Proyecto Final. Cardiotocography

Anabel Gómez Ríos y Gustavo Rivas Gervilla

16 de junio de 2016

```
library(caret) # para qué?
```

```
## Warning: package 'caret' was built under R version 3.2.5
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.2.5
```

1. Definición del problema a resolver y enfoque elegido.

En este proyecto vamos a trabajar con una base da datos algo mayor que las que hemos venido usando en las prácticas (2126 instancias con 23 atributos cada una) con el objetivo de poner en práctica los conocimientos adquiridos en la asignatura para resolver un problema de clasificación del mundo real.

La base de datos elegida es Cardiotocography del respositorio de bases de datos UCI la cual la podemos descargar **aquí**. En esta base de datos se recogen distintas características de cardiotrogafías en las cuales se mide la frencuencia cardiaca fetal (FHR), los movimiento fetales (FM) y las contracciones uterinas (UC), obteniendo las siguientes características a partir de estos datos:

1. LB: punto de referencia del FHR en pulsaciones por minuto.
2. AC: aceleraciones del pulso por segundo.
3. FM: movimientos fetales por segundo.
4. UC: contracciones uterinas por segundo.
5. DL: deceleraciones suaves por segundo.
6. DS: deceleraciones fuertes por segundo.
7. DP: deceleraciones prolongadas por segundo.
8. ASTV: porcentaje de tiempo con variaciones anormales cortas del pulso.
9. MSTV: media de las variaciones anormales cortas del pulso.
10. ALTV: porcentaje de tiempo con variaciones anormales largas del pulso.
11. MLTV: media de las variaciones anormales largas del pulso.
12. Width: amplitud del histograma FHR.
13. Min: mínimo del histograma FHR.
14. Max: máximo del hisotograma FHR.

15. Nmax: número de picos en el histograma.
16. Nzeros: número de ceros en el histograma.
17. Mode: moda del histograma.
18. Mean: media del histograma.
19. Median: mediana del histograma.
20. Variance: varianza del histograma.
21. Tendency: tendencia del histograma.
22. CLASS: código del tipo de patrón del histograma FHR [1-10].
23. NSP: código del estado fetal. [1: Normal, 2: Sospechoso y 3: Patológico]

Lo que queremos es emplear estos datos para poder predecir ante una nueva cardiotocografía si el estado del feto es normal, sospecho o patológico, es decir, vamos a predecir la variable NSP con el resto. Además, vamos a hacer la clasificación también según la variable CLASS, puesto que también es una de las "preguntas" en la base de datos.

El enfoque elegido por tanto es hacer clasificación multiclase para clasificar nuevos datos según dos variables (por separado), una que tiene 3 clases y otra que tiene 10.

```
datos <- read.csv("datos.csv")
```

2. Codificación de los datos de entranda para hacerlos útiles a los algoritmos.

Nuestra base de datos estaba contenida en una hoja de cálculo. Para poder usarla dentro de R lo que hemos hecho es generar un CSV con los datos previamente formateados puesto que hemos tenido que cambiar el formato decimal de algunas columnas para que fuese el que emplea R. Además en el fichero original aparecían más variables como la fehca y el tiempo de inicio y fin de la cardiotocografía las cuales no hemos considerado relevantes para el estudio por lo que no están presententes en el CSV.

3. Valoración del interés de las variables para el problema y selección de un subconjunto en su caso.

En primer lugar tenemos que `Width` se calcula como la diferencia entre `Max` y `Min` con lo cual suponemos que una de las tres no tendrán relevancia ya que la información aportada por ella se puede deducir de las otras dos.

Para el resto de variables dado el poco conocimiento que tenemos en la materia no podemos saber qué factores son los que más influyen en determinar el estado del feto por tanto hemos decidido realizar un análisis de componentes principales para ver si podemos reducir el número de variables a considerar, haciendo que los algoritmos sean mÃ¡s eficientes en tiempo. La técnica que hemos usado en clase para tal propósito ha sido emplear el Lasso para obtener aquellas variables que sus coeficientes estuviesen por encima de un cierto umbral determinado por nosotros. Esto precisamente es lo que nos ha llevado a decantarnos por el PCA ya que con él podemos saber el conjunto de variables que son capaces de explicar al menos 95% de la variabilidad de los datos (aunque podemos cambiar este 95% y aumentarlo para que sea más estricto). Para saber cómo emplear PCA en R hemos consultado el enlace [2] de la bibliografía.

Lo primero que vamos a hacer es separar los datos en las muestras de entrenamiento y test (80-20) que emplearemos a lo largo de todo el estudio. En esta ocasión como la variable a predecir no depende de la media de otras variables entonces vamos a poder realizar un particionado homogéneo de los datos para tener una distribución de las clases de cada muestra lo más uniforme posible (no corremos el riesgo de contaminar la variable con datos de test como un ocurría en prácticas).

```
# Cogemos los índices del 80% de los datos para cada clase
train_idx <- c(sample(which(datos$NSP == 1), size =
                        ceiling(0.8*sum(datos$NSP==1))),
               sample(which(datos$NSP == 2), size =
                        ceiling(0.8*sum(datos$NSP==2))),
               sample(which(datos$NSP == 3), size =
                        ceiling(0.8*sum(datos$NSP==3))))

# Hacemos el conjunto de train con estos índices
train <- datos[train_idx,]
# Hacemos el conjunto de test con todas aquellas variables que no tengan estos
# índices
test <- datos[-train_idx,]
```

Ahora vamos a quitar las variables `NSP` y `CLASS` de `train` y test, ya que son las salidas, y las vamos a guardar en dos vectores aparte.

```
NSP.train <- train$NSP
CLASS.train <- train$CLASS
train <- train[,-c(22,23)]
NSP.test <- test$NSP
CLASS.test <- test$CLASS
test <- test[,-c(22,23)]
```

Vamos a hacer un `summary` sobre los datos de `train` para ver si podemos descartar alguna variable que a simple vista se vea que no va a aportar nada.

```
summary(train)
```

```
##        LB              AC                  FM                UC
##  Min.   :106.0   Min.   :0.000000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:126.0   1st Qu.:0.000000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :133.0   Median :0.000000   Median :0.00000   Median :0.00000
##  Mean   :133.4   Mean   :0.002928   Mean   :0.00893   Mean   :0.00425
##  3rd Qu.:140.0   3rd Qu.:0.010000   3rd Qu.:0.00000   3rd Qu.:0.01000
##  Max.   :160.0   Max.   :0.020000   Max.   :0.48000   Max.   :0.01000
##        DL                  DS          DP                  ASTV
##  Min.   :0.000000   Min.   :0   Min.   :0.000e+00   Min.   :12.00
##  1st Qu.:0.000000   1st Qu.:0   1st Qu.:0.000e+00   1st Qu.:32.00
##  Median :0.000000   Median :0   Median :0.000e+00   Median :48.00
##  Mean   :0.001534   Mean   :0   Mean   :5.879e-06   Mean   :46.92
##  3rd Qu.:0.000000   3rd Qu.:0   3rd Qu.:0.000e+00   3rd Qu.:61.00
##  Max.   :0.020000   Max.   :0   Max.   :1.000e-02   Max.   :87.00
##       MSTV             ALTV              MLTV            Width
##  Min.   :0.200   Min.   : 0.000   Min.   : 0.000   Min.   :  3.00
##  1st Qu.:0.700   1st Qu.: 0.000   1st Qu.: 4.500   1st Qu.: 37.00
##  Median :1.200   Median : 0.000   Median : 7.500   Median : 67.00
```

```
##  Mean   :1.332   Mean   : 9.982   Mean   : 8.195   Mean   : 70.23
##  3rd Qu.:1.700   3rd Qu.:11.000   3rd Qu.:10.900   3rd Qu.: 99.00
##  Max.   :7.000   Max.   :91.000   Max.   :50.700   Max.   :180.00
##       Min              Max             Nmax            Nzeros
##  Min.   : 50.00   Min.   :122.0   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 67.00   1st Qu.:152.0   1st Qu.: 2.000   1st Qu.: 0.000
##  Median : 94.00   Median :162.0   Median : 3.000   Median : 0.000
##  Mean   : 93.84   Mean   :164.1   Mean   : 4.051   Mean   : 0.321
##  3rd Qu.:120.00   3rd Qu.:174.0   3rd Qu.: 6.000   3rd Qu.: 0.000
##  Max.   :159.00   Max.   :238.0   Max.   :18.000   Max.   :10.000
##       Mode            Mean           Median         Variance
##  Min.   : 60.0   Min.   : 73.0   Min.   : 77.0   Min.   :  0.00
##  1st Qu.:129.0   1st Qu.:125.0   1st Qu.:129.0   1st Qu.:  2.00
##  Median :139.0   Median :136.0   Median :140.0   Median :  7.00
##  Mean   :137.7   Mean   :134.9   Mean   :138.3   Mean   : 18.56
##  3rd Qu.:148.0   3rd Qu.:146.0   3rd Qu.:148.0   3rd Qu.: 24.00
##  Max.   :187.0   Max.   :182.0   Max.   :186.0   Max.   :269.00
##     Tendency
##  Min.   :-1.0000
##  1st Qu.: 0.0000
##  Median : 0.0000
##  Mean   : 0.3263
##  3rd Qu.: 1.0000
##  Max.   : 1.0000
```

Como vemos, la variable DS tiene máximo y mínimo 0, con lo que es igual a 0 para todas las variables y por tanto no van a infuir para nuestro análisis en el conjunto de train. Lo que hacemos por tanto es quitarla de dicho conjunto.

```
# Quitamos la variable DS, que ocupa la sexta columna
train <- train[,-6]
test <- test[,-6] # ESTO NO SABEMOS SI SE PUEDE HACER
```

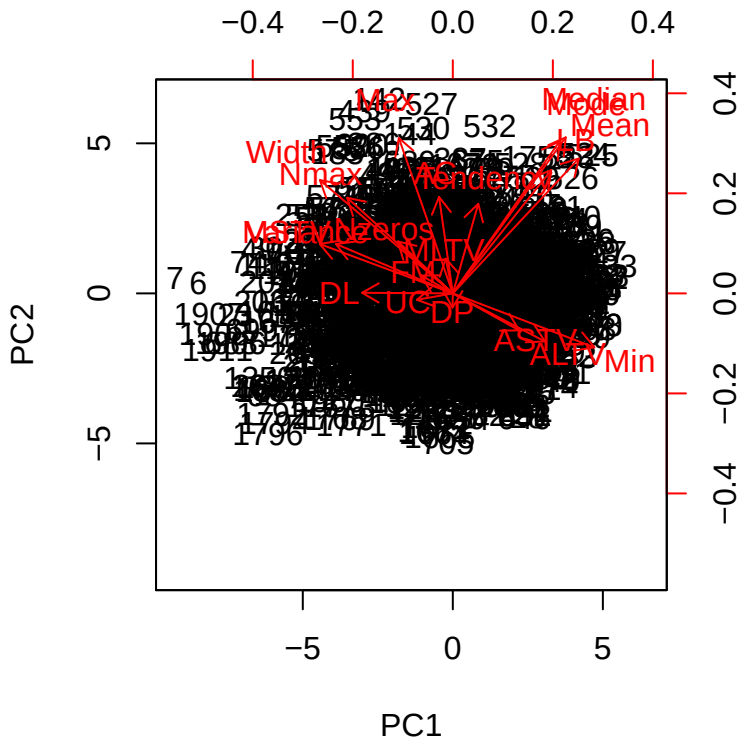
Como hemos podido ver hay muchas que tienen valores cercanos a cero, pero sobre estos no podemos decir nada en claro, así que vamos a pasar a utilizar el algoritmo PCA. Para ello vamos a utilizar la función `prcomp` del paquete `stats` instalado por defecto en `R`.

AQUÍ HAY QUE EXPLICAR POR QUÉ HAY QUE ESCALAR Y CENTRAR POR TEMAS.

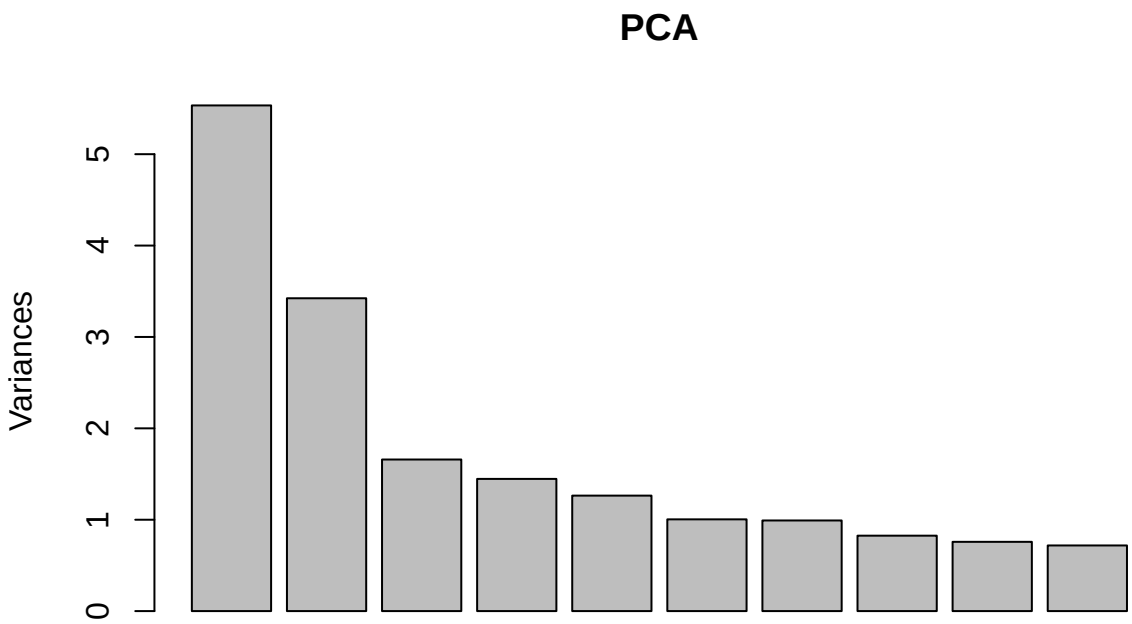
```
pca.out <- prcomp(train, center = TRUE, scale = TRUE)
```

VENDER HUMO

```
biplot(pca.out, scale = 0)
```




```
plot(pca.out, main="PCA")
```




```
summary(pca.out)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6
## Standard deviation     2.3523 1.8502 1.28794 1.20267 1.12405 1.00191
## Proportion of Variance 0.2767 0.1712 0.08294 0.07232 0.06317 0.05019
## Cumulative Proportion  0.2767 0.4478 0.53078 0.60310 0.66627 0.71647
##                           PC7     PC8     PC9   PC10   PC11    PC12
## Standard deviation     0.99578 0.90838 0.87055 0.8474 0.7099 0.67980
## Proportion of Variance 0.04958 0.04126 0.03789 0.0359 0.0252 0.02311
## Cumulative Proportion  0.76604 0.80730 0.84520 0.8811 0.9063 0.92940
##                           PC13    PC14    PC15    PC16    PC17    PC18
## Standard deviation     0.61602 0.58231 0.53273 0.42540 0.36566 0.26163
## Proportion of Variance 0.01897 0.01695 0.01419 0.00905 0.00669 0.00342
## Cumulative Proportion  0.94837 0.96533 0.97952 0.98857 0.99525 0.99867
##                           PC19      PC20
## Standard deviation     0.16288 1.076e-15
## Proportion of Variance 0.00133 0.000e+00
## Cumulative Proportion  1.00000 1.000e+00
```

Como podemos ver con `summary()`, con las 14 primeras componentes principales estamos explicando un 96% de los datos, y son con las que nos vamos a quedar para hacer el estudio reducido y ver si hay mejora al utilizar PCA. Vamos a hacer entonces la combinación lineal que nos da PCA para obtener el nuevo conjunto de train:

```
trainPCA <- apply(pca.out$rotation, 2, function(x) {
  apply(train, 1, function(y) {
    x%*%y
  })
})
```

Vamos a hacerle la combinación lineal al conjunto de test también con las componentes principales de train:

```
testPCA <- apply(pca.out$rotation, 2, function(x) {
  apply(test, 1, function(y) {
    x%*%y
  })
})
```

Bibliografía

1. La base de datos: https://archive.ics.uci.edu/ml/datasets/Cardiotocography#
2. PCA con 'R': http://www.r-bloggers.com/computing-and-visualizing-pca-in-r/
3. Partición de los datos: http://stackoverflow.com/questions/...