rpart prune

Shiva Sankar Modala

2023-03-28

# Loading the necessary packages for solving the question  
library(rpart)  
# Package to create the binary decision tree  
library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.2.3

# Initialize a function for the gini index  
gini <- function(m) {  
 gini.index = 2 \* m \* (1 - m)  
 return (gini.index)  
}

# Initializing the function for the entropy value  
entropy <- function(n) {  
 entropy = (n \* log(n) + (1 - n) \* log(1 - n))  
 return (entropy)  
}

# set the seed value to 150  
set.seed(150)

# Normal distribution for Mean as 5 and Standard Deviation as 2  
x<-rnorm(n=150,mean=5,sd=2)

# Normal distribution for Mean as -5 and Standard Deviation as 2  
y<-rnorm(n=150,mean=-5,sd=2)

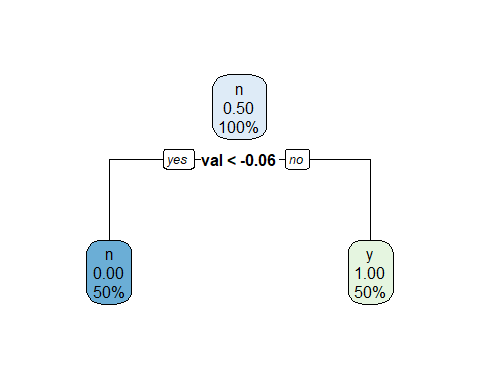
# Make the dataframe based in the values from the Normal distribution  
dt1 <- data.frame(val = x,label=rep("y",150))  
dt2 <- data.frame(val = y,label=rep("n",150))

# Combine the two dataframes into one using rbind  
dt <- rbind(dt1,dt2)

# Separating the label  
dt$label <- as.factor(dt$label)

# Making use of rpart to place the text inside the tree  
dtree <- rpart(label~val,dt,method="class")

# Now, plot the rpart of the decision tree  
rpart.plot(dtree)



cat <- "The threshold value for the first split will be -0.06, as observed from the above. Two leaf nodes and one root node make up the tree. Both classes can be classified individually by a tree, demonstrating empirical distribution."

# P is the probability of each node  
p=c(.5, 0, 1)

# Calculating the gini vlues and entropy based on the above function  
gini\_values=sapply(p, gini)  
gini\_values

## [1] 0.5 0.0 0.0

entropy\_values=sapply(p, entropy)  
entropy\_values

## [1] -0.6931472 NaN NaN

# The gini values for above tree - 0.5, 0.0, 0.0  
# The entropy values for above tree - 0.6931472, NaN, NaN

# set the seed value to 150  
set.seed(150)

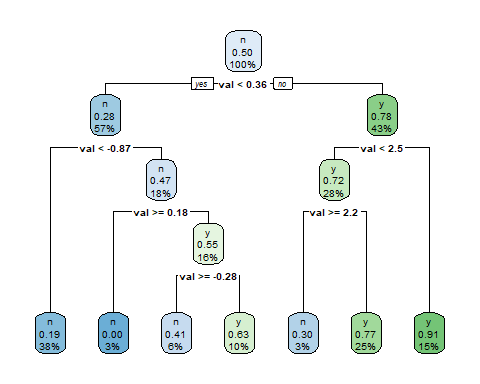
# Normal distribution for Mean as 1 and Standard Deviation as 2  
x1<-rnorm(n=150,mean=1,sd=2)

# Normal distribution for Mean as -1 and Standard Deviation as 2  
y1<-rnorm(n=150,mean=-1,sd=2)

# Make the dataframe based in the values from the Normal distribution  
dt3 <- data.frame(val = x1,label=rep("y",150))  
# Make the dataframe based in the values from the Normal distribution  
dt4 <- data.frame(val = y1,label=rep("n",150))

# Combining the two dataframes into one using rbind  
data <- rbind(dt3,dt4)  
data$label <- as.factor(data$label)

# Making use of rpart to place the text inside the tree  
dtree1 <- rpart(label~val,data,method="class")  
# Now, plot the rpart of the decision tree  
rpart.plot(dtree1)



cat <- "The threshold value for the first split, based on the tree above, is 0.36.   
The tree comprises 13 nodes, one of which is the root node.   
There are a total of 7 leaf nodes on the tree.   
Large tree size indicates the node's presence of more distinct labels, which led to the tree's size.   
Consequently, this tree has higher label overlap in the nodes."

# P1 is the probability of each node  
p1=c(.5,0.22,0.72,0.28,0.53,0.45,0.09,0.23,0.70,0.37,0.59,1.0,0.81)

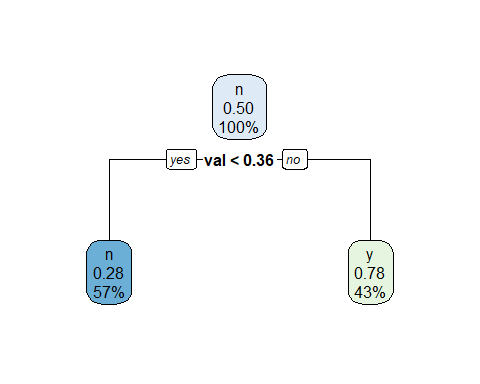
# Used the probability to get the gini and entropy  
gini\_values1=sapply(p1, gini)  
gini\_values1

## [1] 0.5000 0.3432 0.4032 0.4032 0.4982 0.4950 0.1638 0.3542 0.4200 0.4662  
## [11] 0.4838 0.0000 0.3078

entropy\_values1=sapply(p1, entropy)  
entropy\_values1

## [1] -0.6931472 -0.5269080 -0.5929533 -0.5929533 -0.6913461 -0.6881388  
## [7] -0.3025378 -0.5392763 -0.6108643 -0.6589557 -0.6768585 NaN  
## [13] -0.4862230

newtree <- prune.rpart(dtree1,cp=0.1)  
rpart.plot(newtree)



cat <- " The threshold value for the first split will be 0.36. The tree has one root node and 2 leaf nodes."  
# P2 is the probability of each node  
p2=c(.5,0.22,0.72)  
gini\_values2=sapply(p2, gini)  
gini\_values2

## [1] 0.5000 0.3432 0.4032

entropy\_values2=sapply(p2, entropy)  
entropy\_values2

## [1] -0.6931472 -0.5269080 -0.5929533

# The gini values for above tree = 0.5000, 0.3432, 0.4032  
# The entropy values for above tree = -0.6931472, -0.5269080, -0.5929533