Home Work 1 Report
Data Mining and Machine Learning
By : R95922173 伍妮

The goal of HW : Get the experience how to classify data to be training data and testing data in R environment. Find a package to construct tree from training data and try to construct tree and then predict the class label for testing data and analyze the result.

Procedure that I have done :

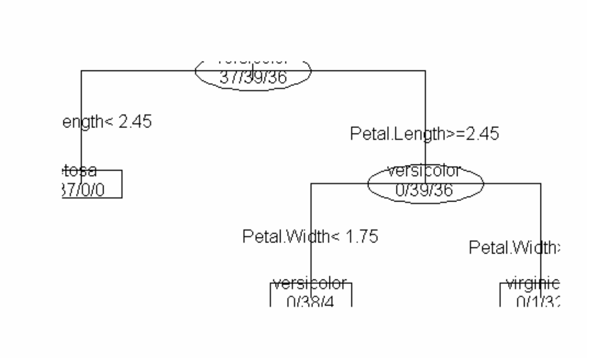
1. Download R from http://cran.csie.ntu.edu.tw/ mirror (R-2.4.1-win32.exe)
2. Install it in windows platform → double click on the icon and follow the instruction
3. Create working directory at "D:\CSIE\2nd semester\data mining\R_work_dir". The working directory is the directory from which `Rgui` or Rterm was launched, unless a shortcut was used when it is given by the `Start in' field of the shortcut's properties.
4. Right click shortcut in desktop and choose properties, change the 'Start in' to "D:\CSIE\2nd semester\data mining\R_work_dir".
5. For English language write LANGUAGE=en at the end of the Target field (*after* any final double quote).
6. Check if installation is not corrupted with run "C:\Program Files\R\R-2.4.1\bin\mdcheck.exe" . The result is "3252 files changed".
7. Download "An Introduction to R" as manual for R
8. Read "An Introduction to R"
9. Load package "rpart" and do training and testing (detail process in next section)

Package to construct decision tree : **rpart package**
Step by step training and testing with iris data using rpart package :

1. Load iris data
   ```
   > data(iris)
   ```
2. Load rpart package, use menu packages -> load package -> select rpart or use command :
   ```
   > library(rpat)
   ```
3. Select a training set randomly (75%) and testing data (25%) from iris data
   ```
   # calculate the number of data
       > x <- nrow(iris)
       > training <- sort (sample (1:x, floor (3*x/4)))
   #  training data will be :
       > training_iris <- iris[training,]
   # to get the test data negate the indices :
       > testing_iris <- iris[-training,]
   ```
4. Construct a tree for the training data
   ```
       > iris_Ctree<rpart(Species ~.,data=iris, subset=training, method=”class”, parms=list(split=”information”))
   ```
   Note : formula all, data = iris, subset = training (indices of training set), method="class" (classification tree)

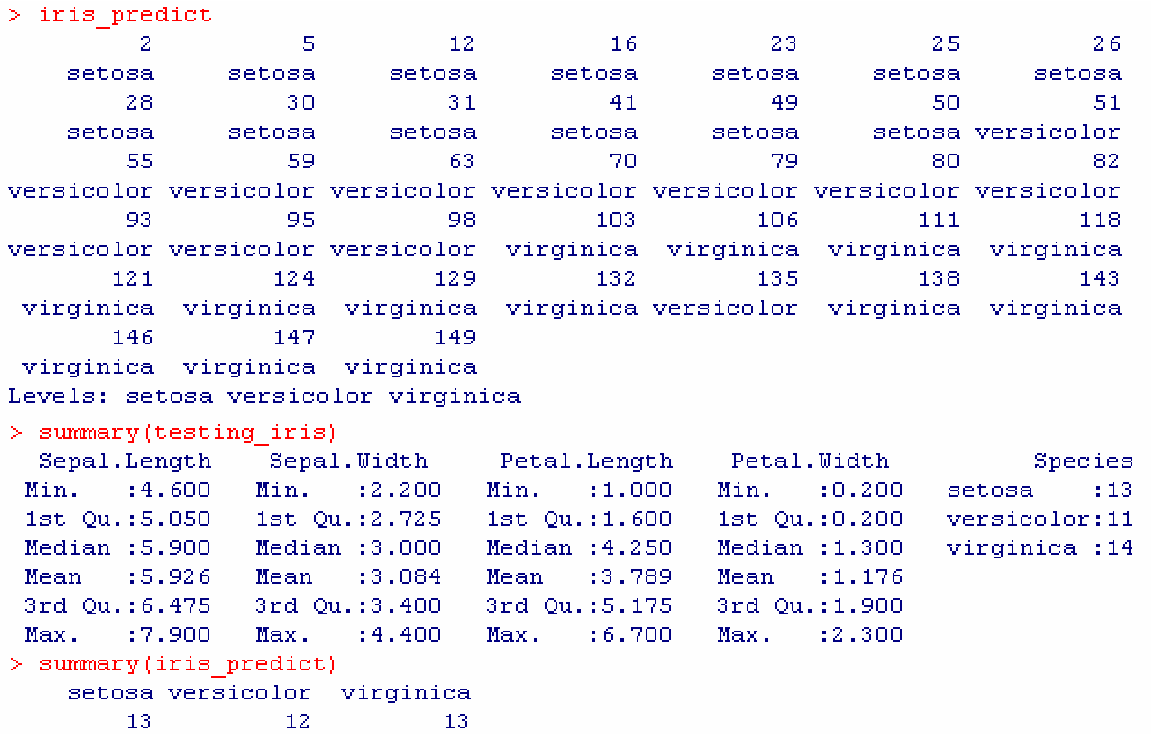
Plot and label classification tree

> plot(iris_Ctree,uniform=TRUE,compress=TRUE,margin=0)
> text(iris_Ctree,use.n=TRUE,all=TRUE,fancy=TRUE)



5. Get the prediction, use testing data (use predict for linear model (lm))
   > iris_predict<-predict(iris_Ctree,newdata=testing_iris,type=”class”)

   result :



```
> iris_predict
         2          5         12         16         23         25         26
    setosa     setosa     setosa     setosa     setosa     setosa     setosa
        28         30         31         41         49         50         51
    setosa     setosa     setosa     setosa     setosa     setosa versicolor
        55         59         63         70         79         80         82
versicolor versicolor versicolor versicolor versicolor versicolor versicolor
        93         95         98        103        106        111        118
versicolor versicolor versicolor  virginica  virginica  virginica  virginica
       121        124        129        132        135        138        143
 virginica  virginica  virginica  virginica versicolor  virginica  virginica
       146        147        149
 virginica  virginica  virginica
Levels: setosa versicolor virginica
> summary(testing_iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.600   Min.   :2.200   Min.   :1.000   Min.   :0.200   setosa    :13
 1st Qu.:5.050   1st Qu.:2.725   1st Qu.:1.600   1st Qu.:0.200   versicolor:11
 Median :5.900   Median :3.000   Median :4.250   Median :1.300   virginica :14
 Mean   :5.926   Mean   :3.084   Mean   :3.789   Mean   :1.176
 3rd Qu.:6.475   3rd Qu.:3.400   3rd Qu.:5.175   3rd Qu.:1.900
 Max.   :7.900   Max.   :4.400   Max.   :6.700   Max.   :2.300
> summary(iris_predict)
    setosa versicolor  virginica
        13         12         13
```

6. Analysis : After I run and try to random training and testing data more than once, the average of testing accuracy is about 97% . Like in this sample analysis, i can conclude that from 38 testing data only 37 data that correct, it means the accuracy is about 37/38 * 100 = 97 % .

	Data for testing (testing_iris)	Prediction (iris_predict)
setosa	13	13
versicolor	11	12
virginica	14	13