

R-PL3

Gabriel López, Sergio Sanz, Álvaro Zamorano

23 de octubre de 2019

1. Ejercicio realizado en clase.

Para obtener la **función de clasificación** mediante el algoritmo construcción de **árboles de decisión de Hunt** es necesario usar los paquetes **rpart** y **tree**. Estos paquetes hay que descargarlos desde la página de CRAN y para instalarlos hay que ejecutar el siguiente código:

```
> install.packages("./Paquetes/rpart_4.1-15.zip")  
  
package 'rpart' successfully unpacked and MD5 sums checked  
  
> install.packages("./Paquetes/tree_1.0-40.zip")  
  
package 'tree' successfully unpacked and MD5 sums checked
```

De esta forma, los paquete únicamente estarán instalados. Para poder usarlos es necesario cargarlos:

```
> library(rpart)  
> library(tree)
```

- Con **rpart** obtendremos las particiones recursivas para la clasificación y los árboles de decisión.
- Con **tree**, los árboles de clasificación y regresión.

1.1. Función de clasificación.

Los datos a usar en este primer ejercicio se componen de 9 calificaciones de estudiantes compuestas por Teoría, Laboratorio, Prácticas y Calificación Global.

Para introducir estos datos en el algoritmo a usar es necesario tener un fichero **.txt** con el siguiente aspecto.

Suceso	Teoría	Lab	Prac	Calif
s1	A	A	B	Ap
s2	A	B	D	Ss
s3	D	D	C	Ss
s4	D	D	A	Ss
s5	B	C	B	Ss
s6	C	B	B	Ap
s7	B	B	A	Ap
s8	C	D	C	Ss
s9	B	A	C	Ss

Procedemos a leer dicho fichero .txt mediante el uso de la función *read.table*.

```
> calificaciones<-read.table("./Datos/Calificaciones.txt")
```

Para asegurarnos de que todo irá bien a la hora de realizar la clasificación, convertimos los datos leídos en un **dataframe**.

```
> muestra<-data.frame(calificaciones)
```

Nuestros datos ya se encuentran preparados para aplicarles la función **rpart**. Es importante destacar el uso de *minsplit* ya que disponemos de una muestra con un número muy reducido de datos.

```
> clasificacion<-rpart(Calif~.,data=muestra,method="class",minsplit=1)
> clasificacion
```

```
n= 9
```

```
node), split, n, loss, yval, (yprob)
* denotes terminal node
```

```
1) root 9 3 Ss (0.3333333 0.6666667)
  2) Lab=A,B 5 2 Ap (0.6000000 0.4000000)
    4) Prac=A,B 3 0 Ap (1.0000000 0.0000000) *
    5) Prac=C,D 2 0 Ss (0.0000000 1.0000000) *
  3) Lab=C,D 4 0 Ss (0.0000000 1.0000000) *
```

Por último, se aplica la función **tree** a la clasificación obtenida.

```
> clasificaciontree<-tree(Calif~.,data=muestra,mincut=1,minsize=2)
> clasificaciontree
```

```
node), split, n, deviance, yval, (yprob)
* denotes terminal node
```

```
1) root 9 11.46 Ss ( 0.3333 0.6667 )
```

```

2) Lab: A,B 5 6.73 Ap ( 0.6000 0.4000 )
4) Prac: A,B 3 0.00 Ap ( 1.0000 0.0000 ) *
5) Prac: C,D 2 0.00 Ss ( 0.0000 1.0000 ) *
3) Lab: C,D 4 0.00 Ss ( 0.0000 1.0000 ) *

```

Para mostrar el árbol de clasificación hacemos uso de una función que hemos definido, pero para poder usarla, en primer lugar es necesario instalar el paquete **rpart.plot**.

```

> install.packages("./Paquetes/rpart.plot_3.0.8.zip")

package 'rpart.plot' successfully unpacked and MD5 sums checked

> library(rpart.plot)

```

Dicha función es:

```

> source("Funciones/plotTree.R")
> plotTree

function(tree, ruta) {

  png(paste("./tmp/", ruta, sep=""))

  rpart.plot(tree, box.palette="RdBu", shadow.col="gray", nn=TRUE)

  dev.off()
}

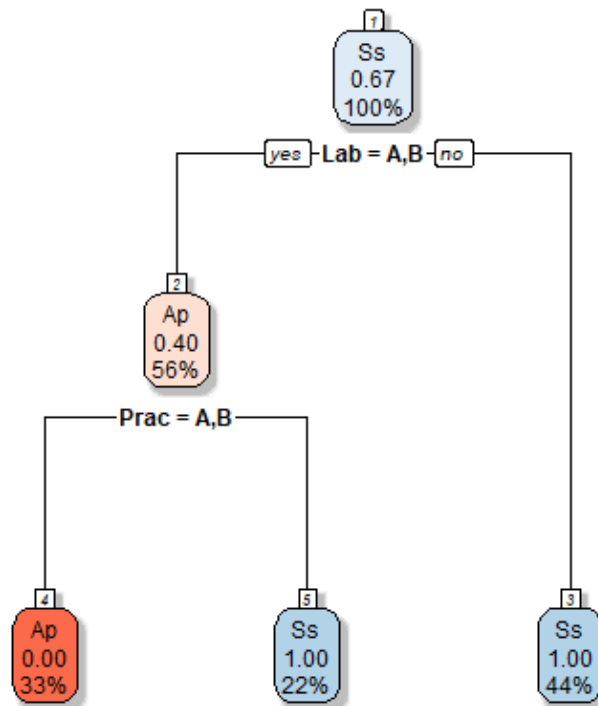
```

Procedemos a su ejecución.

```

> plotTree(clasificacion, "classTree.png")

```



1.2. Análisis de regresión lineal.

En este caso trabajaremos con datos de planetas, en concreto su Radio y su Diámetro. Los planetas de los que se tienen los datos son: Mercurio, Venus, Tierra y Marte, y el .txt del que se leen dichos datos tiene el aspecto que sigue.

Planeta	Radio	Diámetro
Mercurio	2.4	5.4
Venus	6.1	5.2
Tierra	6.4	5.5
Marte	3.4	3.9

Al igual que anteriormente, es necesario leer dicho fichero y pasarlo a dataframe.

```
> planetas<-read.table("./Datos/Planetas.txt")
> muestraP<-data.frame(planetas)
```

El análisis de regresión se hace mediante el uso de la función **lm** contenida en el paquete stats. Cabe destacar que el primero de sus argumentos es de tipo *fórmula* donde una expresión de la forma $y \sim \text{model}$ se interpreta como una especificación de que la respuesta y está modelada por un predictor lineal especificado simbólicamente por model , es decir, en nuestro caso $\text{model}=x$ por lo que su ejecución queda como:

```
> regresionP<-lm(D~R,data=muestraP)
> regresionP
```

Call:

```
lm(formula = D ~ R, data = muestraP)
```

Coefficients:

(Intercept)	R
4.3624	0.1394

De acuerdo a la ecuación de una recta $y=a+b*x$, el primero de los coeficientes es el término independiente (a), y el segundo de ellos la b.

Para mostrar el gráfico de dispersión y la recta de ajuste es necesario hacer uso de varias librerías.

```
> library(foreign)
> library(ggplot2)
> library(psych)
```

Estas librerías se usan en funciones externas usadas para representar los gráficos requeridos.

```
> source("Funciones/plotDisp.R")
> plotDisp(planetas,regresionP,"Radio","Diametro","regPlanetas.png")
```

