Árboles de clasificación

Contents

1 Datos

En este ejemplo vamos a analizar los datos de 150 lirios utilizando árboles de clasificación. Estos datos los podemos encontrar en el paquete *datasets* de R. Para más información podemos teclear *help("iris")* en la consola. Entre otras cosas se obtiene:

This famous (Fisher's or Anderson's) iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are Iris setosa, versicolor, and virginica.

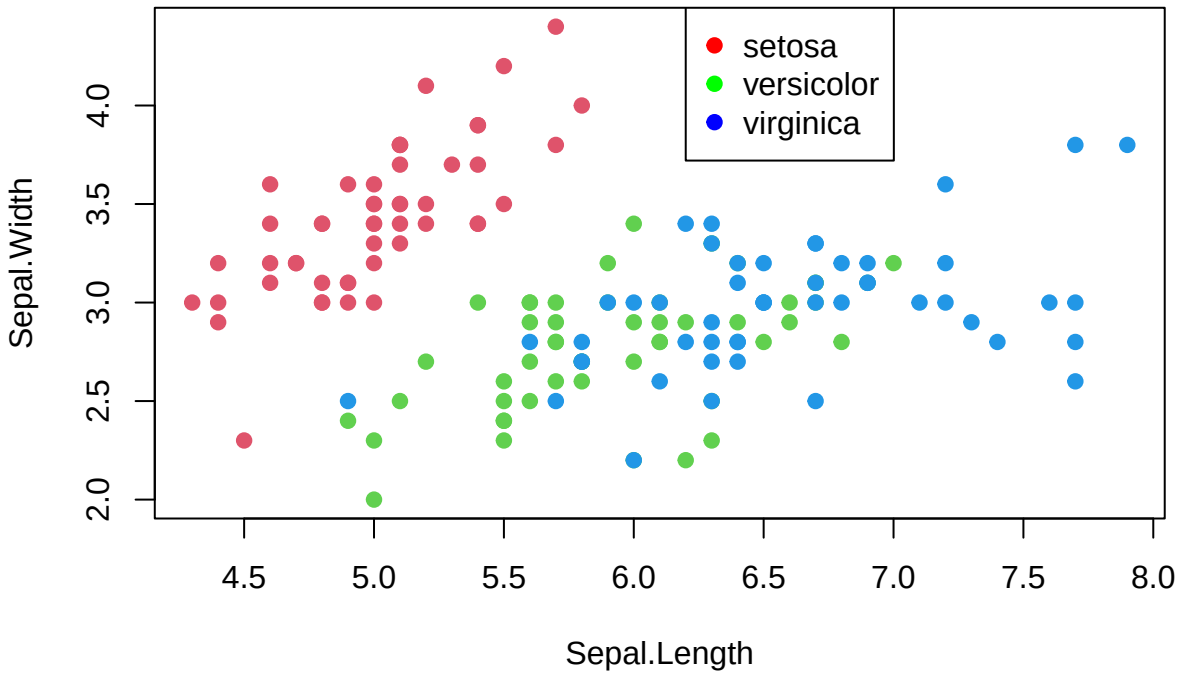
Podemos encontrar más información en https://en.wikipedia.org/wiki/Iris_flower_data_set, como por ejemplo fotos de Iris setosa, versicolor, y virginica.

Para ver el contenido del dataset:

```
d = iris
str(d)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
plot(d$Sepal.Length, d$Sepal.Width, pch = 19, col = as.numeric(d$Species)+1, xlab = "Sepal.Length", yla
legend(6.2,4.5,legend=c("setosa","versicolor","virginica"), col=c("red","green","blue"), pch=19)
```



Como vemos, la variable respuesta no es un número, sino que se trata de una variable cualitativa. En este caso tenemos **árboles de clasificación**.

2 Estimación del árbol

Primero vamos a dividir aleatoriamente los datos en training set/validation set:

```
set.seed(1) # utilizamos una semilla para que los resultados sean reproducibles
ndat = nrow(d)
pos_train = sample(1:ndat, ndat/2, replace = F) # la mitad de los datos para entranamiento
datos_train = d[pos_train,] # trainning set
datos_test = d[-pos_train,] # test set
```

Estimamos (o entrenamos) un arbol de clasificación con los datos de entrenamiento:

```
library(rpart)
t1 <- rpart(Species ~ Sepal.Width + Sepal.Length, data = datos_train, method = "class")
```

```
print(t1)
```

```
## n= 75
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 75 47 setosa (0.37333333 0.26666667 0.36000000)
##   2) Sepal.Length< 5.55 31  3 setosa (0.90322581 0.09677419 0.00000000) *
##   3) Sepal.Length>=5.55 44 17 virginica (0.00000000 0.38636364 0.61363636)
##     6) Sepal.Length< 6.15 16  4 versicolor (0.00000000 0.75000000 0.25000000) *
##     7) Sepal.Length>=6.15 28  5 virginica (0.00000000 0.17857143 0.82142857) *
```

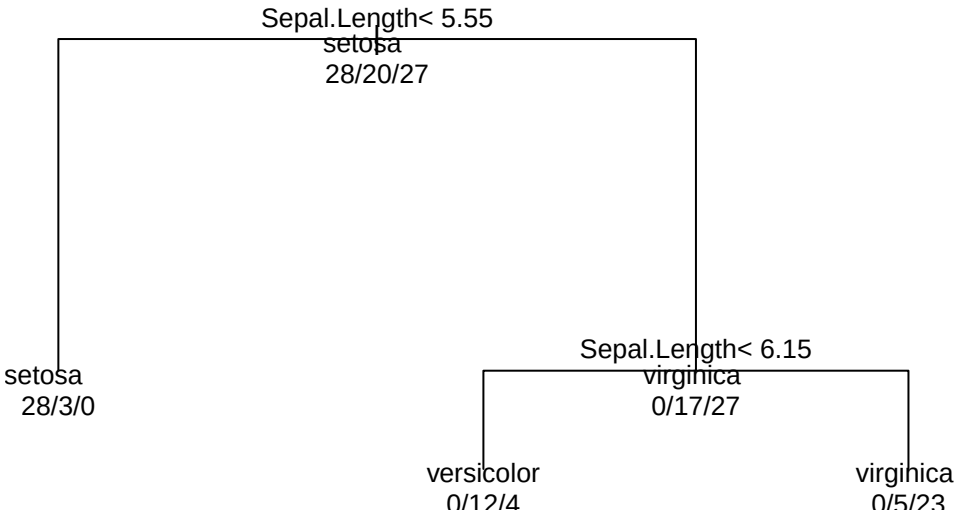
- El arbol de clasificación predice la clase más frecuente.
- Para dividir un nodo se tiene que minimizar el error de clasificación, que en este caso viene dado por el **índice de Gini**:

$$G = \sum_{k=1}^{K} p_k(1 - p_k)$$

- donde K es el numero de clases (3 en nuestro ejemplo) y p_k es la proporción de datos pertenecientes a cada clase. Cuanto más pequeño es G, más homogéneo es un nodo. Por ejemplo, si todos los datos pertenecen a la misma clase, $p_k = 1$, G = 0.
- Por ejemplo, en el nodo 2) de la tabla anterior, $G = 0.9032 * (1 - 0.9032) + 0.0967 * (1 - 0.0967) + 0 = 0.17$.
- En la tabla se devuelve el criterio por el que se divide el nodo, el numero de datos del nodo, el numero de datos mal clasificados (loss), la clase más frecuente (la predicción), y la proporción de datos que hay de cada clase.

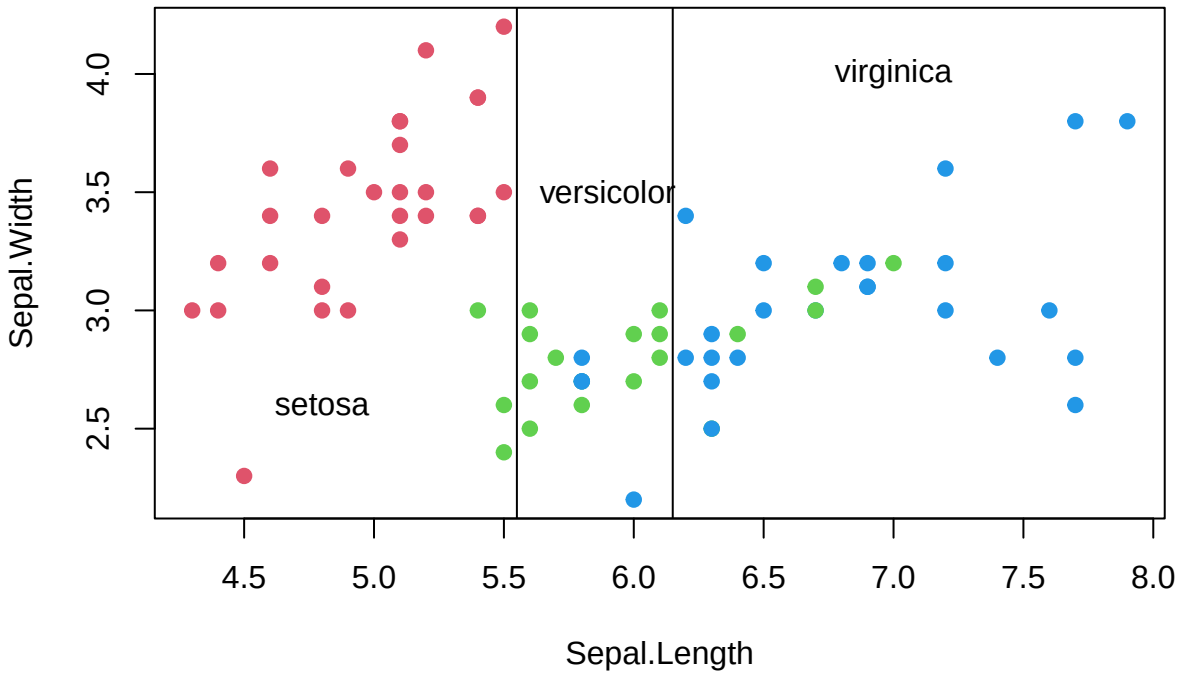
Dibujamos el árbol:

```
plot(t1, margin = 0.05)
text(t1, use.n = T, all = T, cex=0.8)
```



Como sólo tenemos dos regresores, podemos visualizar las particiones que propone el árbol estimado de la siguiente forma:

```
plot(datos_train$Sepal.Length, datos_train$Sepal.Width, pch = 19, col = as.numeric(datos_train$Species)+
abline(v = 5.55)
text(4.8,2.6, labels = "setosa")
abline(v = 6.15)
text(5.9,3.5, labels = "versicolor")
text(7,4, labels = "virginica")
```



Calculamos el error del modelo

```
yt_p = predict(t1, datos_train, type="class") # para arboles de clasificiacion hay que poner type = "cl
# creamos una tabla
table(yt_p, datos_train$Species)
```

```
##
## yt_p         setosa versicolor virginica
##   setosa         28          3         0
##   versicolor      0         12         4
##   virginica       0          5        23
```

La tabla nos indica que se han predicho correctamente $(28 + 12 + 23) = 63$, y que hay 12 datos que no se predicen bien (como se observa en el grafico anterior). Por ejemplo, se han predicho 3 setosa que en realidad eran versicolor.

Ahora vamos a calcular el error cometido en los datos test (error de predicción):

```
yv_p = predict(t1, datos_test, type="class")
# creamos una tabla
table(yv_p, datos_test$Species)
```

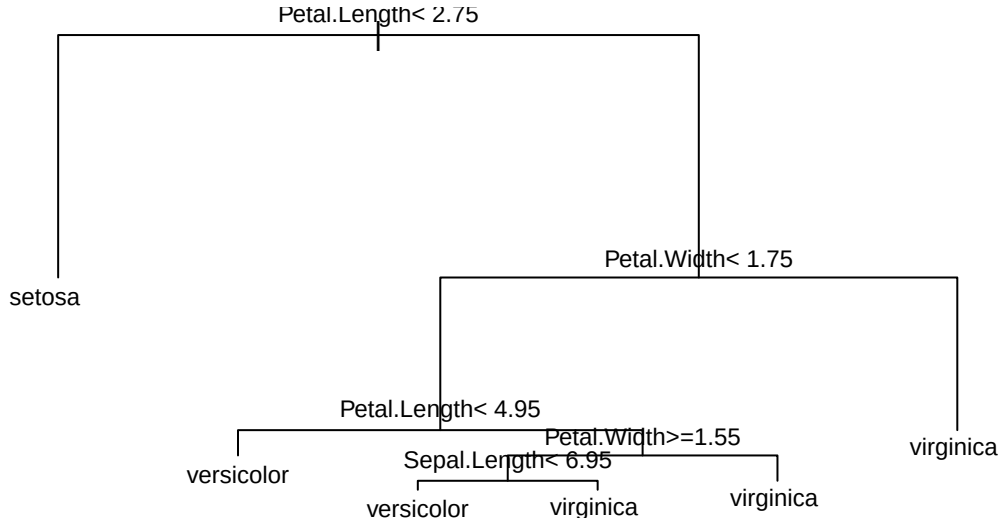
```
##
## yv_p         setosa versicolor virginica
##   setosa         19          8         1
##   versicolor      3         11         6
##   virginica       0         11        16
```

- Los resultados empeoran un poco. Tenemos un error de clasificación igual a $(8+1+6+11+3)/75 = 0.39$

3 Podado

- Utilizamos ahora todos los regresores:

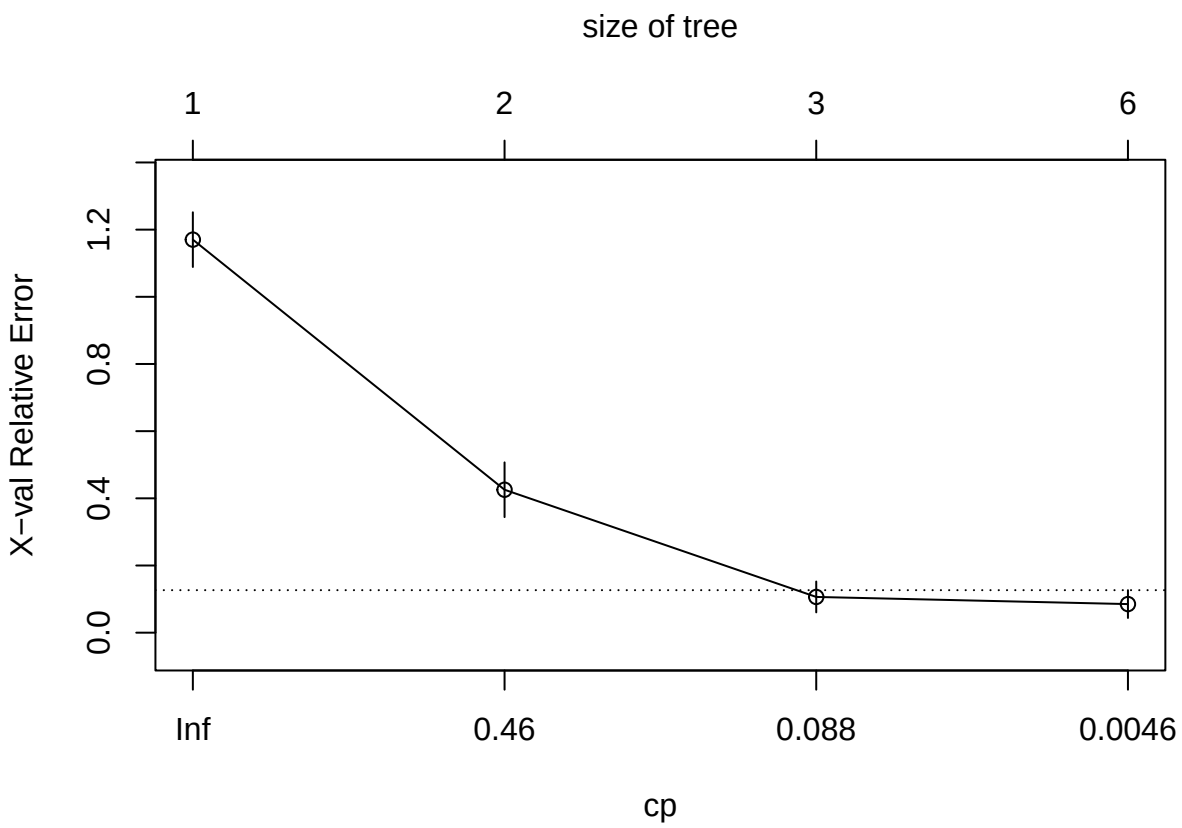
```
t2 <- rpart(Species ~ ., data = datos_train, method = "class",
            control = rpart.control(minsplit = 2, cp = 0.001))
plot(t2, margin = 0.02)
text(t2, cex=.75)
```




```
t2_printcp = printcp(t2) # lo guardamos en una variable para utilizarlo despues
```

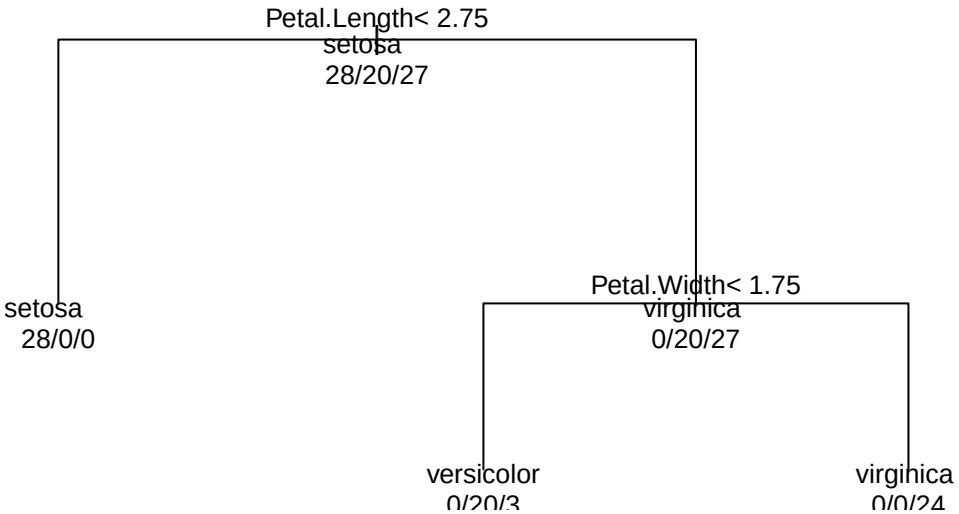
```
##
## Classification tree:
## rpart(formula = Species ~ ., data = datos_train, method = "class",
##     control = rpart.control(minsplit = 2, cp = 0.001))
##
## Variables actually used in tree construction:
## [1] Petal.Length Petal.Width  Sepal.Length
##
## Root node error: 47/75 = 0.62667
##
## n= 75
##
##         CP nsplit rel error   xerror     xstd
## 1 0.574468      0   1.00000 1.170213 0.081483
## 2 0.361702      1   0.42553 0.425532 0.081483
## 3 0.021277      2   0.06383 0.106383 0.045963
## 4 0.001000      5   0.00000 0.085106 0.041403
```

```
plotcp(t2)
```



- Ahora podamos el arbol:

```
t2_prune = prune(t2, cp = 0.021277)
plot(t2_prune, margin = 0.05)
text(t2_prune, use.n = T, all = T, cex=0.8)
```



- El error del modelo es:

```
ytrain_p2a = predict(t2_prune, datos_train, type="class") # para arboles de clasificiacion hay que pone
# creamos una tabla
table(ytrain_p2a, datos_train$Species)
```

```
##
```

```
## ytrain_p2a   setosa versicolor virginica
##   setosa         28          0         0
##   versicolor      0         20         3
##   virginica       0          0        24
```

- Error de predicción:

```
ytest_p2a = predict(t2_prune, datos_test, type="class")
# creamos una tabla
table(ytest_p2a, datos_test$Species)
```

```
##
## ytest_p2a    setosa versicolor virginica
##   setosa         22          0         0
##   versicolor      0         29         2
##   virginica       0          1        21
```

4 Random forest

```
library(randomForest)
rf1 = randomForest(Species ~ ., data = datos_train, mtry = 4, ntree = 500) # bagging
```

- error de predicción:

```
ytest_p_rf1 = predict(rf1, datos_test, type="class")
# creamos una tabla
table(ytest_p_rf1, datos_test$Species)
```

```
##
## ytest_p_rf1  setosa versicolor virginica
##   setosa         22          0         0
##   versicolor      0         29         1
##   virginica       0          1        22
```

- Tenemos un error de clasificación igual a $3/75 = 0.04$.