PRACTICAL – 06

AIM: Demo of Decision Tree

THEORY:

A decision tree is a flowchart-like tree structure where an internal node represents feature(or attribute), the

branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision

tree is known as the root node. It learns to partition on the basis of the attribute value. It partitions the tree in

recursively manner call recursive partitioning. This flowchart-like structure helps you in decision making. It's

visualization like a flowchart diagram which easily mimics the human level thinking. That is why decision trees

are easy to understand and interpret.

Decision Tree is a white box type of ML algorithm. It shares internal decision-making logic, which is not

available in the black box type of algorithms such as Neural Network. Its training time is faster compared to the

neural network algorithm. The time complexity of decision trees is a function of the number of records and

number of attributes in the given data. The decision tree is a distribution-free or non-parametric method,

which does not depend upon probability distribution assumptions. Decision trees can handle high dimensional

data with good accuracy.

How does the Decision Tree Algorithm Work?

The basic idea behind any decision tree algorithm is as follows:

1. Select the best attribute using Attribute Selection Measures(ASM) to split the records.

2. Make that attribute a decision node and breaks the dataset into smaller subsets.

3. Starts tree building by repeating this process recursively for each child until one of the condition will

match:

• All the tuples belong to the same attribute value.

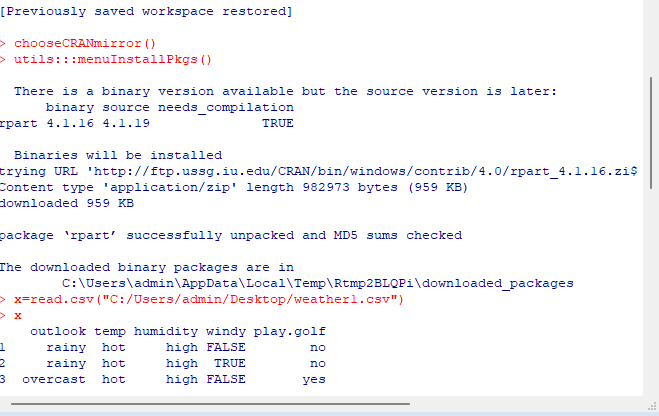
• There are no more remaining attributes.

• There are no more instances.

IMPLEMENTATION AND CODE:

Step1: click on packages and set cran mirror.

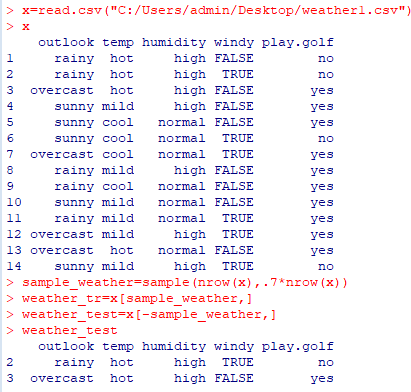
Step2: click on packages and select install packages and install 3 packages (rpart,tree,rattle) Step3:(OPTIONAL Application for version 4.2 )

install.packages("rpart") install.packages("tree") install.packages("rattle")

## Step4: Create an excel data save it with .csv extension.

**Code:**

# Read excel data in rstudio

* x=read.csv("C:/weather1.csv")
* x

# Create sample partition of the excel data

* sample\_weather=sample(nrow(x),.7\*nrow(x))

# Create a weather partition for training

* weather\_tr=x[sample\_weather,]

# Create a weather partition for testing

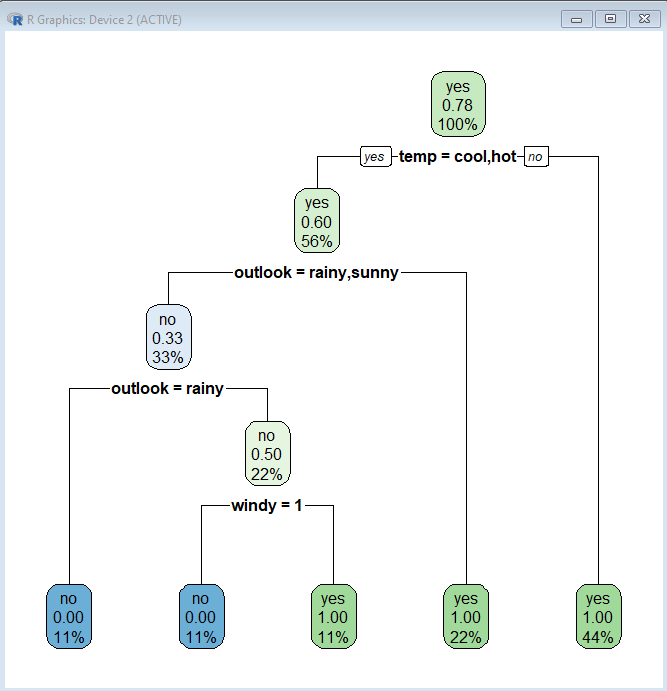
* weather\_test=x[-sample\_weather,]
* weather\_test

# Call rpart packages

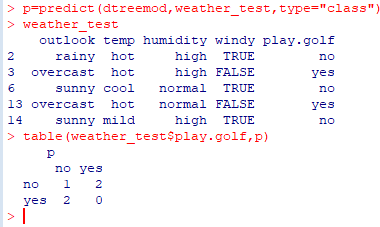
* library(rpart)
* library(rpart.plot)

# Plot tree

dtreemod=rpart(play.golf~.,data=weather\_tr,method="class",control=rpart.control(minsplit=1,min bucket=1))

rpart.plot(dtreemod)

# Predict Tree:

* p=predict(dtreemod,weather\_test,type="class")
* weather\_test
* table(weather\_test$play.golf,p)

## Printing rules with rpart.rules

rpart.rules(dtreemod) play.golf

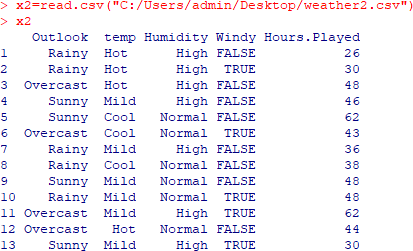
0.00 when temp is hot

1.00 when temp is cool or mild

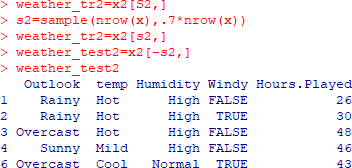
>

# Regression Tree:

* x2=read.csv("C:/Users/admin/Desktop/weather2.csv")
* x2

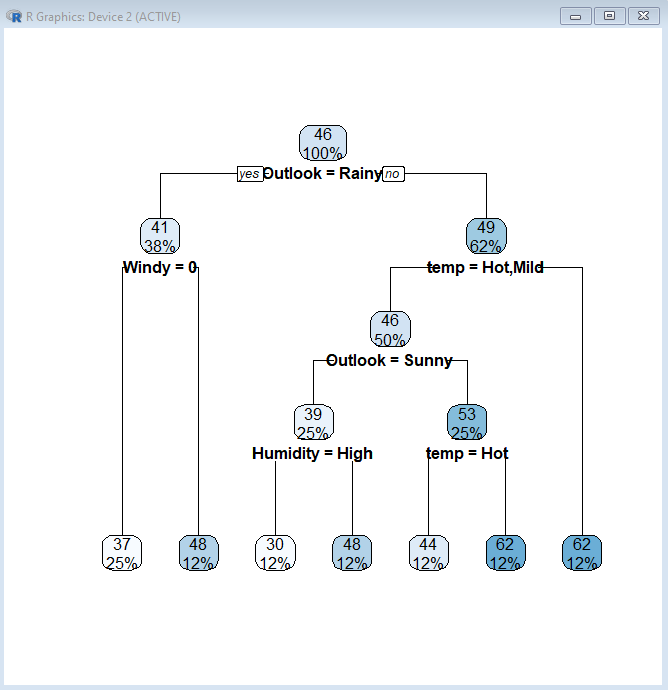


weather\_tr2=x2[S2,]

* s2=sample(nrow(x),.7\*nrow(x))
* weather\_tr2=x2[s2,]
* weather\_test2=x2[-s2,]
* weather\_test2

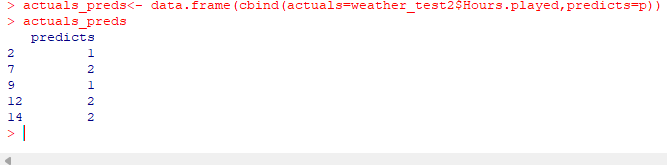
dtreemod2=rpart(Hours.Played~.,data=weather\_tr2,method="anova",control=rpart.control(minsp lit=1,minbucket=1))

* rpart.rules(dtreemod2)



## Prediction:

* actuals\_preds<- data.frame(cbind(actuals=weather\_test2$Hours.played,predicts=p))
* actuals\_preds



CONCLUSION: Hence we successfully implemented Demo of Decision Tree