
cat("Porcentaje de acierto: ", (temporal[4]/(temporal[2] + temporal[4]) *
100), "%")
```

```
## Porcentaje de acierto:  19.23077 %
```

```r
# Creamos la matriz de confusion (K=5)
knn.pred= knn(train.X, test.X, train.Y, k=5)
table(knn.pred, test.Y)
```

```
##         test.Y
## knn.pred  No Yes
##      No  930  55
##      Yes  11   4
```

```r
temporal<- table(knn.pred, test.Y)
# Porcentaje de acierto
cat("Porcentaje de acierto: ", (temporal[4]/(temporal[2] + temporal[4]) *
100), "%")
```

```
## Porcentaje de acierto:  26.66667 %
```

```r
# Creamos la matriz de confusion (K=9)
knn.pred= knn(train.X, test.X, train.Y, k=9)
table(knn.pred, test.Y)
```

```
##         test.Y
## knn.pred  No Yes
##      No  941  58
##      Yes   0   1
```

```r
temporal<- table(knn.pred, test.Y)
# Porcentaje de acierto
cat("Porcentaje de acierto: ", (temporal[4]/(temporal[2] + temporal[4]) *
100), "%")
```

```
## Porcentaje de acierto:  100 %
```

```r
# Creamos la matriz de confusion (K=17)
knn.pred= knn(train.X, test.X, train.Y, k=17)
table(knn.pred, test.Y)
```

```
##         test.Y
## knn.pred  No Yes
##      No  941  59
##      Yes   0   0
```

```r
temporal<- table(knn.pred, test.Y)
# Porcentaje de acierto
cat("Porcentaje de acierto: ", (temporal[4]/(temporal[2] + temporal[4]) *
100), "%")
```

```
## Porcentaje de acierto:  NaN %
```

No me funcionaba la curva ROC, pero podemos apreciar con los que he realizado anteriormente que con cualquier valor de k>=9 es un porcentaje de acierto del 100%