Decision Tree

Bhuvan Kapoor [23MDT1003]

2024-03-26

## DECISION TREE  
  
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(rpart.plot)

## Loading required package: rpart

library(e1071)  
  
data\_url <- c('https://archive.ics.uci.edu/ml/machine-learning-databases/car/car.data')  
download.file(url = data\_url , destfile = "car.data")  
  
car\_df <- read.csv('car.data', sep = ',', colClasses = c(rep('factor',7)), header = F)  
head(car\_df)

## V1 V2 V3 V4 V5 V6 V7  
## 1 vhigh vhigh 2 2 small low unacc  
## 2 vhigh vhigh 2 2 small med unacc  
## 3 vhigh vhigh 2 2 small high unacc  
## 4 vhigh vhigh 2 2 med low unacc  
## 5 vhigh vhigh 2 2 med med unacc  
## 6 vhigh vhigh 2 2 med high unacc

str(car\_df)

## 'data.frame': 1728 obs. of 7 variables:  
## $ V1: Factor w/ 4 levels "high","low","med",..: 4 4 4 4 4 4 4 4 4 4 ...  
## $ V2: Factor w/ 4 levels "high","low","med",..: 4 4 4 4 4 4 4 4 4 4 ...  
## $ V3: Factor w/ 4 levels "2","3","4","5more": 1 1 1 1 1 1 1 1 1 1 ...  
## $ V4: Factor w/ 3 levels "2","4","more": 1 1 1 1 1 1 1 1 1 2 ...  
## $ V5: Factor w/ 3 levels "big","med","small": 3 3 3 2 2 2 1 1 1 3 ...  
## $ V6: Factor w/ 3 levels "high","low","med": 2 3 1 2 3 1 2 3 1 2 ...  
## $ V7: Factor w/ 4 levels "acc","good","unacc",..: 3 3 3 3 3 3 3 3 3 3 ...

summary(car\_df)

## V1 V2 V3 V4 V5 V6   
## high :432 high :432 2 :432 2 :576 big :576 high:576   
## low :432 low :432 3 :432 4 :576 med :576 low :576   
## med :432 med :432 4 :432 more:576 small:576 med :576   
## vhigh:432 vhigh:432 5more:432   
## V7   
## acc : 384   
## good : 69   
## unacc:1210   
## vgood: 65

set.seed(3013)  
intrain <- createDataPartition(y = car\_df$V7,   
 p=0.7,   
 list = F)  
train\_data <- car\_df[intrain,]  
test\_data <- car\_df[-intrain,]  
  
dim(train\_data)

## [1] 1211 7

dim(test\_data)

## [1] 517 7

trctrl <- trainControl(method = 'cv', number = 10)  
set.seed(3333)  
dtree\_fit <- train(V7 ~ .,   
 data = train\_data,   
 method = 'rpart',   
 parms = list(split = 'Information'),   
 trControl = trctrl)  
dtree\_fit

## CART   
##   
## 1211 samples  
## 6 predictor  
## 4 classes: 'acc', 'good', 'unacc', 'vgood'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 1090, 1092, 1089, 1089, 1089, 1090, ...   
## Resampling results across tuning parameters:  
##   
## cp Accuracy Kappa   
## 0.04395604 0.7985873 0.5622043  
## 0.05769231 0.7787306 0.5551887  
## 0.07142857 0.7696396 0.5045378  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was cp = 0.04395604.

test\_pred <- predict(dtree\_fit, test\_data)  
test\_pred

## [1] unacc unacc unacc unacc acc unacc acc acc acc acc unacc acc   
## [13] acc unacc acc unacc acc unacc unacc acc unacc unacc unacc unacc  
## [25] unacc acc unacc acc unacc acc acc unacc unacc unacc acc acc   
## [37] unacc unacc unacc acc acc unacc unacc unacc unacc unacc unacc acc   
## [49] acc unacc unacc unacc unacc acc acc acc unacc unacc acc acc   
## [61] unacc acc unacc unacc unacc acc unacc acc acc unacc unacc acc   
## [73] acc acc unacc unacc unacc acc unacc acc unacc unacc acc unacc  
## [85] unacc unacc acc unacc acc acc unacc unacc unacc unacc acc unacc  
## [97] acc unacc acc unacc unacc acc unacc acc unacc unacc unacc unacc  
## [109] unacc acc unacc unacc unacc unacc unacc acc acc unacc unacc unacc  
## [121] unacc unacc unacc unacc acc acc unacc unacc acc acc unacc unacc  
## [133] unacc unacc unacc acc unacc acc acc unacc acc unacc unacc unacc  
## [145] acc unacc unacc acc unacc unacc unacc acc acc acc acc acc   
## [157] unacc unacc acc unacc unacc acc acc unacc unacc acc unacc acc   
## [169] acc unacc unacc unacc unacc unacc acc acc unacc unacc unacc unacc  
## [181] unacc unacc acc acc unacc unacc unacc unacc acc unacc acc acc   
## [193] acc unacc unacc unacc unacc acc unacc unacc unacc acc unacc unacc  
## [205] acc acc unacc acc unacc unacc unacc acc unacc acc unacc acc   
## [217] acc unacc unacc acc acc unacc unacc unacc acc unacc unacc unacc  
## [229] unacc unacc acc acc acc unacc acc unacc unacc unacc unacc acc   
## [241] unacc unacc acc acc unacc unacc unacc unacc unacc unacc unacc acc   
## [253] acc unacc unacc acc unacc unacc unacc unacc unacc acc unacc unacc  
## [265] acc acc acc unacc unacc unacc unacc unacc unacc unacc unacc acc   
## [277] unacc unacc acc unacc acc unacc acc unacc acc unacc unacc unacc  
## [289] acc unacc unacc unacc acc unacc unacc unacc unacc acc acc acc   
## [301] unacc unacc unacc unacc unacc unacc unacc acc acc acc unacc acc   
## [313] unacc unacc acc unacc acc unacc acc acc acc acc acc acc   
## [325] acc unacc acc unacc unacc unacc acc unacc unacc acc acc acc   
## [337] unacc unacc unacc acc unacc unacc unacc unacc unacc acc acc acc   
## [349] unacc unacc unacc unacc acc acc unacc acc unacc unacc acc acc   
## [361] unacc unacc unacc unacc acc unacc unacc unacc acc unacc unacc unacc  
## [373] unacc unacc acc acc acc acc acc unacc unacc unacc unacc unacc  
## [385] unacc acc acc unacc unacc unacc unacc acc acc unacc acc unacc  
## [397] unacc unacc acc acc unacc unacc acc acc unacc unacc unacc unacc  
## [409] unacc unacc acc unacc acc unacc unacc acc acc acc acc unacc  
## [421] unacc unacc unacc unacc unacc acc acc unacc unacc unacc unacc unacc  
## [433] unacc unacc unacc acc acc acc unacc acc unacc unacc unacc acc   
## [445] acc acc unacc unacc acc unacc acc unacc unacc unacc unacc unacc  
## [457] acc unacc acc acc acc unacc unacc unacc acc acc unacc unacc  
## [469] acc unacc unacc unacc unacc acc acc unacc acc unacc unacc unacc  
## [481] unacc unacc unacc unacc acc unacc unacc unacc unacc acc unacc acc   
## [493] acc unacc unacc unacc unacc unacc acc unacc unacc unacc unacc acc   
## [505] unacc unacc acc unacc unacc acc acc unacc unacc acc acc unacc  
## [517] unacc  
## Levels: acc good unacc vgood

confusionMatrix(test\_pred, test\_data$V7)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction acc good unacc vgood  
## acc 100 16 57 19  
## good 0 0 0 0  
## unacc 15 4 306 0  
## vgood 0 0 0 0  
##   
## Overall Statistics  
##   
## Accuracy : 0.7853   
## 95% CI : (0.7474, 0.8199)  
## No Information Rate : 0.7021   
## P-Value [Acc > NIR] : 1.282e-05   
##   
## Kappa : 0.549   
##   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: acc Class: good Class: unacc Class: vgood  
## Sensitivity 0.8696 0.00000 0.8430 0.00000  
## Specificity 0.7711 1.00000 0.8766 1.00000  
## Pos Pred Value 0.5208 NaN 0.9415 NaN  
## Neg Pred Value 0.9538 0.96132 0.7031 0.96325  
## Prevalence 0.2224 0.03868 0.7021 0.03675  
## Detection Rate 0.1934 0.00000 0.5919 0.00000  
## Detection Prevalence 0.3714 0.00000 0.6286 0.00000  
## Balanced Accuracy 0.8204 0.50000 0.8598 0.50000

## Using RPART  
library(rpart)  
library(rpart.plot)  
  
data('iris')  
head(iris)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 1 5.1 3.5 1.4 0.2 setosa  
## 2 4.9 3.0 1.4 0.2 setosa  
## 3 4.7 3.2 1.3 0.2 setosa  
## 4 4.6 3.1 1.5 0.2 setosa  
## 5 5.0 3.6 1.4 0.2 setosa  
## 6 5.4 3.9 1.7 0.4 setosa

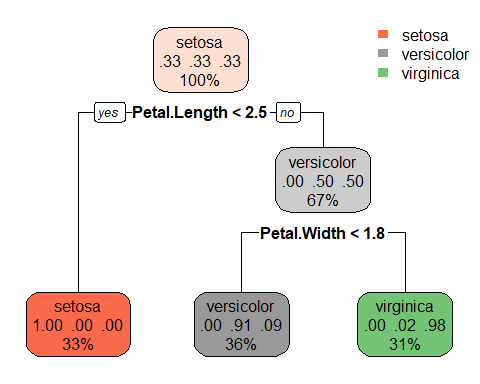
str(iris)

## '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 ...

summary(iris)

## Sepal.Length Sepal.Width Petal.Length Petal.Width   
## Min. :4.300 Min. :2.000 Min. :1.000 Min. :0.100   
## 1st Qu.:5.100 1st Qu.:2.800 1st Qu.:1.600 1st Qu.:0.300   
## Median :5.800 Median :3.000 Median :4.350 Median :1.300   
## Mean :5.843 Mean :3.057 Mean :3.758 Mean :1.199   
## 3rd Qu.:6.400 3rd Qu.:3.300 3rd Qu.:5.100 3rd Qu.:1.800   
## Max. :7.900 Max. :4.400 Max. :6.900 Max. :2.500   
## Species   
## setosa :50   
## versicolor:50   
## virginica :50   
##   
##   
##

set.seed(33)  
tree <- rpart(Species ~ ., data = iris, method = 'class')  
rpart.plot(tree)



unseen <- data.frame(Sepal.Length = c(5.3, 7.2),  
 Sepal.Width = c(2.9,3.9),  
 Petal.Length = c(1.7,5.4),  
 Petal.Width = c(0.8,2.3))  
unseen

## Sepal.Length Sepal.Width Petal.Length Petal.Width  
## 1 5.3 2.9 1.7 0.8  
## 2 7.2 3.9 5.4 2.3

predict(tree, unseen, type = 'class')

## 1 2   
## setosa virginica   
## Levels: setosa versicolor virginica