Trabajo Final Introducción a la Ciencia de Datos

Juanjo Sierra

27 de diciembre de 2018

Planteamiento

El trabajo final de la asignatura *Introducción a la Ciencia de Datos* se divide en dos secciones. Consiste en realizar un estudio sobre un conjunto de datos de regresión y otro sobre un conjunto de datos de clasificación. Se aplicarán distintas técnicas aprendidas durante la asignatura para conseguir los resultados adecuados.

Librerías y paquetes a cargar

```
library(ggplot2)
library(dplyr)
```

```
##
## 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(tidyr)
```

Trabajo Final Regresión

En primer lugar se realizará el estudio de la base de datos de regresión. En este caso el conjunto de datos a analizar es **Friedman**, que se ha descargado desde el repositorio de datasets de la asignatura. Se puede leer utilizando la siguiente orden:

```
friedman = read.csv("Datos/friedman/friedman.dat", header = FALSE, comment.char = "@")
head(friedman)
```

```
##          V1        V2        V3         V4         V5       V6
## 1 0.6964817 0.3584375 0.4258343 0.33031373 0.22249090 11.09496
## 2 0.5903899 0.4306749 0.8690418 0.07091161 0.63430253 13.22921
## 3 0.8276557 0.6178330 0.9494409 0.67013843 0.64080838 25.33973
## 4 0.8107169 0.2621162 0.4541944 0.85470608 0.27976951 15.18159
## 5 0.4068430 0.8161745 0.8611055 0.12890196 0.15747881 14.43310
## 6 0.6299940 0.3821170 0.9819543 0.98471273 0.07506318 20.97857
```

Dado que los nombres asignados a las variables no aportan ninguna información, y en el resumen del dataset en formato KEEL podemos comprobar que sus nombres tampoco son representativos, se procede a asignarles una notación genérica.

```
n = length(names(friedman))-1
names(friedman)[1:n] = paste ("X", 1:n, sep="")
names(friedman)[n+1] = "Y"
head(friedman)
```

```
##          X1        X2        X3         X4         X5        Y
## 1 0.6964817 0.3584375 0.4258343 0.33031373 0.22249090 11.09496
## 2 0.5903899 0.4306749 0.8690418 0.07091161 0.63430253 13.22921
## 3 0.8276557 0.6178330 0.9494409 0.67013843 0.64080838 25.33973
## 4 0.8107169 0.2621162 0.4541944 0.85470608 0.27976951 15.18159
## 5 0.4068430 0.8161745 0.8611055 0.12890196 0.15747881 14.43310
## 6 0.6299940 0.3821170 0.9819543 0.98471273 0.07506318 20.97857
```

Ahora podemos comprobar de forma más directa que existen 5 variables de entrada (X1-5) que determinan una única variable de salida (Y). Es interesante comprobar las dimensiones del dataset para poder asegurar que se está asumiendo lo correcto.

```
dim(friedman)
```

```
## [1] 1200    6
```

Con esto se puede confirmar que existen un total de 1200 ejemplos en el conjunto de datos, cada uno con 6 variables (5 de entrada y 1 de salida).

Se puede comprobar también si existen valores perdidos en el dataset. Para esto se puede utilizar la función `anyNA`:

```
anyNA(friedman)
```

```
## [1] FALSE
```

Este resultado indica que no hay valores perdidos y que por lo tanto no es necesario imputar ni tomar ninguna decisión para restablecer dichos valores.

A continuación se puede comprobar si existen ejemplos duplicados, y eliminarlos del dataset. Para ello se utiliza la función `duplicated` acompañado de la función `any`:

```
any(duplicated(friedman))
```

```
## [1] FALSE
```

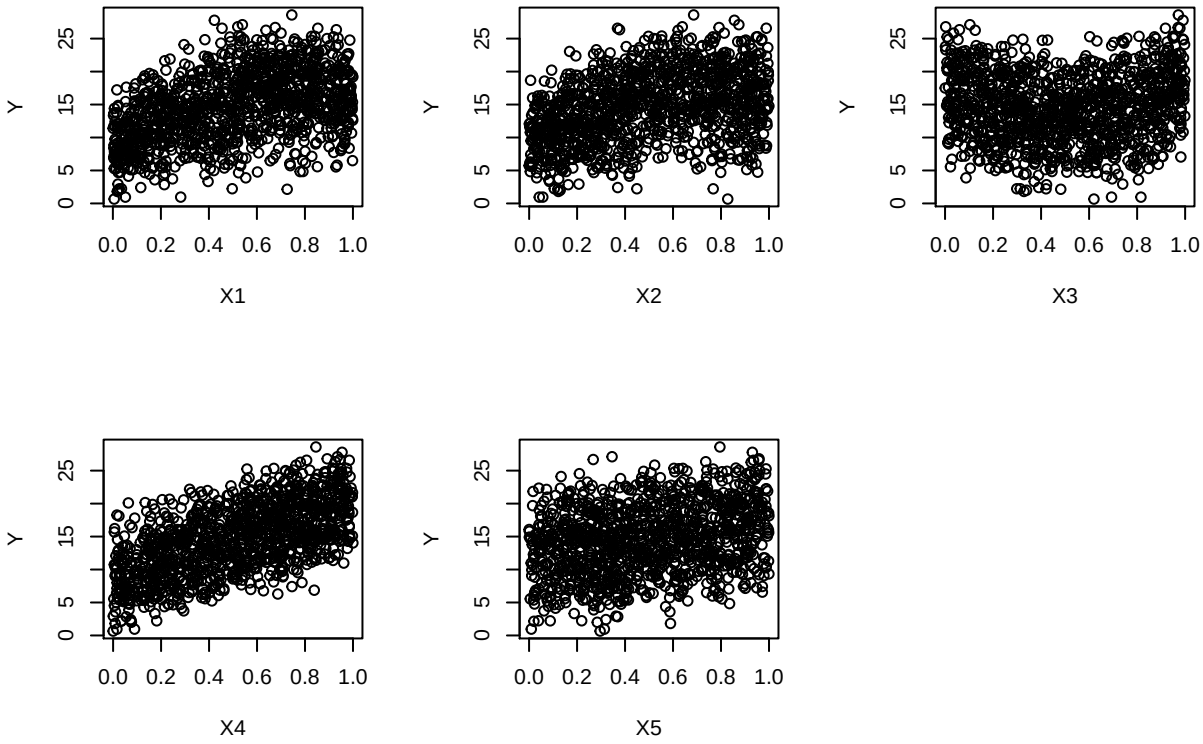
Como no hay duplicados se puede continuar con el estudio sin realizar ninguna alteración en el conjunto de datos.

Además, como el rango de las variables está entre 0 y 1, no es necesario realizar un escalado ni una transformación en los valores. En este punto se puede afirmar que los datos están listos para poder trabajar con ellos.

Como primer paso para el análisis del dataset se puede mostrar cada una de las variables de entrada con respecto a la variable de salida. Esto permitirá averiguar de un vistazo cuál tiene más potencial de determinar qué valor de salida obtendrá dicho ejemplo.

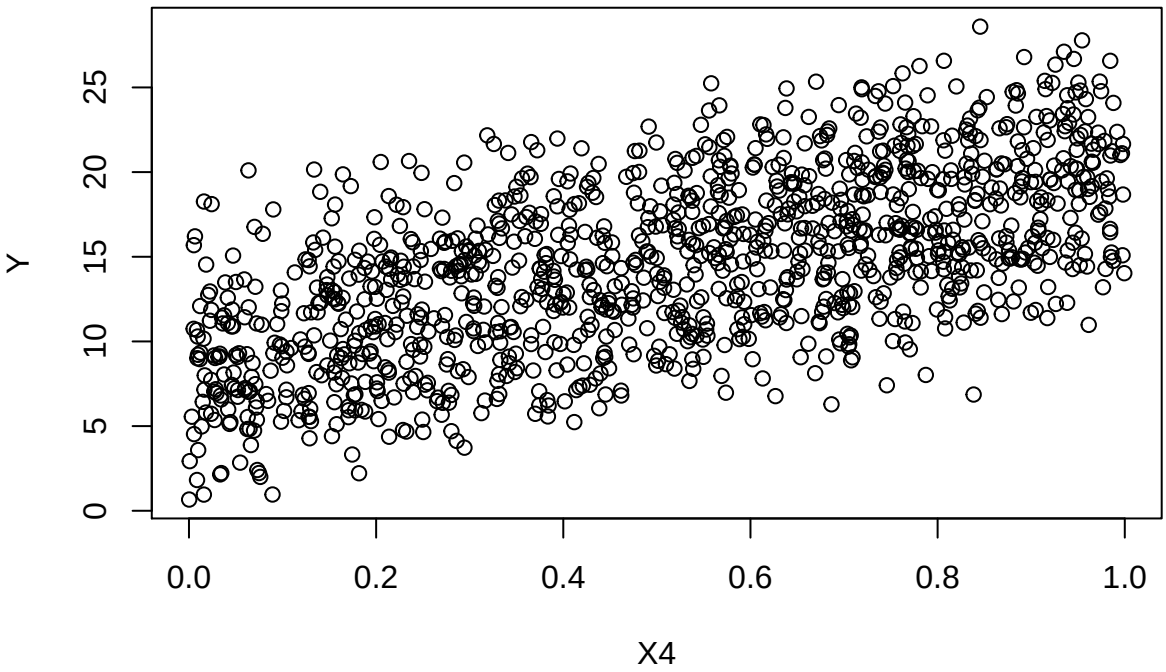
```
plotY = function (x,y) {
  plot(friedman[,y]~friedman[,x], xlab=names(friedman)[x], ylab=names(friedman)[y])
}

par(mfrow=c(2,3))
x = sapply(1:(dim(friedman)[2]-1), plotY, dim(friedman)[2])
par(mfrow=c(1,1))
```



Basándose en un modelo de regresión lineal, se puede especular que la variable X4 parece tener una correlación más alta con la variable de salida Y, y por tanto podría resultar en un mejor modelo. A continuación se muestra la gráfica de X4 frente a Y más grande para poder apreciar mejor la posible correlación.

```
plotY(4,dim(friedman)[2])
```



El primer objetivo del trabajo final de regresión es generar un modelo lineal con cada una de las variables de entrada del dataset. De esta forma se puede obtener de una manera sencilla la información sobre qué variable es mejor para un modelo lineal, es decir, qué variable es más representativa de la de salida.

Para realizar los modelos de regresión lineal se va a utilizar la función `lm` que ya viene entre las funciones base de R. Es necesario indicar cuál es la variable de salida y cuál (o cuáles) son las que se van a utilizar para construir el modelo.

Se analiza en primer lugar el modelo que utiliza la variable X1. Con la función `summary` se obtiene una información más detallada del modelo resultante.

```
lmsimple1 = lm(Y~X1, data=friedman)
summary(lmsimple1)
```

```
##
## Call:
## lm(formula = Y ~ X1, data = friedman)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.1161  -3.3974   0.0156   3.3261  13.8565
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.6586     0.2708   39.37   <2e-16 ***
## X1            7.7211     0.4637   16.65   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.674 on 1198 degrees of freedom
## Multiple R-squared:  0.1879, Adjusted R-squared:  0.1872
## F-statistic: 277.2 on 1 and 1198 DF,  p-value: < 2.2e-16
```