PECL3 - Fundamentos de la ciencia de datos

Marcos Barranquero Adrián Montesinos
Eduardo Graván

4 de noviembre de 2019

1. Apartado 1 - Análisis de clasificación supervisada

1.1. Introducción

En este apartado vemos dos ejercicios de clasificación supervisada realizados en clase, donde resolvemos uno con árboles de decisión de Hunt y otro mediante regresión lineal.

1.2. Árboles de decisión

Se pide, para las siguientes calificaciones de la tabla 1, obtener una función de clasificación, utilizando como medida de impureza el Gini a la hora de realizarla.

Cuadro 1: Muestra de calificaciones:

A	A	B	Ap
A	B	D	Ss
D	D	C	Ss
D	D	A	Ss
B	C	B	Ss
C	B	B	Ap
B	B	A	Ap
C	D	C	Ss
B	A	C	Ss

Para realizar el análisis, el primer paso es cargar las librerías de *rpart* y *tree*.

```
> # Cargamos las librerias:
> library('rpart')
> library('tree')
```

Tras esto, leemos el fichero de texto con la información de las calificaciones, y las insertamos en un *dataframe*:

```
> # Trabajo con datos y la libreria rpart
> calificaciones <- read.table("./Parte 1/calificaciones.txt")
> muestra <- data.frame(calificaciones)
```

Ahora podemos clasificar utilizando rpart o utilizando rtree.

Para rpart:

```
> clasificacion <- rpart(Calificacion~., data=muestra, method="class", minsplit=1)
> print(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) Pract=A,B 3 0 Ap (1.0000000 0.0000000) *
    5) Pract=C,D 2 0 Ss (0.0000000 1.0000000) *
  3) Lab=C,D 4 0 Ss (0.0000000 1.0000000) *
```

Observamos la estructura de los nodos en la impresión por pantalla que realiza, y vemos que consigue realizar la clasificación.

Para tree:

```
> calificaciontree <- tree(Calificacion~., data=muestra, mincut=1, minsize=2)
> print(calificaciontree)

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) Pract: A,B 3  0.00 Ap ( 1.0000 0.0000 ) *
    5) Pract: C,D 2  0.00 Ss ( 0.0000 1.0000 ) *
  3) Lab: C,D 4  0.00 Ss ( 0.0000 1.0000 ) *
```

Observamos que para ambas librerías obtenemos el mismo resultado. Consultando la documentación, leemos que ambas librerías emplean por defecto el Gini para el cálculo de impureza, por lo que no es necesario modificar la llamada.

1.3. Regresión lineal

Se pide, para los siguientes planetas y radios mostrados en la tabla 2, realizar un análisis de regresión lineal.

Cuadro 2: Planetas con radio y densidad

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

En esta ocasión no tenemos que importar ninguna librería ya que podemos realizar el análsis con las librerías que ya trae por defecto R. Tan solo debemos cargar los datos:

```
> # Cargamos los datos
> planetas <- read.table("./Parte 1/planetas.txt")
```

Y realizar el análisis, especificando que queremos la realación entre la densidad (D) y el radio (R), y obteniendo los coeficientes asociados:

```
> # Hacemos el estudio de la regresion
> regresion = lm(D~R, data=planetas)
> print(regresion)

Call:
lm(formula = D ~ R, data = planetas)

Coefficients:
(Intercept)            R
     4.3624       0.1394
```

2. Apartado 2 - Ejercicios de clasificación

2.1. Ejercicio - Clasificación de vehículos

Se pide, dada la siguiente tabla 3,de características de vehículos , realizar una función clasificadora mediante áboles de decisión.

Cuadro 3: Vehículos y características

B	4	5	Coche
A	2	2	Moto
N	2	1	Bicicleta
B	6	4	Camión
B	4	6	Coche
B	4	4	Coche
N	2	2	Bicicleta
B	2	1	Moto
B	6	2	Camión
N	2	1	Bicicleta

Para resolverlo, empleamos una aproximación similar al ejercicio anterior de clasificación:

```
> # Leemos el archivo de texto y lo sacamos por pantalla
> datos <- read.table("./Parte 2/datos21.txt")
> # Convertimos los datos leidos en un data frame
> muestra <- data.frame(datos)
> # Clasificamos usando la libreria rpart, que usa el metodo Gini por defecto
> clasificacion <- rpart(TipoVehiculo~., data=muestra, method="class", minsplit=1)
> print(clasificacion)

n= 10

node), split, n, loss, yval, (yprob)
```

```
      * denotes terminal node

 1) root 10 7 Bicicleta (0.3000000 0.2000000 0.3000000 0.2000000)
   2) TipoCarnet=N 3 0 Bicicleta (1.0000000 0.0000000 0.0000000 0.0000000) *
   3) TipoCarnet=A,B 7 4 Coche (0.0000000 0.2857143 0.4285714 0.2857143)
     6) NumRuedas>=3 5 2 Coche (0.0000000 0.4000000 0.6000000 0.0000000)
      12) NumRuedas>=5 2 0 Camion (0.0000000 1.0000000 0.0000000 0.0000000) *
      13) NumRuedas< 5 3 0 Coche (0.0000000 0.0000000 1.0000000 0.0000000) *
     7) NumRuedas< 3 2 0 Moto (0.0000000 0.0000000 0.0000000 1.0000000) *
```

Además, hemos añadido un plot para poder visualizar el árbol:

```
> # Pintamos el arbol de clasificacion para este ejercicio
> plot(clasificacion, uniform=TRUE, main="Arbol de clasificacion para los vehiculos")
> text(clasificacion, use.n=TRUE, all=TRUE, cex=.7, fancy=TRUE, fwidth=0.5, fheight=0.7)
```

Obteniendo el siguiente gráfico:

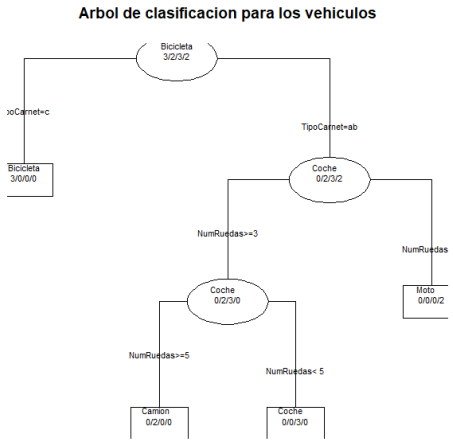

Figura 1: Árbol de decisión

2.2. Ejercicio - Regresión

Se pide, para las siguientes muestras, realizar un análisis de regresión.

2.2.1. Muestra 1

Se tiene la siguiente muestra de pares de datos:
Se resuelve así:

```
> # Cargamos los datos y los dividimos en 4 muestras
> datos <- read.table("./Parte 2/datos22.txt", header=TRUE)
> muestras <- split(datos, factor(sort(rank(row.names(datos))%%4)))
> cat("Muestra 1: \n\n")
```

Cuadro 4: Muestra 1

10	8.04
8	6.95
13	7.58
9	8.81
11	8.33
14	9.96
6	7.24
4	4.26
12	10.84
7	4.82
5	5.68

```
Muestra 1:
```

```
> regresion1 = lm(Y~X, data=muestras[[1]])
> print(regresion1)
```

```
Call:
lm(formula = Y ~ X, data = muestras[[1]])

Coefficients:
(Intercept)            X
     3.0001       0.5001
```

2.2.2. Muestra 2

Se tiene la siguiente muestra de pares de datos:

Cuadro 5: Muestra 2

10	9.14
8	8.14
13	8.74
9	8.77
11	9.26
14	8.1
6	6.13
4	3.1
12	9.13
7	7.26
5	5.74

Se resuelve así:

```
> # Cargamos los datos y los dividimos en 4 muestras
> datos <- read.table("./Parte 2/datos22.txt", header=TRUE)
> muestras <- split(datos, factor(sort(rank(row.names(datos))%%4)))
> cat("\nMuestra 2: \n\n")
```

```
Muestra 2:
```

```
> regresion2 = lm(Y~X, data=muestras[[2]])
> print(regresion2)

Call:
lm(formula = Y ~ X, data = muestras[[2]])

Coefficients:
(Intercept)            X
      3.001        0.500
```

2.2.3. Muestra 3

Se tiene la siguiente muestra de pares de datos:

Cuadro 6: Muestra 3

10	7.46
8	6.77
13	12.74
9	7.11
11	7.81
14	8.84
6	6.08
4	5.39
12	8.15
7	6.42
5	5.73

Se resuelve así:

```
> # Cargamos los datos y los dividimos en 4 muestras
> datos <- read.table("./Parte 2/datos22.txt", header=TRUE)
> muestras <- split(datos, factor(sort(rank(row.names(datos))%%4)))
> cat("\nMuestra 3: \n\n")

Muestra 3:

> regresion3 = lm(Y~X, data=muestras[[3]])
> print(regresion3)

Call:
lm(formula = Y ~ X, data = muestras[[3]])

Coefficients:
(Intercept)            X
     3.0025       0.4997
```

2.2.4. Muestra 3

Se tiene la siguiente muestra de pares de datos:
Se resuelve así:

Cuadro 7: Muestra 4

8	6.58
8	5.76
8	7.71
8	8.84
8	8.47
8	7.04
8	5.25
19	12.5
8	5.56
8	7.91
8	6.89

```
> # Cargamos los datos y los dividimos en 4 muestras
> datos <- read.table("./Parte 2/datos22.txt", header=TRUE)
> muestras <- split(datos, factor(sort(rank(row.names(datos))%%4)))
> cat("\nMuestra 4: \n\n")

Muestra 4:

> regresion4 = lm(Y~X, data=muestras[[4]])
> print(regresion4)

Call:
lm(formula = Y ~ X, data = muestras[[4]])

Coefficients:
(Intercept)            X
     3.0017       0.4999
```

Finalmente, hemos realizado la gráfica para poder visualizar la regresión:

```
> # Pintamos las graficas resultantes de las regresiones
> par(mfrow=c(2,2))
> plot(muestras[[1]]$X, muestras[[1]]$Y, main="Muestra 1", xlab="x", ylab="y")
> abline(regresion1, col="red")
> plot(muestras[[2]]$X, muestras[[2]]$Y, main="Muestra 2", xlab="x", ylab="y")
> abline(regresion2, col="red")
> plot(muestras[[3]]$X, muestras[[3]]$Y, main="Muestra 3", xlab="x", ylab="y")
> abline(regresion3, col="red")
> plot(muestras[[4]]$X, muestras[[4]]$Y, main="Muestra 4", xlab="x", ylab="y")
> abline(regresion4, col="red")
> mtext(expression(paste(bold("Comparativa de rectas de regresion de las muestras"))),
+          side = 3, line = -2, outer = TRUE)
```

Generando el siguiente gráfico:

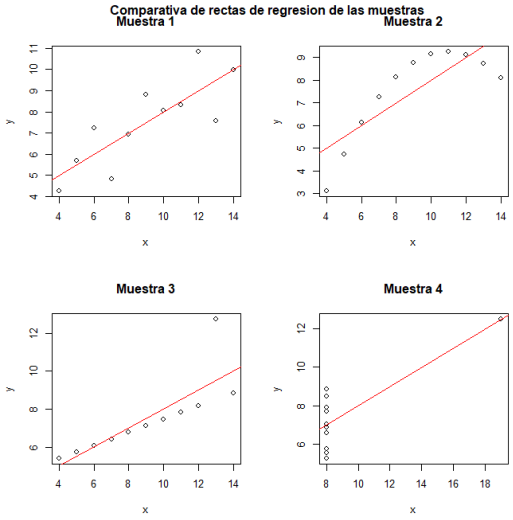

Figura 2: Regresiones de las muestras

2.3. Ejercicio propio

Para esta sección, se ha decidido realizar una clasificiación supervisada mediante Machine Learning. Hemos tomado el ejercicio de clasificación de las notas del primer apartado, ampliando la muestra. El ejercicio consiste en dividir la muestra en set de entrenamiento y set de prueba. Emplearemos el set de entrenamiento para entrenar un clasificador, y, una vez entrenado, le pasaremos el set de prueba para que realice la clasificación. Comprobando la matriz de confusión, el campo *Accuracy* referencia la precisión y exactitud a la hora de clasificar adecuadamente el set de prueba.

En primer lugar, debemos instalar los paquetes adecuados:

```
install.packages("rpart")
install.packages("dplyr")
install.packages("caret")
install.packages("e1071") # Dependencia oculta de 'caret'
```

Una vez los paquetes instalados y el directorio de trabajo establecido, importamos las librerías y leemos la muestra:

```
> # Cargamos librerías
> library('rpart')
> library('dplyr')
> library('caret')
> # Cargamos la muestra
> tablaNotas <- read.table(
+     "./Parte 2/notas23.txt",
+     col.names = c("Teoria1", "Teoria2", "Laboratorio", "Calificacion")
+ )
> notas <- data.frame(tablaNotas)
```

Dividimos la muestra en set de entrenamiento y prueba. Usamos una seed fija en el generador de números aleatorios para que siempre generemos el mismo documento final.

```
> set.seed(3454)
> notas_entrenamiento <- sample_frac(notas, 0.7)
> notas_prueba <- setdiff(notas, notas_entrenamiento)
```

Entrenamos el clasificador y predecimos el set de prueba:

```
> # Entrenamos el clasificador.
> clasificacion <- rpart(Calificacion~.,
+                        data=notas_entrenamiento, method="class", minsplit=1)
> print(clasificacion)

n= 14

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

```
1) root 14 4 Ss (0.2857143 0.7142857)
  2) Teoria2=A,B 6 2 Ap (0.6666667 0.3333333)
    4) Laboratorio=A,B 4 0 Ap (1.0000000 0.0000000) *
    5) Laboratorio=C,D 2 0 Ss (0.0000000 1.0000000) *
  3) Teoria2=C,D 8 0 Ss (0.0000000 1.0000000) *
```

```
> # Predecimos el set de prueba.
> prediccion <- predict(clasificacion,
+                       newdata=notas_prueba, type="class")
```

Finalmente, comprobamos la matriz de precisión, prestando atención al campo 'Accuracy':

```
> # Comprobamos la matriz de precisión: nos interesa el campo 'Accuracy'.
> confusion = confusionMatrix(table(prediccion, notas_prueba$Calificacion))
> print(confusion)

Confusion Matrix and Statistics


prediccion Ap Ss
        Ap  3  0
        Ss  0  3

               Accuracy : 1
                 95% CI : (0.5407, 1)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : 0.01563

                  Kappa : 1

 Mcnemar's Test P-Value : NA

            Sensitivity : 1.0
            Specificity : 1.0
         Pos Pred Value : 1.0
         Neg Pred Value : 1.0
             Prevalence : 0.5
         Detection Rate : 0.5
   Detection Prevalence : 0.5
      Balanced Accuracy : 1

       'Positive' Class : Ap
```

Observamos que obtenemos un valor de Accuracy de 1, es decir, ha conseguido clasificar todos los elementos del set de pruebas.

3. Conclusiones

Mediante esta práctica hemos empleado herramientas que permiten clasificar elementos partiendo de una muestra. Observamos que existe gran cantidad de soporte y paquetes por parte de la comunidad para realizar este tipo de análisis de datos. Contemplamos también la facilidad de uso a la hora de realizar estos análisis, y la extensión y potencia de estos análisis.