Unidad 4 - Módulo 1

Pablo Anicich

24 de julio de 2016

Ejercicio árbol 2 resuelto

Lo que primero que hago es setear el directorio de trabajo para leer el archivo Arbol2.csv:

```
setwd("C:/Documentos/Diplomatura_en_BI_UTN/Modulo_1/Unidad_4/");
arbol2 <- read.csv("Arbol2.csv",
                   header=TRUE, sep=";", na.strings=c("NA"));
```

Para familiarme con los datos, extraigo la info sobre la estructura de los datos en el dataframe "arbol2".

```
str(arbol2)
```

```
## 'data.frame':    20000 obs. of  7 variables:
##  $ Atributo.1: int  0 0 0 0 0 0 0 0 1 0 ...
##  $ Atributo.2: int  0 0 0 0 0 0 0 0 1 0 ...
##  $ Atributo.3: int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Atributo.4: int  0 0 0 0 0 1 1 0 0 0 ...
##  $ Atributo.5: int  1 0 1 0 1 1 1 1 0 0 ...
##  $ Atributo.6: int  1 1 1 1 1 1 1 1 0 1 ...
##  $ Resultado : int  0 0 0 0 0 0 0 0 1 0 ...
```

Son 7 variables con 20000 observaciones. Todas las variables son enteras. Separo las observaciones de acuerdo a dos grupos:

- Conjunto de entrenamiento: "arbol2.train",
- Conjunto de test: "arbol2.test";

y realizo el correspondiente control para ver que todo haya ido bien:

```
arbol2.train <- arbol2[1:10000,]
arbol2.test <- arbol2[10001:20000,]

str(arbol2.train)
```

```
## 'data.frame':    10000 obs. of  7 variables:
##  $ Atributo.1: int  0 0 0 0 0 0 0 0 1 0 ...
##  $ Atributo.2: int  0 0 0 0 0 0 0 0 1 0 ...
##  $ Atributo.3: int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Atributo.4: int  0 0 0 0 0 1 1 0 0 0 ...
##  $ Atributo.5: int  1 0 1 0 1 1 1 1 0 0 ...
##  $ Atributo.6: int  1 1 1 1 1 1 1 1 0 1 ...
##  $ Resultado : int  0 0 0 0 0 0 0 0 1 0 ...
```

```
str(arbol2.test)
```

```
## 'data.frame':    10000 obs. of  7 variables:
##  $ Atributo.1: int  0 0 0 0 0 0 0 0 1 0 ...
##  $ Atributo.2: int  0 0 0 0 0 0 0 0 1 1 ...
##  $ Atributo.3: int  0 0 1 0 1 0 0 1 1 0 ...
##  $ Atributo.4: int  1 1 1 1 1 1 1 0 0 1 ...
##  $ Atributo.5: int  1 1 1 1 1 1 1 1 0 1 ...
##  $ Atributo.6: int  1 0 1 1 1 1 1 0 0 1 ...
##  $ Resultado : int  0 0 0 0 0 0 0 0 1 0 ...
```

Cargo todos los paquetes necesarios para trabajar con árboles:

```
library(rpart)
suppressWarnings(library(rattle))
```

```
## Rattle: A free graphical interface for data mining with R.
## Versión 4.1.0 Copyright (c) 2006-2015 Togaware Pty Ltd.
## Escriba 'rattle()' para agitar, sacudir y  rotar sus datos.
```

```
suppressWarnings(library(rpart.plot))
suppressWarnings(library(RColorBrewer))
```

Procedo a generar sendos árboles a partir de los dataframes "arbol2.train" y "arbol2.test":

```
fit.arbol2.train <- rpart(formula = Resultado ~ .,
                          data = arbol2.train, method = "class")
fit.arbol2.test <- rpart(formula = Resultado ~ .,
                         data = arbol2.test, method = "class")
```

Puesto que los árboles fueron construídos a partir de conjuntos de observaciones diferentes, se espera que los resultados sean diferentes, sin ninguna duda. Lo chequeo:

```
identical(fit.arbol2.train, fit.arbol2.test)
```

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

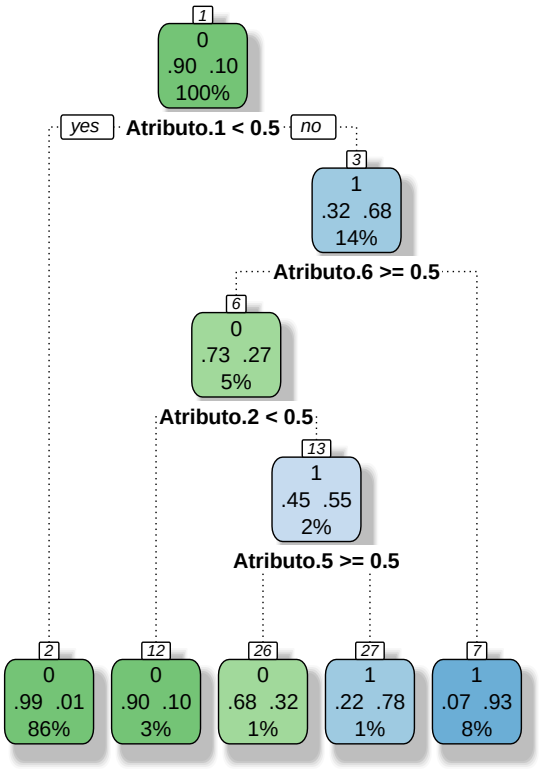
... chequeo además la primer afirmación:

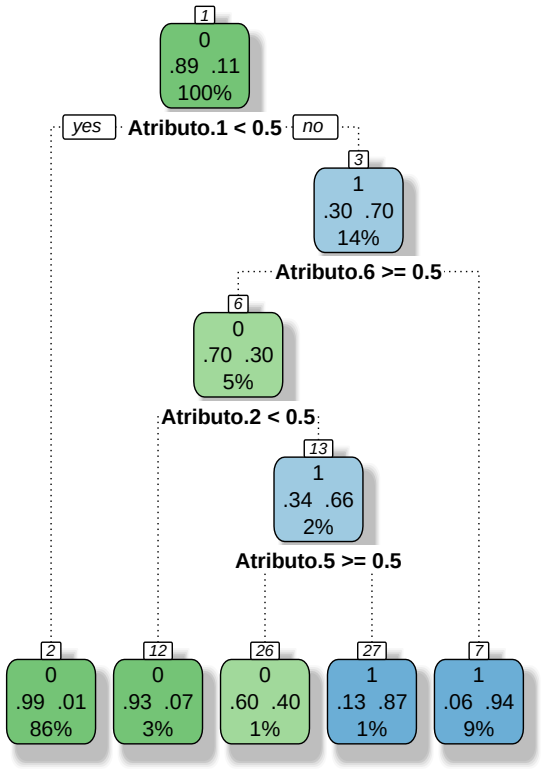
```
identical(arbol2.train, arbol2.test)
```

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

Es decir que rpart() aprende cosas diferentes de conjuntos de observaciones diferentes. De todos modos, se esperaría que los dos árboles consideren la relevancia de cada atributo de igual modo. Veámoslo!

```
par(mfrow = c(1,2))
fancyRpartPlot(fit.arbol2.train)
fancyRpartPlot(fit.arbol2.test)
```






```
par(mfrow = c(1,1))
```

Se observa que la importancia relativa de los atributos predictivos en los dos modelos es la misma.