# Trabajo 3

*Samuel Cardenete Rodríguez y Juan José Sierra González*

*11 de mayo de 2017*

## Introducción:

Para la realización de esta práctica obtendremos el ajuste de de modelos lineales basados en dos problemas centrados en dos conjuntos de datos diferentes. En primer lugar trabajaremos con un problema de clasificación, basado en el conjunto de datos "South African Heart Disease", para el reconocimiento de enfermedades cardiovasculares en una población de sudáfrica; y un problema de regresión, basado en el conjunto de datos "Los Angeles Ozone", para predecir los niveles de Ozono en Los angeles.

Comenzaremos primeramente abordando el problema de clasificación:

## Clasificación: "South African Heart Disease"

Procedemos a la lectura de datos tanto de la base de datos de clasificación 'South African Heart Disease', como para la de regresión 'Los Angeles Ozone'.

Lo primero es hacer numéricos aquellos atributos que esten definidos como texto en nuestro conjunto de datos:

Ahora procedemos a ver si reducimos los datos, para ello aplicamos el PCA (lo que signifique..) sobre el conjunto train, y con las transformaciones indicadas en los parametros. tras esto observamos el parametro rotation, como influye la varianza de cada parámetro en los parámetros preprocesator mediante PCA obtenidos (PC1, PC2...)

```
sudafricaTrans = preProcess(sudafrica_train, method = c("BoxCox", "center", "scale", "pca"),thresh = 0.9
summary(sudafricaTrans$rotation)
```

```
##       PC1                PC2                PC3          
##  Min.   :-0.47849   Min.   :-0.46176   Min.   :-0.30144  
##  1st Qu.:-0.37722   1st Qu.:-0.30116   1st Qu.:-0.19191  
##  Median :-0.29432   Median :-0.07644   Median :-0.02913  
##  Mean   :-0.28825   Mean   :-0.05606   Mean   : 0.07046  
##  3rd Qu.:-0.22889   3rd Qu.: 0.18078   3rd Qu.: 0.29827  
##  Max.   :-0.03329   Max.   : 0.49188   Max.   : 0.66094  
##       PC4                 PC5                PC6          
##  Min.   :-0.242637   Min.   :-0.53414   Min.   :-0.87139  
##  1st Qu.:-0.162339   1st Qu.:-0.08373   1st Qu.:-0.12690  
##  Median : 0.004241   Median :-0.03965   Median : 0.05408  
##  Mean   : 0.085613   Mean   :-0.01638   Mean   :-0.02827  
##  3rd Qu.: 0.189590   3rd Qu.: 0.05600   3rd Qu.: 0.19933  
##  Max.   : 0.792789   Max.   : 0.69538   Max.   : 0.25503  
##       PC7                 PC8          
##  Min.   :-0.513369   Min.   :-0.47345  
##  1st Qu.:-0.262509   1st Qu.:-0.21153  
##  Median : 0.070389   Median : 0.01293  
##  Mean   : 0.004661   Mean   : 0.01647  
##  3rd Qu.: 0.177744   3rd Qu.: 0.11471
```

```
##  Max.   : 0.447704   Max.   : 0.73000
```

```
nearZeroVar(sudafricaTrans$rotation)
```

```
## integer(0)
```

Como comprobamos con la función near cero observamos que no existe ningun atributo cuyas varianzas respecto a las demás sean todas cercanas a cero, por lo que quitar un atributo no sería aconsejable pues no podemos asegurar que no sea importante. Por tanto nos quedamos con 10 atributos.

Para conlcuir el preprocesamiento de los datos, los centramos, aplicamos el BoxCox y los escalamos:

```
sudafricaTrans = preProcess(sudafrica_train[,-ncol(sudafrica_train)], method = c("BoxCox", "center", "s
sudafrica_train[,-ncol(sudafrica_train)] =predict(sudafricaTrans,sudafrica_train[,-ncol(sudafrica_train
```

Para realizarar un modelo, vemos cuales son las características más representativao (varianza mayor):

```
regsub_sudafrica =regsubsets(datos_sudafrica[,-ncol(datos_sudafrica)], datos_sudafrica[,ncol(datos_suda

summary(regsub_sudafrica)
```

```
## Subset selection object
## 9 Variables  (and intercept)
##           Forced in Forced out
## famhist       FALSE      FALSE
## sbp           FALSE      FALSE
## tobacco       FALSE      FALSE
## ldl           FALSE      FALSE
## adiposity     FALSE      FALSE
## typea         FALSE      FALSE
## obesity       FALSE      FALSE
## alcohol       FALSE      FALSE
## age           FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          famhist sbp tobacco ldl adiposity typea obesity alcohol age
## 1  ( 1 ) " "     " " " "     " " " "       " "   " "     " "     "*"
## 2  ( 1 ) "*"     " " " "     " " " "       " "   " "     " "     "*"
## 3  ( 1 ) "*"     " " "*"     " " " "       " "   " "     " "     "*"
## 4  ( 1 ) "*"     " " "*"     "*" " "       " "   " "     " "     "*"
## 5  ( 1 ) "*"     " " "*"     "*" " "       "*"   " "     " "     "*"
## 6  ( 1 ) "*"     " " "*"     "*" " "       "*"   "*"     " "     "*"
## 7  ( 1 ) "*"     "*" "*"     "*" " "       "*"   "*"     " "     "*"
## 8  ( 1 ) "*"     "*" "*"     "*" "*"       "*"   "*"     " "     "*"
```

Ahora que sabemos cuáles son las características más recomendables para realizar modelos, vamos a construir una serie de ellos con algunas de estas características y validaremos con el conjunto de test para comprobar los errores que reflejan.

```
#############################################################
# ESTO ES UNA CHAPUZA Y NO SABEMOS SI HABRÁ QUE HACERLO ASÍ Y/O AQUÍ
#############################################################

sudafricaTrans = preProcess(sudafrica_test[,-ncol(sudafrica_test)], method = c("BoxCox", "center", "sca
sudafrica_test[,-ncol(sudafrica_test)] =predict(sudafricaTrans,sudafrica_test[,-ncol(sudafrica_test)])
```

Para empezar calculamos un modelo lineal de forma que predecimos chd (etiquetas) a partir del atributo más representativo, en nuestro caso como hemos comprobado 'age'.

Una vez calculado el modelo, empleamos la función predict para obtener la probabilidad de cada etiqueta. Como en nuestro caso

```
m1_sudafrica = lm(chd ~ age, data=sudafrica_train)

prob_test_m1sud = predict(m1_sudafrica, data.frame(sudafrica_test[,-ncol(sudafrica_test)]), type="respor

pred_test_m1sud = rep(0, length(prob_test_m1sud))
 # predicciones por defecto 0
pred_test_m1sud[prob_test_m1sud >=0.5] = 1
 # >= 0.5 clase 1
table(pred_test_m1sud, sudafrica_test[,ncol(sudafrica_test)])
```

```
##
## pred_test_m1sud  0  1
##               0 75 33
##               1 10 21
```

```
eout_m1sud = mean(pred_test_m1sud != sudafrica_test[,ncol(sudafrica_test)])
cat("Eval corockn el modelo LR "); print(m1_sudafrica$call)
```

```
## Eval corockn el modelo LR
## lm(formula = chd ~ age, data = sudafrica_train)
```

```
eout_m1sud
```

```
## [1] 0.3093525
```

Obtenemos un error de 0.35, para nada aceptable, por tanto busquemos un modelo diferente empleando otra carasterística, la siguiente más representativa para el cálculo del modelo que en nuestro caso es famhist:

```
m1_sudafrica = lm(chd ~ famhist + age, data=sudafrica_train)

prob_test_m1sud = predict(m1_sudafrica, data.frame(sudafrica_test[,-ncol(sudafrica_test)]), type="respor

pred_test_m1sud = rep(0, length(prob_test_m1sud))
 # predicciones por defecto 0
pred_test_m1sud[prob_test_m1sud >=0.5] = 1
 # >= 0.5 clase 1
table(pred_test_m1sud, sudafrica_test[,ncol(sudafrica_test)])
```

```
##
## pred_test_m1sud  0  1
##               0 72 30
##               1 13 24
```

```
eout_m1sud = mean(pred_test_m1sud != sudafrica_test[,ncol(sudafrica_test)])
cat("Eval con el modelo LR "); print(m1_sudafrica$call)
```

```
## Eval con el modelo LR
## lm(formula = chd ~ famhist + age, data = sudafrica_train)
```

```
eout_m1sud
```

```
## [1] 0.3093525
```