Decision Tree

2024-03-26

##Decision tress-   
  
library(caret)

## Warning: package 'caret' was built under R version 4.3.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.3.2

## Loading required package: lattice

library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.3.3

## Loading required package: rpart

## Warning: package 'rpart' was built under R version 4.3.3

library(e1071)

## Warning: package 'e1071' was built under R version 4.3.3

url = c("https://archive.ics.uci.edu/ml/machine-learning-databases/car/car.data")  
download.file(url = url, destfile = "car.data")  
  
car\_df = read.csv("car.data", sep = ",", colClasses = c("factor","factor","factor",  
 "factor","factor","factor","factor"),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(3033)  
intrain = createDataPartition(y = car\_df$V7, p = 0.7, list = F)  
  
training = car\_df[intrain,]  
testing = car\_df[-intrain,]  
  
dim(training)

## [1] 1211 7

dim(testing)

## [1] 517 7

trctrl = trainControl(method = "cv", number = 10) #cv-cross validation  
  
set.seed(3333)  
dtree\_fit = train(V7~.,data = training, 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   
## 0.02472527 0.8085676 0.5890963  
## 0.06043956 0.7844405 0.5570500  
## 0.07005495 0.7579942 0.3904587  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was cp = 0.02472527.

test\_pred = predict(dtree\_fit, newdata = testing)  
test\_pred

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

confusionMatrix(test\_pred,testing$V7)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction acc good unacc vgood  
## acc 83 20 44 19  
## good 0 0 0 0  
## unacc 32 0 319 0  
## vgood 0 0 0 0  
##   
## Overall Statistics  
##   
## Accuracy : 0.7776   
## 95% CI : (0.7392, 0.8127)  
## No Information Rate : 0.7021   
## P-Value [Acc > NIR] : 7.25e-05   
##   
## Kappa : 0.5078   
##   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: acc Class: good Class: unacc Class: vgood  
## Sensitivity 0.7217 0.00000 0.8788 0.00000  
## Specificity 0.7935 1.00000 0.7922 1.00000  
## Pos Pred Value 0.5000 NaN 0.9088 NaN  
## Neg Pred Value 0.9088 0.96132 0.7349 0.96325  
## Prevalence 0.2224 0.03868 0.7021 0.03675  
## Detection Rate 0.1605 0.00000 0.6170 0.00000  
## Detection Prevalence 0.3211 0.00000 0.6789 0.00000  
## Balanced Accuracy 0.7576 0.50000 0.8355 0.50000

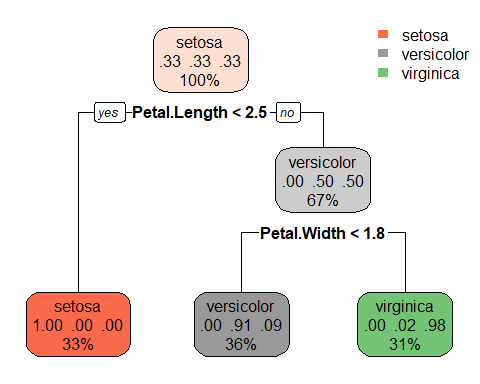
#Decision tress- RPART - Recursive Partitioning and Regression Tree  
data(iris)  
str(iris)

## '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   
##   
##   
##

library(rpart.plot)  
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