ML Algorithms for classification

Decision trees

Hicham Zmarrou

Notebook – <http://bit.ly/2q9NPSU>

Table of Contents

[Visit my website](http://aardona.nl/) for more like this!

**References**

Most of this material is borrowed from:

* Textbook: [An Introduction to Recursive Partitioning Using the RPART Routines](https://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf)

You are here: Home / Exercises / Exercises (beginner) / Recursive Partitioning and Regression Trees Exercises

## 

## Exercise 1

## 

Consider the Kyphosis data frame(type help("kyphosis") for more details), that contains:

* Kyphosis:a factor with levels “absent” and “present”" indicating if a kyphosis (a type of deformation) was present after the operation.
* **Age**: in months.
* **Number**: the number of vertebrae involved.
* **Start**: the number of the first (topmost) vertebra operated on.

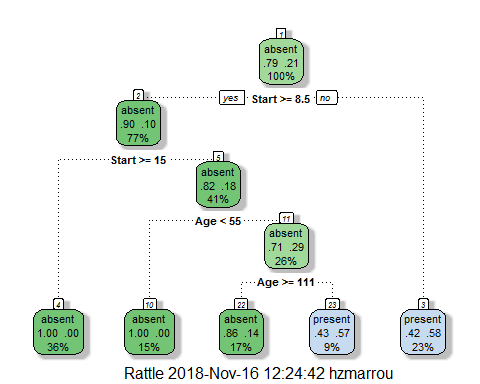
1. Build a tree to classify Kyphosis from Age, Number and Start.

library('rattle')

## Rattle: A free graphical interface for data science with R.  
## Version 5.2.0 Copyright (c) 2006-2018 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

library('rpart')  
library("rpart.plot")  
TREE=rpart(Kyphosis ~ Age + Number + Start, data=kyphosis,method="class")  
#TREE  
fancyRpartPlot(TREE)

## Warning in as.POSIXlt.POSIXct(x, tz): unable to identify current timezone 'C':  
## please set environment variable 'TZ'



asRules(TREE)

##   
## Rule number: 3 [Kyphosis=present cover=19 (23%) prob=0.58]  
## Start< 8.5  
##   
## Rule number: 23 [Kyphosis=present cover=7 (9%) prob=0.57]  
## Start>=8.5  
## Start< 14.5  
## Age>=55  
## Age< 111  
##   
## Rule number: 22 [Kyphosis=absent cover=14 (17%) prob=0.14]  
## Start>=8.5  
## Start< 14.5  
## Age>=55  
## Age>=111  
##   
## Rule number: 10 [Kyphosis=absent cover=12 (15%) prob=0.00]  
## Start>=8.5  
## Start< 14.5  
## Age< 55  
##   
## Rule number: 4 [Kyphosis=absent cover=29 (36%) prob=0.00]  
## Start>=8.5  
## Start>=14.5

### How to read this these results:

Consider for example **Rule number 23**. The rule gives the path to this final node (1. if Start>=8.5; 2. if Start< 14.5; 3. if Age>=55 and if Age< 111) The prob means how likely will the Kyphosis present under the path of this rule. cover= xx means there are xx records split into that path (branch) of the tree at that node.

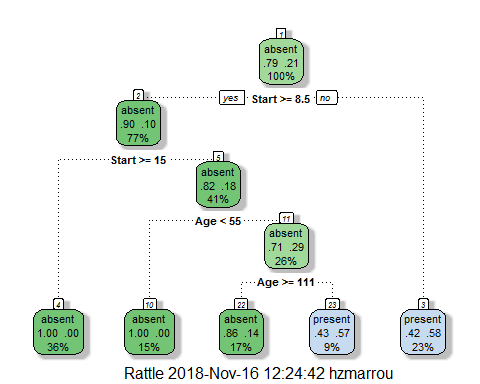
## 

## Exercise 2

## 

Consider the tree build in exercise 1. 1. Which variables are used to explain kyhosis presence? 2. How many observations contains the terminal nodes.

fancyRpartPlot(TREE)



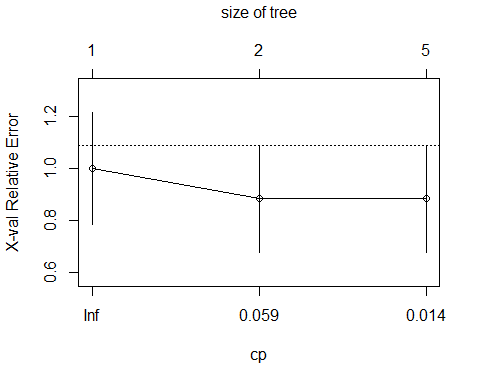
TREE

## n= 81   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 81 17 absent (0.79012346 0.20987654)   
## 2) Start>=8.5 62 6 absent (0.90322581 0.09677419)   
## 4) Start>=14.5 29 0 absent (1.00000000 0.00000000) \*  
## 5) Start< 14.5 33 6 absent (0.81818182 0.18181818)   
## 10) Age< 55 12 0 absent (1.00000000 0.00000000) \*  
## 11) Age>=55 21 6 absent (0.71428571 0.28571429)   
## 22) Age>=111 14 2 absent (0.85714286 0.14285714) \*  
## 23) Age< 111 7 3 present (0.42857143 0.57142857) \*  
## 3) Start< 8.5 19 8 present (0.42105263 0.57894737) \*

printcp(TREE)

##   
## Classification tree:  
## rpart(formula = Kyphosis ~ Age + Number + Start, data = kyphosis,   
## method = "class")  
##   
## Variables actually used in tree construction:  
## [1] Age Start  
##   
## Root node error: 17/81 = 0.20988  
##   
## n= 81   
##   
## CP nsplit rel error xerror xstd  
## 1 0.176471 0 1.00000 1.00000 0.21559  
## 2 0.019608 1 0.82353 0.88235 0.20565  
## 3 0.010000 4 0.76471 0.88235 0.20565

plotcp(TREE)



## 

## Exercise 3

## 

Consider the Kyphosis data frame.

1. Build a tree using the first 60 observations of kyphosis.

TREE=rpart(Kyphosis ~ Age + Number + Start, data=kyphosis[1:60,],method="class")

1. Predict the kyphosis presence for the other 21 observations.

PR=predict(TREE,kyphosis[61:81,],type='class')

1. Which is the misclassification rate (prediction error)

test=kyphosis$Kyphosis[61:81]  
table(PR,test)

## test  
## PR absent present  
## absent 14 2  
## present 3 2

rate=100\*length(which(PR!=test))/length(PR)

The misclassification rate is: 23.8095238

## 

## Exercise 4

## 

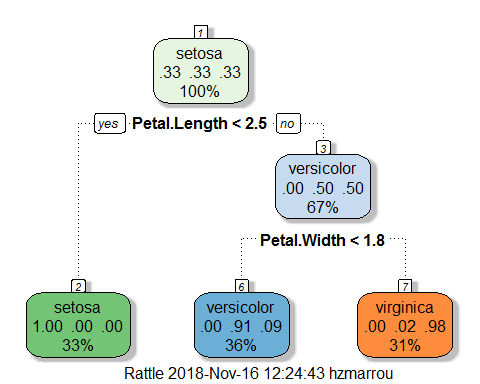
Consider the iris data frame(type help('iris') for more details). 1. Build a tree to classify Species from the other variables.

TREE2=rpart(Species ~ ., data=iris,method="class")  
TREE2

## n= 150   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 150 100 setosa (0.33333333 0.33333333 0.33333333)   
## 2) Petal.Length< 2.45 50 0 setosa (1.00000000 0.00000000 0.00000000) \*  
## 3) Petal.Length>=2.45 100 50 versicolor (0.00000000 0.50000000 0.50000000)   
## 6) Petal.Width< 1.75 54 5 versicolor (0.00000000 0.90740741 0.09259259) \*  
## 7) Petal.Width>=1.75 46 1 virginica (0.00000000 0.02173913 0.97826087) \*

1. Plot the trees.

fancyRpartPlot(TREE2)



## 

## Exercise 5

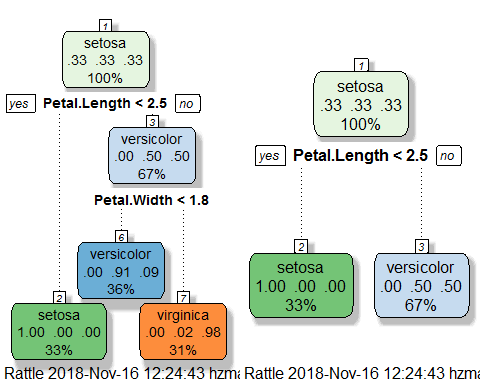
## 

Consider the tree build in exercise 4. 1. Prune the the using median complexity parameter (cp) associated to the tree.

TP=prune(TREE2,cp=median(TREE2$cptable[,'CP']))

1. Plot in the same window the pruned and the original tree.

par(mfrow=c(1,2))  
  
fancyRpartPlot(TREE2)  
fancyRpartPlot(TP)



## 

## Exercise 6

## 

Consider the tree build in exercise 4.

1. In which terminal nodes is clasified each observations of iris?

TREE2$where

## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18   
## 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2   
## 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36   
## 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2   
## 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54   
## 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 4 4 4   
## 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72   
## 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 4   
## 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90   
## 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4   
## 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108   
## 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 4 5   
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126   
## 5 5 5 5 5 5 5 5 5 5 5 4 5 5 5 5 5 5   
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144   
## 5 5 5 4 5 5 5 4 4 5 5 5 5 5 5 5 5 5   
## 145 146 147 148 149 150   
## 5 5 5 5 5 5

1. Which Specie has a flower of Petal.Length greater than 2.45 and Petal.Width less than 1.75.

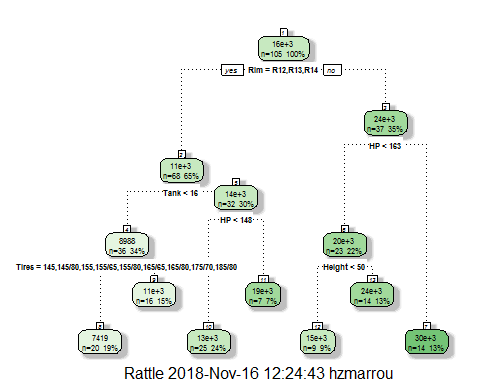
## 

## Exercise 7

## 

Consider the car90 data frame(type help(‘car90’) for more details). 1. Build a tree to predict Price from the other variables.

tree=rpart(Price ~ ., data=car90,method="anova")  
fancyRpartPlot(tree)



1. Plot the trees, add nodes information.

## 

## Exercise 8

## 

Consider the tree build in exercise 7. 1. Which variables are used to explain the price?

#tree  
#printcp(tree)  
asRules(tree)

##   
## Rule number: 10 [Price=13014.72 cover=25 (24%)]  
## Rim=R12,R13,R14  
## Tank>=15.65  
## HP< 147.5  
##   
## Rule number: 8 [Price=7419.05 cover=20 (19%)]  
## Rim=R12,R13,R14  
## Tank< 15.65  
## Tires=145,145/80,155,155/65,155/80,165/65,165/80,175/70,185/80  
##   
## Rule number: 9 [Price=10949.8125 cover=16 (15%)]  
## Rim=R12,R13,R14  
## Tank< 15.65  
## Tires=185/60,185/70,195/60,195/70,195/75,205/75  
##   
## Rule number: 7 [Price=29643.0714285714 cover=14 (13%)]  
## Rim=R15,R16,R17  
## HP>=162.5  
##   
## Rule number: 13 [Price=23614.2857142857 cover=14 (13%)]  
## Rim=R15,R16,R17  
## HP< 162.5  
## Height>=49.75  
##   
## Rule number: 12 [Price=14755 cover=9 (9%)]  
## Rim=R15,R16,R17  
## HP< 162.5  
## Height< 49.75  
##   
## Rule number: 11 [Price=18886.2857142857 cover=7 (7%)]  
## Rim=R12,R13,R14  
## Tank>=15.65  
## HP>=147.5

1. Which terminal nodes have a value of mean Price, less than mean(car90$Price)?

summary(tree)

## Call:  
## rpart(formula = Price ~ ., data = car90, method = "anova")  
## n=105 (6 observations deleted due to missingness)  
##   
## CP nsplit rel error xerror xstd  
## 1 0.50539502 0 1.0000000 1.0307996 0.1640696  
## 2 0.11023313 1 0.4946050 0.7049729 0.1016778  
## 3 0.06712695 2 0.3843719 0.6655547 0.1182335  
## 4 0.06040402 3 0.3172449 0.6510914 0.1189696  
## 5 0.02648632 4 0.2568409 0.6448514 0.1474723  
## 6 0.01556719 5 0.2303546 0.6644783 0.1482997  
## 7 0.01000000 6 0.2147874 0.6509424 0.1486581  
##   
## Variable importance  
## Rim Tires Tank HP Disp   
## 18 15 14 14 11   
## Disp2 Weight Type Height Steering   
## 10 3 3 2 2   
## Sratio.p Wheel.base Length Rear.Hd Rear.Seating   
## 2 2 1 1 1   
## Gear2 Model2   
## 1 1   
##   
## Node number 1: 105 observations, complexity param=0.505395  
## mean=15805.22, MSE=6.779302e+07   
## left son=2 (68 obs) right son=3 (37 obs)  
## Primary splits:  
## Rim splits as LLLRRR, improve=0.5053950, (0 missing)  
## Tires splits as LLLLLLLLLLRLLLRLRLLRRRLLRRRRRR, improve=0.4721902, (0 missing)  
## Disp < 156 to the left, improve=0.4601461, (0 missing)  
## Disp2 < 2.55 to the left, improve=0.4601461, (0 missing)  
## HP < 154 to the left, improve=0.4548845, (0 missing)  
## Surrogate splits:  
## Tires splits as LLLLLLLLLLRLLLRLRLLRRRLRRRRRRR, agree=0.876, adj=0.649, (0 split)  
## Tank < 17.8 to the left, agree=0.848, adj=0.568, (0 split)  
## Disp < 156 to the left, agree=0.829, adj=0.514, (0 split)  
## Disp2 < 2.55 to the left, agree=0.829, adj=0.514, (0 split)  
## HP < 126.5 to the left, agree=0.819, adj=0.486, (0 split)  
##   
## Node number 2: 68 observations, complexity param=0.06712695  
## mean=11487.5, MSE=1.619855e+07   
## left son=4 (36 obs) right son=5 (32 obs)  
## Primary splits:  
## Tank < 15.65 to the left, improve=0.4337967, (0 missing)  
## Type splits as RRRLRR, improve=0.4153956, (0 missing)  
## Weight < 2567.5 to the left, improve=0.4063382, (0 missing)  
## Length < 173.5 to the left, improve=0.3843167, (0 missing)  
## HP < 105.5 to the left, improve=0.3775031, (0 missing)  
## Surrogate splits:  
## Tires splits as LLLLLLLLRLRRRL-LRLRR-RR--R----, agree=0.838, adj=0.656, (0 split)  
## Weight < 2705 to the left, agree=0.838, adj=0.656, (0 split)  
## Length < 173.5 to the left, agree=0.824, adj=0.625, (0 split)  
## Steering splits as LRL, agree=0.809, adj=0.594, (0 split)  
## Type splits as RRRLLR, agree=0.809, adj=0.594, (0 split)  
##   
## Node number 3: 37 observations, complexity param=0.1102331  
## mean=23740.49, MSE=6.538458e+07   
## left son=6 (23 obs) right son=7 (14 obs)  
## Primary splits:  
## HP < 162.5 to the left, improve=0.3243466, (0 missing)  
## Tires splits as ----------R---LLR--LRRLLRRRRRR, improve=0.2236680, (0 missing)  
## Tank < 16.45 to the left, improve=0.1964397, (0 missing)  
## HP.revs < 5650 to the left, improve=0.1822724, (0 missing)  
## Country splits as -RLRL---LL, improve=0.1741968, (0 missing)  
## Surrogate splits:  
## Tank < 18.65 to the left, agree=0.784, adj=0.429, (0 split)  
## Tires splits as ----------L---LLL--LLRLLLLRRLR, agree=0.757, adj=0.357, (0 split)  
## Weight < 3272.5 to the left, agree=0.757, adj=0.357, (0 split)  
## Disp < 164 to the left, agree=0.730, adj=0.286, (0 split)  
## Disp2 < 2.85 to the left, agree=0.703, adj=0.214, (0 split)  
##   
## Node number 4: 36 observations, complexity param=0.01556719  
## mean=8988.278, MSE=5254037   
## left son=8 (20 obs) right son=9 (16 obs)  
## Primary splits:  
## Tires splits as LLLLLLLL-R-R-L-R-RR---R-------, improve=0.5858533, (0 missing)  
## Type splits as R-RLRR, improve=0.4787252, (0 missing)  
## Rim splits as LLR---, improve=0.4188695, (0 missing)  
## HP < 94 to the left, improve=0.4179510, (0 missing)  
## Disp < 94 to the left, improve=0.3838392, (0 missing)  
## Surrogate splits:  
## Rim splits as LLR---, agree=0.944, adj=0.875, (0 split)  
## Type splits as R-RLRR, agree=0.917, adj=0.812, (0 split)  
## Steering splits as LRL, agree=0.889, adj=0.750, (0 split)  
## HP < 104 to the left, agree=0.861, adj=0.688, (0 split)  
## Weight < 2447.5 to the left, agree=0.861, adj=0.688, (0 split)  
##   
## Node number 5: 32 observations, complexity param=0.02648632  
## mean=14299.12, MSE=1.357901e+07   
## left son=10 (25 obs) right son=11 (7 obs)  
## Primary splits:  
## HP < 147.5 to the left, improve=0.4338882, (0 missing)  
## Country splits as --RRLLL-RR, improve=0.2173648, (0 missing)  
## Reliability splits as RRRRL, improve=0.2134859, (10 missing)  
## Sratio.p < 0.825 to the left, improve=0.2008423, (0 missing)  
## Tires splits as -------LL-RLL--LRLRR-LR--L----, improve=0.1849114, (0 missing)  
## Surrogate splits:  
## Disp < 166.5 to the left, agree=0.906, adj=0.571, (0 split)  
## Disp2 < 2.75 to the left, agree=0.906, adj=0.571, (0 split)  
## Gear2 < 2.405 to the right, agree=0.906, adj=0.571, (0 split)  
## Model2 splits as L----R----L-RR---L-LR, agree=0.906, adj=0.571, (0 split)  
## Tires splits as -------LL-LLL--LLLLR-LR--L----, agree=0.875, adj=0.429, (0 split)  
##   
## Node number 6: 23 observations, complexity param=0.06040402  
## mean=20147.61, MSE=3.755274e+07   
## left son=12 (9 obs) right son=13 (14 obs)  
## Primary splits:  
## Height < 49.75 to the left, improve=0.4978181, (0 missing)  
## Type splits as RRR-LL, improve=0.4858895, (0 missing)  
## Wheel.base < 104.5 to the left, improve=0.4573396, (0 missing)  
## Rear.Seating < 28.25 to the left, improve=0.3966693, (0 missing)  
## Sratio.p < 0.755 to the left, improve=0.3849062, (0 missing)  
## Surrogate splits:  
## Sratio.p < 0.74 to the left, agree=0.913, adj=0.778, (0 split)  
## Wheel.base < 104.5 to the left, agree=0.913, adj=0.778, (0 split)  
## Rear.Hd < 0.75 to the left, agree=0.870, adj=0.667, (0 split)  
## Type splits as RRR-LR, agree=0.870, adj=0.667, (0 split)  
## Rear.Seating < 25 to the left, agree=0.826, adj=0.556, (0 split)  
##   
## Node number 7: 14 observations  
## mean=29643.07, MSE=5.506053e+07   
##   
## Node number 8: 20 observations  
## mean=7419.05, MSE=1592680   
##   
## Node number 9: 16 observations  
## mean=10949.81, MSE=2905020   
##   
## Node number 10: 25 observations  
## mean=13014.72, MSE=4782130   
##   
## Node number 11: 7 observations  
## mean=18886.29, MSE=1.806262e+07   
##   
## Node number 12: 9 observations  
## mean=14755, MSE=9671366   
##   
## Node number 13: 14 observations  
## mean=23614.29, MSE=2.47642e+07

## 

## Exercise 9

## 

Consider the car.test.frame data frame (type help(‘car.test.frame’) for more details). 1. Build a tree to explain Mileage using the other variables.

TC=rpart(Mileage~., data=car.test.frame)  
TC

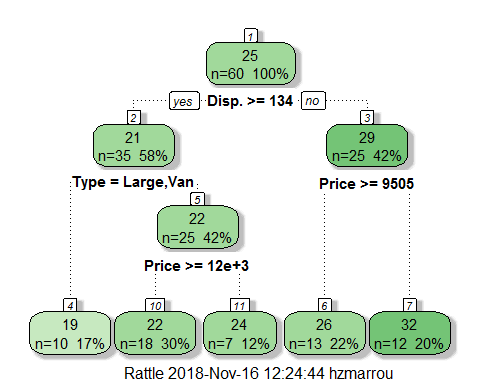
## n= 60   
##   
## node), split, n, deviance, yval  
## \* denotes terminal node  
##   
## 1) root 60 1354.58300 24.58333   
## 2) Disp.>=134 35 154.40000 21.40000   
## 4) Type=Large,Van 10 22.10000 19.30000 \*  
## 5) Type=Compact,Medium,Sporty 25 70.56000 22.24000   
## 10) Price>=11522 18 26.44444 21.55556 \*  
## 11) Price< 11522 7 14.00000 24.00000 \*  
## 3) Disp.< 134 25 348.96000 29.04000   
## 6) Price>=9504.5 13 32.30769 26.23077 \*  
## 7) Price< 9504.5 12 102.91670 32.08333 \*

1. Snip the tree in nodes number 2.

TS=snip.rpart(TC,toss=2)

1. Plot both tree together

fancyRpartPlot(TC)



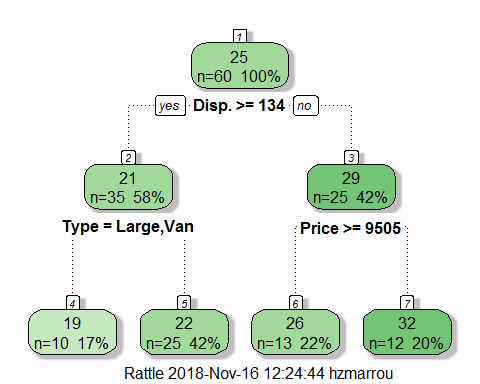
## 

## Exercise 10

## 

Consider the tree build in exercise 9. 1. Which is the depth of the tree (with the root node counted as depth 0). 2. Set the maximum depth of the final tree on 2

TC2=rpart(Mileage~., data=car.test.frame,maxdepth=2)  
fancyRpartPlot(TC2)



#plot both together  
par(mfrow=c(1,2))  
fancyRpartPlot(TC)  
fancyRpartPlot(TC2)

