R Decision Trees – A Tutorial to Tree Based Modeling in R

**Contents**

* 1. Objective
* 2. What is R Decision Trees?
* 3. Applications of Decision Trees
* 4. Principle of Decision Trees
* 5. Building Decision Trees
* 6. CART, C5.0, and CHAID Trees
* 7. Prediction by Decision Tree
* 8. Building Decision Trees in R
* 9. Advantages of R Decision Trees
* 10. Disadvantages of R Decision Trees

**1. Objective**

This blog on **R** Decision Trees will take you through the complete introduction of Decision tree in R. Here we will discuss how to build decision tree in R, the principle of decision tree, what are the applications of Decision tree, what are the types of the decision tree, advantages and disadvantages of R decision trees.

[](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/r-decision-tree.jpg)

## 2. What is R Decision Trees?

One of the most intuitive and popular methods of **data mining** that provides explicit rules for **classification** and copes well with heterogeneous data, missing data, and nonlinear effects is **decision tree**. It predicts the target value of an item by mapping observations about the item.

You can perform either classification or regression tasks here. For example, identifying fraudulent transactions using credit cards would be a classification task while forecasting prices of stock would be regression task.

## 3. Applications of Decision Trees

Following are the common areas for applying decision trees:

* **Direct Marketing –** While the marketing of products and services, business should track products and services offered by the competitors as it identifies the best combination of products and marketing channels that target specific sets of consumers.
* **Customer Retention –** Decision trees helps organizations keep their valuable customers and get new ones by providing good quality products, discounts, and gift vouchers. These can also analyze buying behaviors of the customers and know their satisfaction levels.
* **Fraud Detection –** Fraud is a major problem for many industries. Using classification tree, a business can detect frauds beforehand and can drop fraudulent customers.
* **Diagnosis of Medical Problems –** Classification trees identifies patients who are at risk of suffering from serious diseases such as cancer and diabetes.

## 4. Principle of Decision Trees

You can use decision tree technique to detect the criteria for dividing individual items of a group into n predetermined classes (Often, n=2 represents a balanced tree, which means a largest of two child nodes for each parent node.)

Firstly, a variable is taken as the root node. This variable should best separate the classes. Then, each individual variable is divided into the given classes such that subpopulations called as nodes are generated. Same operation repeats on each new node obtained until no further separation of individuals are possible. Construction is such that each terminal node consists of individuals of a single class.

A tree with each node having no more than two child nodes is called binary tree. The first node is called root and the terminal nodes are known as leaves.

To create a decision tree, you need to follow certain steps:

**i) Choosing a Variable/vector –** Choice of the variable that best separates the individuals of each class depends on the type of decision tree. The Precise criterion for choice of variable and separation condition on a variable depends on the type of tree.

A binary variable allows single separation condition while for a continuous variable having n separate values, there are n-1 possible separation conditions.

The condition for separation is:

X <= mean(xk, xk+1)

When best separation is found, it is applied and then the operation is repeated on each node to increase the discrimination.

The density of node is the ratio of its number of individuals to the total number of population.

**ii)** Once the best separation is found, classes are then split to create child nodes. A variable is taken out at this step also. To choose the best separation, used criteria are:

**iii) The X2 Test –** To test the independence of two variables X and Y, X2 test is used. If  
Oij gives the term on the left-hand side of the equality symbol and Tij gives the term on the right, the test of independence of X and Y is X2, which is calculated by using the following formula:

**The number of degrees of freedom can be calculated as follows:**

p = (number of rows – 1) X (number of columns – 1)

**i) The Gini Index –** It is a measurement of the purity of nodes. It is used for all types of dependent variables and is calculated by using the following formula:

In the preceding formula: fi, i=1, . ., p, are the relative frequencies in the node of the p classes to be predicted.

More evenly distributed the classes are in the node, higher the Gini index will be. As the purity of node increases, Gini index decreases. Gini index measures the probability that 2 individuals picked at random with replacement from a node belong to two different classes.

**ii) Assigning Data to Nodes –** Once the tree is constructed and division criterion has been established, each individual can be assigned to exactly one leaf. This is determined by the values of the independent variables for the individual.

A leaf is assigned to a class if the cost of assigning that leaf to any other class is higher than assigning the leaf to the current class.

The cost is calculated as:

Starting with an error rate of each leaf, error rate of the tree also called as the total cost of tree or risk of the tree is calculated.

**iii) Pruning the Tree**: When you have deep decision trees, you may need to do pruning because they may contain some irrelevant nodes in the leaves. An algorithm is said to be good if it creates a largest-sized tree and automatically prunes it after detecting the optimal pruning threshold. It is necessary to use Cross-validation and combine error rates found for all possible subtrees to choose the best subtree.

It is important to shorten the branches of very deep trees to avoid creating very small nodes with no real statistical significance.

## 5. Building Decision Trees

Decision trees belong to a class of **recursive partitioning algorithms** that are simple to describe and implement. For each decision tree algorithms described earlier, the algorithm steps are as follows:

* You should assess the best way to split data into subgroups for each candidate input variable.
* Then select the best split and divide data into the subgroups defined by the split.
* Now you pick a subgroup and repeat Step 1 for every subgroup.
* You should continue splitting until all the records after a split belongs to the same target variable value or until another stop condition occurs.

The stop condition may be as complicated as a statistical significance test or as simple as a least record count.

Decision trees are nonlinear predictors. This means that the decision boundary between target variable classes is nonlinear. The extent of nonlinearities depends on a number of splits in the tree.

As tree becomes more complex by increasing its depth, more piecewise constant separators are built into the decision boundary to provide nonlinear separation.

Decision trees are based on forwarding selection mechanism; so you cannot visit a split once it is created.

Certain guidelines should be followed for creating decision trees:

* Decision trees incorporate only one variable for each split. So if no variable splits the individuals on its own, decision trees may not start well. Trees need attributes that provide some lift right away. Try including multivariate features as candidates if modeler is aware of it.
* These are considered weak learners or unstable models as small changes in data can produce significant changes in how the tree looks and behaves. Examining competitors to the winning split can be very helpful in understanding how valuable the winning splits are and if other variables may do as well.
* Decision trees are biased toward selecting categorical variables with large numbers of levels. If one has a high number of levels in categorical variables, turn on cardinality penalties or try removing the variables to have fewer levels.
* Trees can run out of data before it finds good models. Because each split reduces the number of remaining records, later splits are based on fewer and fewer records and hence have less statistical power.
* Single trees are often not as accurate as other algorithms in predictive accuracy. This is because of forwarding variable selection and piecewise constant splits of nodes.

Decision Tree Options

* **Maximum Depth –** It determines the number of levels deep a tree can go thereby limiting the complexity of a tree.
* **Minimum Number of Records in Terminal Nodes –** It determines the least number of records allowed in a terminal node. If split results in fewer than the least number of records in of the terminal nodes split is not made and growth in the branch ceases.
* **Yields Clearly Differentiated Clusters**
* **Minimum Number of Records in Parent Node –** It is like least records in a terminal node option but applies to a node where a split can occur. When records are lesser than number of records specified by it, no further split is allowed in this branch.
* **Bonferroni Correction –** It adjusts for many comparisons when the chi-square statistic is computed for a categorical input variable compared to a categorical target variable.

## 6. CART, C5.0, and CHAID Trees

There are three most Common Decision Tree Algorithms:

* Classification and Regression Tree (**CART**), which investigate all kinds of variables.
* 0 (developed by J.R. Quinlan), works by aiming to maximize information gain achieved by assigning each individual to a branch of the tree.
* Chi-Square Automation Interaction Detection (**CHAID**) – It is reserved for the investigation of discrete and qualitative independent and dependent variables.

**CART**is invented in 1984 by L Breiman, JH Friedman, RA Olshen and CJ Stone and is one of the most effective and widely used decision trees. It uses the Gini index to find the best separation of each node.

CART has two primary benefits; generality and performance:

**i) Generality** – It means many categories of the dependent variable can be finite or unlimited, and CART can be used for both classification and prediction – there is an appropriate node splitting criterion for each type of problem.

Generality is enhanced by the capacity to process missing values by replacing each variable with an equally splitting variable(Provide approximately same purity of nodes as an original variable) or an equally reducing variable(Distribute individuals in an approximately same way as an original variable).

**ii) Performance** – The performance of CART is primarily due to its pruning mechanism. Continuing node splitting process constructs Largest trees as far as possible. The algorithm then deduces many nested subtrees by successive pruning operations.

CART has the following drawbacks as well:

* Due to **binary structure** of CART, it produces narrow, deep, and complex decision trees whose reading is difficult to read in some cases.
* CART trees are **biased** in favor of the variables having the largest number of categories, thereby reducing the **reliability**.

The Australian researcher J. Ross Quinlandevelope**C5.0** as an improvement of his earlier trees ID3 and C4.5 but it is less popular than CART tree.

**7. Prediction by Decision Tree**

Decision trees that are for classification can be used for prediction. It achieves this by changing the node splitting criterion. The aim of doing this is that:

* As compared to the parent node, the dependent variable must have a smaller variance in the child nodes.
* The dependent variable must be a mean that is as different as possible from one child node to another.

This means you must choose child nodes that cut intra-class variance and maximize interclass variance.

To explain the above points, let us take an example of a CHAID tree. It divides 163 countries into five groups on the basis of the difference in GNP per citizen. The most discriminating criterion, here, is the energy consumption. The group with medium GNP split again by life expectancy.

**8. Building Decision Trees in R**

You need a package for building decision trees in R. This Is called rpart. It is used for classification by generating a decision and regression trees. This depends on recursive partitioning algorithm. It helps the user identify the structure of data; develop decision rules for decision trees. R builds decision trees as a two stage process as follows:

* Identifies a single variable that best splits data into two groups.
* It applies the above process on each subgroup until the subgroup reach to the minimum size or no improvement in a subgroup is present.

Mcycle dataset shows the relationship between acceleration and time. The aim is to predict acceleration as a function of time. Data is as below:

|  |  |  |
| --- | --- | --- |
| S No | Times | Accel |
| 1 | 2.4 | 0 |
| 2 | 2.6 | -1.3 |
| 3 | 3.2 | -2.7 |
| 4 | 3.6 | 0 |
| 5 | 4 | -2.7 |
| 6 | 6.2 | -2.7 |
| 7 | 6.6 | -2.7 |
| 8 | 6.8 | -1.3 |
| 9 | 7.8 | -2.7 |
| 10 | 8.2 | -2.7 |
| 11 | 8.8 | -1.3 |

Perform the following steps for building an R decision trees for the given motorcycle data.

library(rpart)

Loading the Data of mcycle as below:

data(mcycle)

Creating Scatterplot for acceleration and times as below:

plot(accel~times,data=mcycle)

Generating the Tree and Storing Result in mct object as below:

mct <- rpart(accel ~ times, data=mcycle)

Plot the Graph for the Tree

Plot(mct)

**9. Advantages of R Decision Trees**

Decision trees are one of the most popular statistical techniques. The advantages of using R decision trees are as follows:

* You can understand the results which express as explicit conditions on the original variables. IT staff can program the resulting model, and the execution is faster when this model applies to new individuals. Since the calculations consist of numeric comparisons or inclusion tests depending on whether an individual is a quantitative or qualitative model.
* The decision tree method is non-parametric. This means that the independent variables are not assumed to follow any particular probability distributions. These variables can be collinear. When they are not discriminating, then it does not affect the tree because it does not need to select them.
* The presence of extreme individuals can not affect the Decision trees. These can be isolated in small nodes and do not affect the classification as a whole.
* Decision trees can deal with missing data.**CHAID** treats all missing values of a variable as a category which can either remain isolated or can be merged with another category.
* Some trees such as **CART** and **C5.0** allow variables of all types to handle directly. For example Continuous, discrete and qualitative variables.
* A decision tree provides a visual interpretation of a situation for decision making. So the use of decision trees enhances communication. Branches of the decision tree represent all factors that are important in decision making.

**10. Disadvantages of R Decision Trees**

The disadvantages of using R decision trees are as follows:

* The definition of the nodes at level n + 1 is dependent on the definition of level n. If the condition at level n is true, then only condition at level n+1 will be true.
* The tree detects local, not global, optima. It evaluates all the independent variables sequentially, not simultaneously. The choice of a division for a node at a certain level of the tree is never revised subsequently. This may be troublesome, particularly as some trees (CART) make biased choices.
* The modification of a single variable may change the whole tree if this variable is located near the top of the tree. This results in a lack of robustness. You can overcome it by resampling, in which one can construct the trees on many successive samples and can aggregate by a vote or a mean. But this will lead to losing the simplicity and readability of the model, which are the advantages of decision trees. For example, an individual item has all categories of a group A except the value of a variable that splits the tree. In this case, a variable is misclassified in another group as the tree has tested this particular variable.

## Install R Package

Use the below command in R console to install the package. You also have to install the dependent packages if any.

install.packages("party")

The package "party" has the function **ctree()** which is used to create and analyze decison tree.

## Syntax

The basic syntax for creating a decision tree in R is −

ctree(formula, data)

Following is the description of the parameters used −

* **formula** is a formula describing the predictor and response variables.
* **data** is the name of the data set used.

## Input Data

We will use the R in-built data set named **readingSkills** to create a decision tree. It describes the score of someone's readingSkills if we know the variables "age","shoesize","score" and whether the person is a native speaker or not.

Here is the sample data.

# Load the party package. It will automatically load other dependent packages.

library(party)

# Print some records from data set readingSkills.

print(head(readingSkills))

When we execute the above code, it produces the following result and chart −

nativeSpeaker age shoeSize score

1 yes 5 24.83189 32.29385

2 yes 6 25.95238 36.63105

3 no 11 30.42170 49.60593

4 yes 7 28.66450 40.28456

5 yes 11 31.88207 55.46085

6 yes 10 30.07843 52.83124

Loading required package: methods

Loading required package: grid

...............................

...............................

## Example

We will use the **ctree()** function to create the decision tree and see its graph.

# Load the party package. It will automatically load other dependent packages.

library(party)

# Create the input data frame.

input.dat <- readingSkills[c(1:105),]

# Give the chart file a name.

png(file = "decision\_tree.png")

# Create the tree.

output.tree <- ctree(

nativeSpeaker ~ age + shoeSize + score,

data = input.dat)

# Plot the tree.

plot(output.tree)

# Save the file.

dev.off()

When we execute the above code, it produces the following result −

null device

1

Loading required package: methods

Loading required package: grid

Loading required package: mvtnorm

Loading required package: modeltools

Loading required package: stats4

Loading required package: strucchange

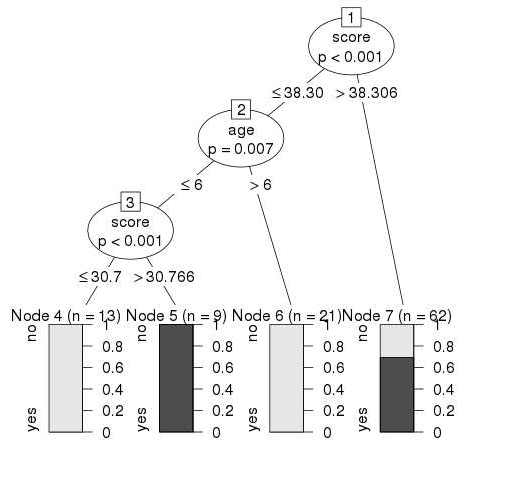
Loading required package: zoo

Attaching package: ‘zoo’

The following objects are masked from ‘package:base’:

as.Date, as.Date.numeric

Loading required package: sandwich



### Conclusion

From the decision tree shown above we can conclude that anyone whose readingSkills score is less than 38.3 and age is more than 6 is not a native Speaker.