Module 4: Machine Learning Techniques Using R –Part 2

Assignment Solution

edureka!

edureka!
© 2014 Brain4ce Education Solutions Pvt. Ltd.

Version 1.0

Module 4 – Machine Learning Techniques using R – Part 2

Assignment Solution

Table of Contents

edureka!

Decision Trees

This shows how to build predictive models with packages "party", "rpart". It starts with building decision trees with package "party" and using the built tree for classification, followed by another way to build decision trees with package "rpart".

Decision Trees with package "party"

This section shows how to build a decision tree for the 'iris' data with function 'ctree()' in package "party".

The following are used to predict species of flowers:

→ Sepal.Length
→ Sepal.Width
→ Petal.Length
→ Petal.Width

Ctree(): Builds a decision tree.

Predict(): Makes prediction for new data.

Before modelling, the 'iris' data is split below into two subsets: Training (70%) and test (30%). The random seed is set to a fixed value below to make the results reproducible.

```
> 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$
```

```
> set.seed(1234)
> ind<-sample(2,nrow(iris), replace=TRUE, prob=c(0.7, 0.30))
> trainData<-iris[ind==1,]
> testData<-iris[ind==2,]
>
```

The function ctree() provides some parameters such as:

→ MinSplit
→ MinBusket
→ MaxSurrogate
→ MaxDepth

These parameters control the training of decision trees.

Below we use default settings to build a decision tree. In the code below, 'myFormula' specifies that "Species" is the target variable and all other variables are independent variables.

```
> library(party)
Loading required package: grid
Loading required package: zoo

Attaching package: 'zoo'

The following object(s) are masked from 'package:base':

    as.Date, as.Date.numeric

Loading required package: sandwich
Loading required package: strucchange
Loading required package: modeltools
Loading required package: stats4
Warning messages:
1: package 'party' was built under R version 2.15.3
2: package 'zoo' was built under R version 2.15.3
3: package 'sandwich' was built under R version 2.15.3
4: package 'strucchange' was built under R version 2.15.3
5: package 'modeltools' was built under R version 2.15.3
>
```

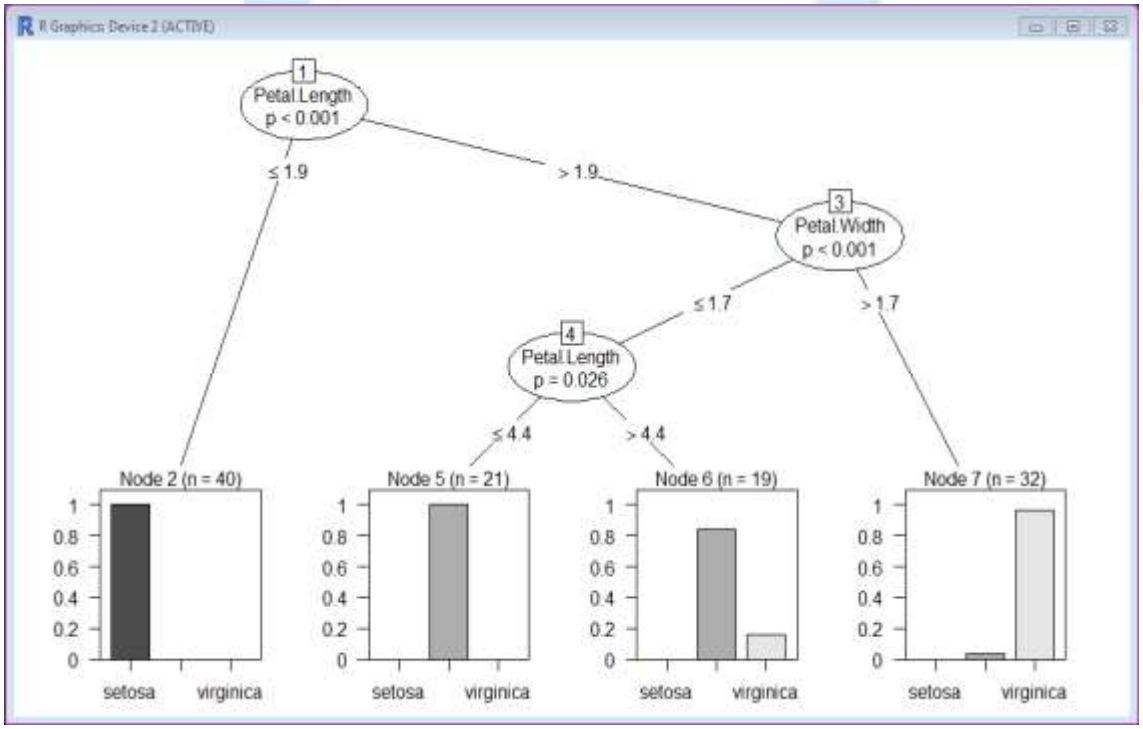
```
> myFormula<-Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width
> iris_ctree<-ctree(myFormula, data=trainData)
>
```

```
> table(predict(iris_ctree), trainData$Species)

             setosa versicolor virginica
  setosa         40          0         0
  versicolor      0         37         3
  virginica       0          1        31
>
```

Plot: Decision Tree

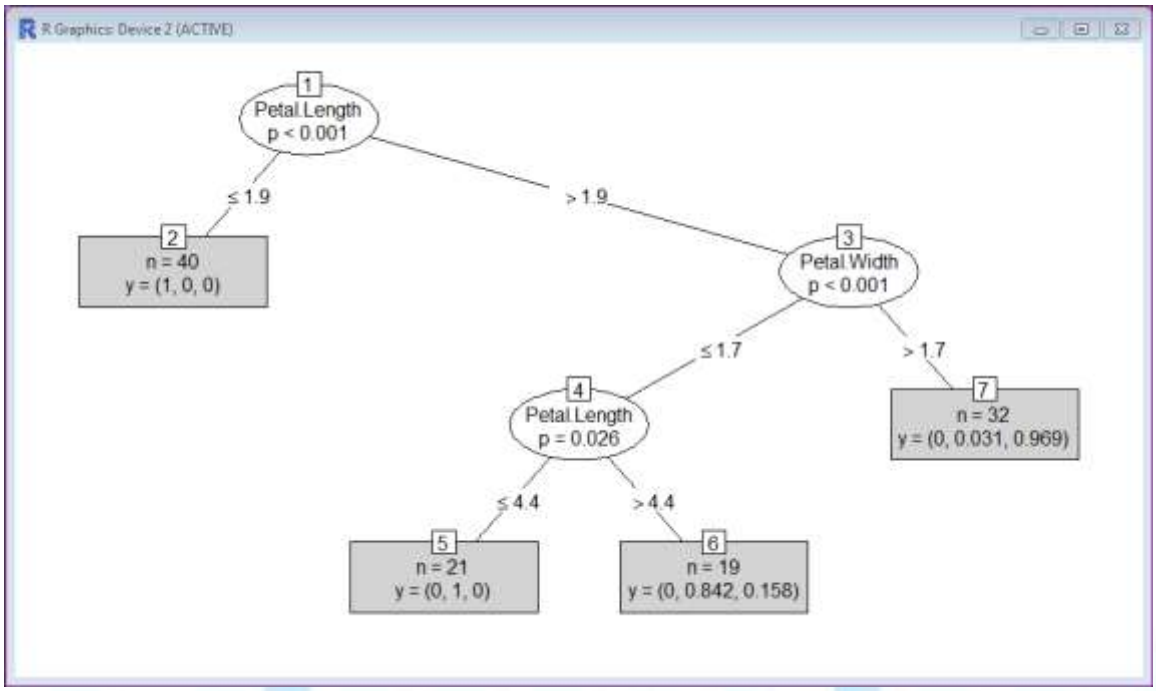
```
> print(iris_ctree)

        Conditional inference tree with 4 terminal nodes

Response:  Species
Inputs:  Sepal.Length, Sepal.Width, Petal.Length, Petal.Width
Number of observations:  112

1) Petal.Length <= 1.9; criterion = 1, statistic = 104.643
  2)*  weights = 40
1) Petal.Length > 1.9
  3) Petal.Width <= 1.7; criterion = 1, statistic = 48.939
    4) Petal.Length <= 4.4; criterion = 0.974, statistic = 7.397
      5)*  weights = 21
    4) Petal.Length > 4.4
      6)*  weights = 19
  3) Petal.Width > 1.7
    7)*  weights = 32
>
```

```
> plot(iris_ctree)
```



Plot: Decision Tree (Simple)

```
> plot(iris_ctree, type="simple")
```



In the above figure, bar plot for each leaf node shows the probabilities of an instance falling into the three species.

Predict on test data

```
> testPred<-predict(iris_ctree, newdata=testData)
> table(testPred, testData$Species)

testPred     setosa versicolor virginica
  setosa         10          0         0
  versicolor      0         12         2
  virginica       0          0        14
>
```

Decision Trees with Package "rpart"

Package "rpart" is used to build a decision tree on the bodyfat data. Function rpart() is used to build a decision tree, and the tree with the minimum prediction error is selected. After that, it is applied to the new data to make prediction with function predict().

Now loading the bodyfat data and having a look at it.

```
> library(mboost)
Loading required package: parallel
Loading required package: survival
Loading required package: splines
This is mboost 2.2-3. See 'package?mboost' and the NEWS file
for a complete list of changes.
Note: The default for the computation of the degrees of freedom has changed.
      For details see section 'Global Options' of '?bols'.
Warning message:
package 'mboost' was built under R version 2.15.3
>
```

```
> data(bodyfat)
> data("bodyfat", package= "mboost")
> dim(bodyfat)
[1] 71 10
>
```

```
> attributes(bodyfat)
$names
 [1] "age"          "DEXfat"       "waistcirc"    "hipcirc"
 [5] "elbowbreadth" "kneebreadth"  "anthro3a"     "anthro3b"
 [9] "anthro3c"     "anthro4"

$row.names
 [1] "47"  "48"  "49"  "50"  "51"  "52"  "53"  "54"  "55"  "56"  "57"  "58"
[13] "59"  "60"  "61"  "62"  "63"  "64"  "65"  "66"  "67"  "68"  "69"  "70"
[25] "71"  "72"  "73"  "74"  "75"  "76"  "77"  "78"  "79"  "80"  "81"  "82"
[37] "83"  "84"  "85"  "86"  "87"  "88"  "89"  "90"  "91"  "92"  "93"  "94"
[49] "95"  "96"  "97"  "98"  "99"  "100" "101" "102" "103" "104" "105" "106"
[61] "107" "108" "109" "110" "111" "112" "113" "114" "115" "116" "117"

$class
[1] "data.frame"

>
```

```
> bodyfat[1:5,]
   age DEXfat waistcirc hipcirc elbowbreadth kneebreadth anthro3a anthro3b
47  57  41.68     100.0   112.0          7.1         9.4      4.42      4.95
48  65  43.29      99.5   116.5          6.5         8.9      4.63      5.01
49  59  35.41      96.0   108.5          6.2         8.9      4.12      4.74
50  58  22.79      72.0    96.5          6.1         9.2      4.03      4.48
51  60  36.42      89.5   100.5          7.1        10.0      4.24      4.68
   anthro3c anthro4
47     4.50    6.13
48     4.48    6.37
49     4.60    5.82
50     3.91    5.66
51     4.15    5.91
> |
```

Next, the data is split into training and test subsets, and a decision tree is built on the training data.

```
> set.seed(1234)
> ind<-sample(2,nrow(bodyfat), replace=TRUE, prob=c(0.7, 0.3))
> bodyfat.train<-bodyfat[ind==1,]
> bodyfat.test<-bodyfat[ind==2,]
> |
```

```
> library(rpart)
```

```
> myFormula<-DEXfat ~ age+ waistcirc + hipcirc + elbowbreadth + kneebreadth
> bodyfat_rpart<-rpart(myFormula, data=bodyfat.train,
+ control=rpart.control(minsplit=10))
> attributes(bodyfat_rpart)
$names
 [1] "frame"     "where"     "call"      "terms"     "cptable"   "splits"    "method"
 [8] "parms"     "control"   "functions" "y"         "ordered"

$class
[1] "rpart"

> |
```

```
> print(bodyfat_rpart$cptable)
          CP nsplit  rel error    xerror       xstd
1 0.67272638      0 1.00000000 1.0194546 0.18724382
2 0.09390665      1 0.32727362 0.4415438 0.10853044
3 0.06037503      2 0.23336696 0.4271241 0.09362895
4 0.03420446      3 0.17299193 0.3842206 0.09030539
5 0.01708278      4 0.13878747 0.3038187 0.07295556
6 0.01695763      5 0.12170469 0.2739808 0.06599642
7 0.01007079      6 0.10474706 0.2693702 0.06613618
8 0.01000000      7 0.09467627 0.2695358 0.06620732
> |
```

```
> print(bodyfat_rpart)
n= 56

node), split, n, deviance, yval
      * denotes terminal node

 1) root 56 7265.0290000 30.94589
   2) waistcirc< 88.4 31  960.5381000 22.55645
     4) hipcirc< 96.25 14  222.2648000 18.41143
       8) age< 60.5 9   66.8809600 16.19222 *
       9) age>=60.5 5   31.2769200 22.40600 *
     5) hipcirc>=96.25 17  299.6470000 25.97000
      10) waistcirc< 77.75 6   30.7345500 22.32500 *
      11) waistcirc>=77.75 11  145.7148000 27.95818
        22) hipcirc< 99.5 3    0.2568667 23.74667 *
        23) hipcirc>=99.5 8   72.2933500 29.53750 *
   3) waistcirc>=88.4 25 1417.1140000 41.34880
     6) waistcirc< 104.75 18  330.5792000 38.09111
      12) hipcirc< 109.9 9   68.9996200 34.37556 *
      13) hipcirc>=109.9 9   13.0832000 41.80667 *
     7) waistcirc>=104.75 7  404.3004000 49.72571 *
>
```

Plot: Decision Tree with Package 'rpart'

With the code below, the built tree is plotted as following:

```
> plot(bodyfat_rpart)
> text(bodyfat_rpart, use.n=T)
```

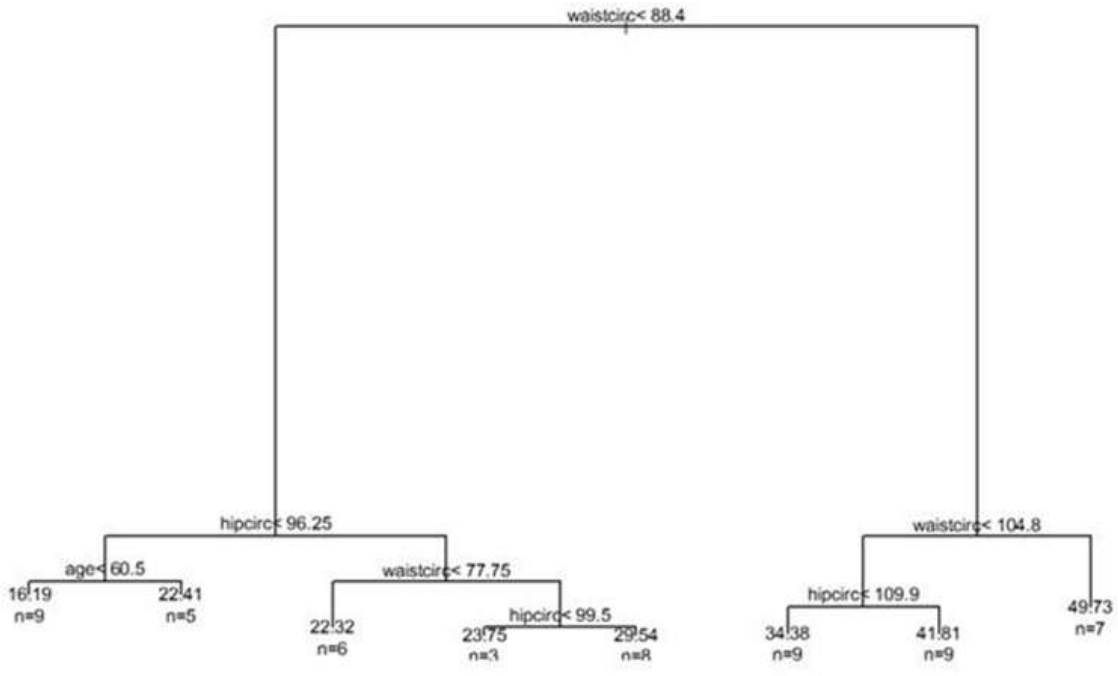


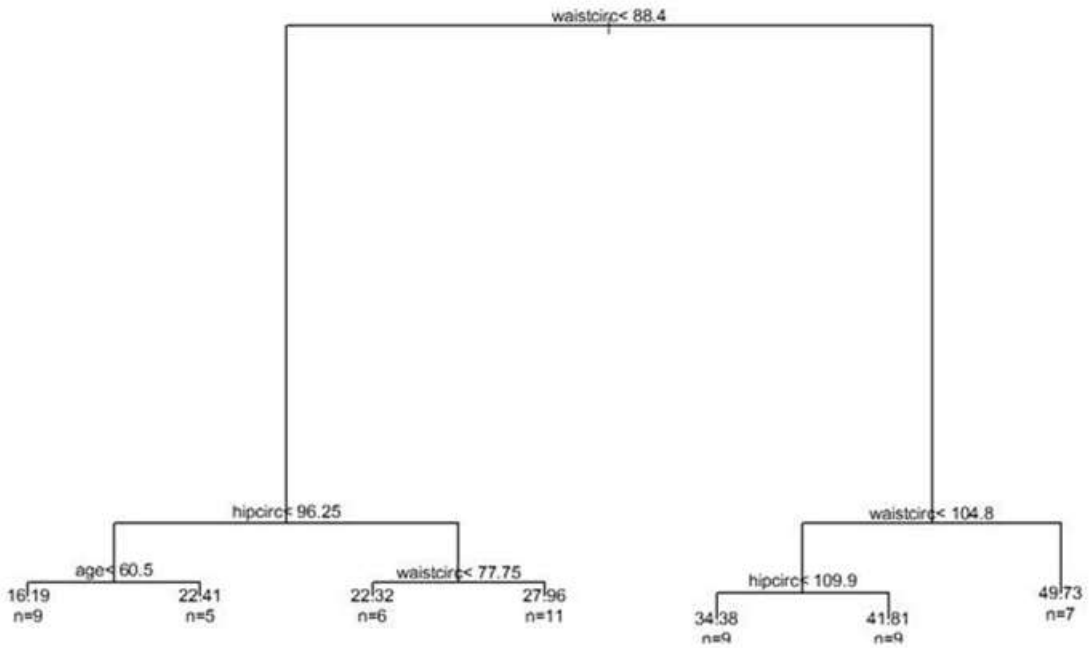
Fig: Decision Tree with Package *'rpart'*

Now we select the tree with the minimum prediction error.

```
> opt<-which.min(bodyfat_rpart$cptable[, "xerror"])
> cp<-bodyfat_rpart$cptable[opt, "CP"]
> bodyfat_prune<-prune(bodyfat_rpart, cp = cp)
> print(bodyfat_prune)
n= 56

node), split, n, deviance, yval
      * denotes terminal node

 1) root 56 7265.02900 30.94589
   2) waistcirc< 88.4 31  960.53810 22.55645
     4) hipcirc< 96.25 14  222.26480 18.41143
       8) age< 60.5 9   66.88096 16.19222 *
       9) age>=60.5 5   31.27692 22.40600 *
     5) hipcirc>=96.25 17  299.64700 25.97000
      10) waistcirc< 77.75 6   30.73455 22.32500 *
      11) waistcirc>=77.75 11  145.71480 27.95818 *
   3) waistcirc>=88.4 25 1417.11400 41.34880
     6) waistcirc< 104.75 18  330.57920 38.09111
      12) hipcirc< 109.9 9   68.99962 34.37556 *
      13) hipcirc>=109.9 9   13.08320 41.80667 *
     7) waistcirc>=104.75 7  404.30040 49.72571 *
>
```

Plot: Selected Decision Tree

```
> plot(bodyfat_prune)
> text(bodyfat_prune, use.n=T)
```



After that, the selected tree is used to make prediction and the predicted values are compared with actual labels. Function abline() draws a diagonal line. The predictions of a good model are expected to be equal or very close to their actual values.

Plot: Prediction Result

```
> DEXfat_pred<-predict(bodyfat_prune, newdata=bodyfat.test)
> xlim<-range(bodyfat$DEXfat)
> plot(DEXfat_pred ~ DEXfat, data= bodyfat.test, xlab="observed",
+ ylab="predicted", ylim=xlim, xlim=xlim)
> abline(a=0, b=1)
```

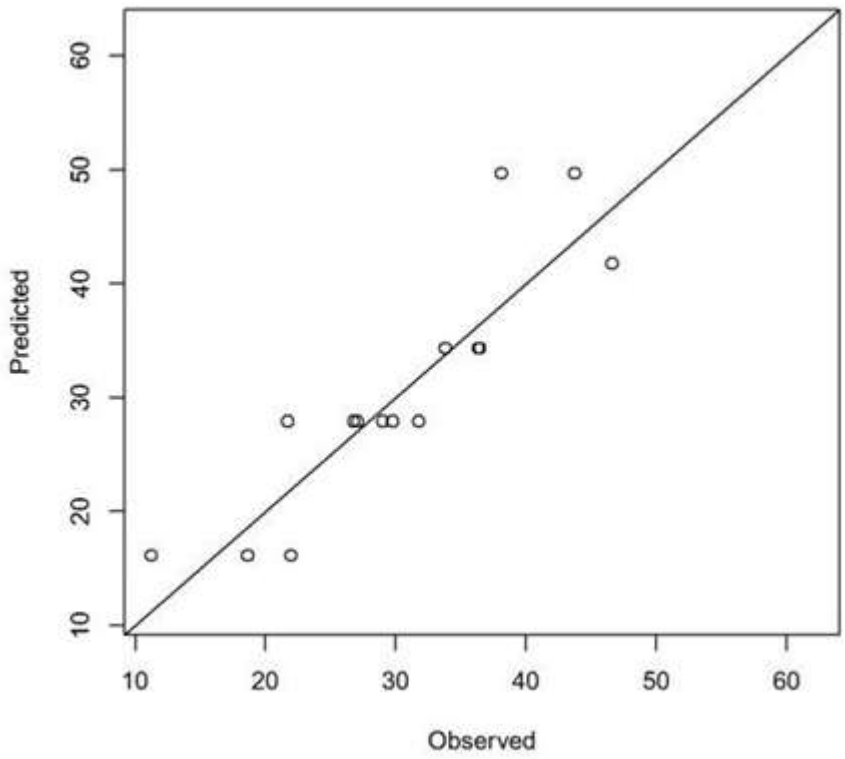


Fig: Prediction Result