# Introduction to Decision Tree - I

# Contents

# Introduction to Decision Tree

- One of the popular predictive modeling techniques, **Decision Tree** uses data mining techniques for predicting a class or value

- Decision Tree **breaks down a data set into smaller subsets and presents association between target variable(dependent) and independent variables as a tree structure.**

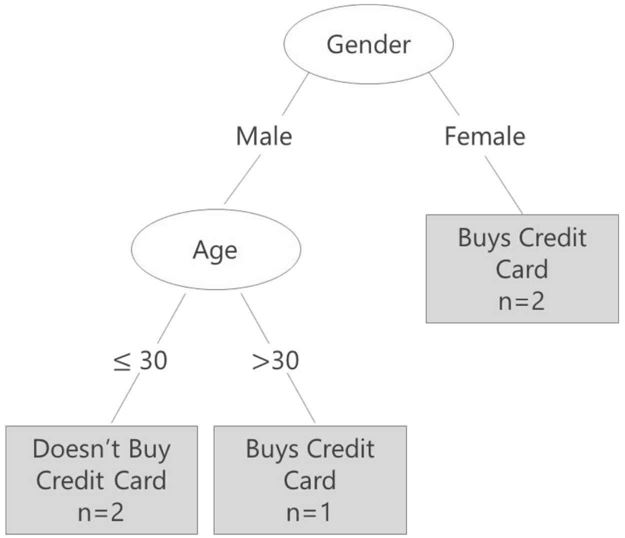
- Final result is a **tree with Decision Nodes and Leaf Nodes.**

- A decision node has two or more branches and leaf node represents a classification or decision.

## Decision Tree – Basic Framework

Suppose we have information about 5 customers and their decision about buying a credit card. This data can be represented as a Decision Tree:

| Customer No. | Gender | Age | Occupation | Buys Credit Card |
|---|---|---|---|---|
| 01 | Male | ≤30 | Student | No |
| 02 | Male | ≤30 | Professional | No |
| 03 | Female | ≤30 | Business | Yes |
| 04 | Male | >30 | Business | Yes |
| 05 | Female | >30 | Professional | Yes |



- First subset is based on **Gender**. **All females opt for credit cards.**
- Males, however, can be split further, based on **Age**. **Male customers of age ≤30 years do not buy a credit card, whereas those >30 do buy cards.**

## What is CART ?

The term **Classification And Regression Tree (CART)** analysis is an umbrella term used to refer to predictive decision tree procedures, first introduced by Breiman et al.

### Classification Tree

When the predicted outcome is the class to which the data belongs

In such cases, **dependent variable is categorical**

Independent variables can be either continuous or categorical or both

### Regression Tree

When the predicted outcome can be considered a real number

In such cases, **dependent variable is continuous**

Independent variables can be either continuous or categorical or both

# Decision Tree Algorithms

Trees used for regression and trees used for classification have some similarities - but also some differences, such as the procedure used to determine where to split.

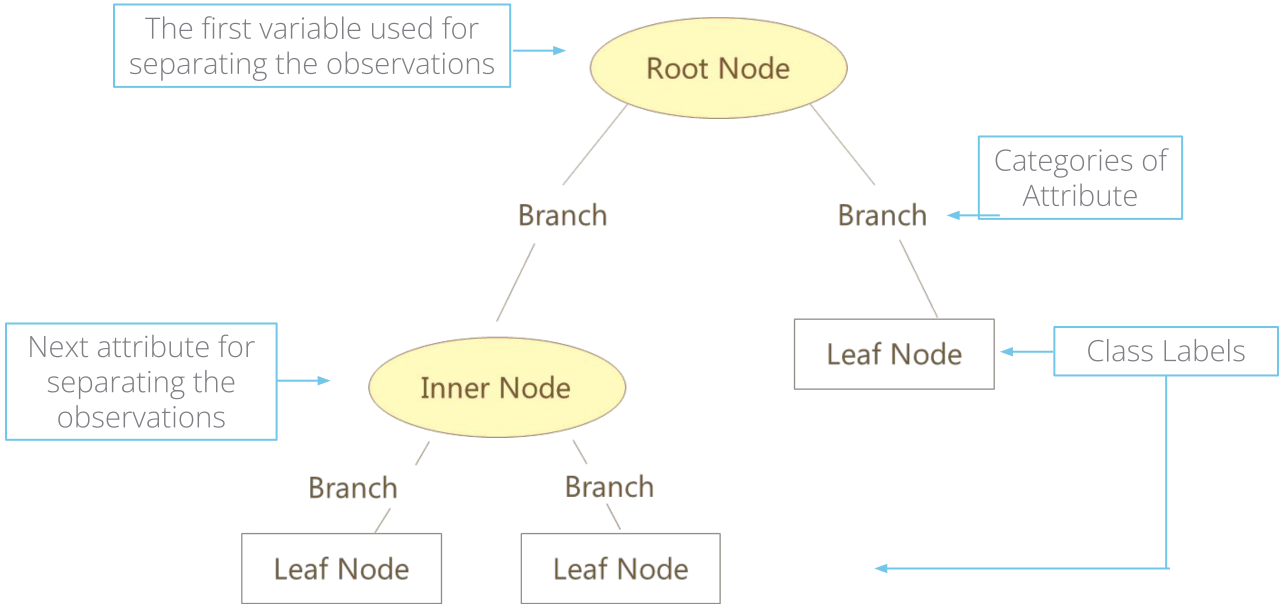
There are many specific decision-tree algorithms. Notable ones include:

1. ID3 (Iterative Dichotomiser 3)
2. C4.5 (Successor of ID3)
3. CART (Classification And Regression Tree)
4. CHAID (CHi-squared Automatic Interaction Detector)
   Performs multi-level splits when computing classification trees
5. MARS (Multivariate Adaptive Regression Splines)
   Extends decision trees to handle numerical data better
6. Conditional Inference Trees
   Statistics-based approach that uses non-parametric tests as splitting criteria, corrected for multiple testing to avoid overfitting. This approach results in unbiased predictor selection and does not require pruning

## Decision Tree – Basic Components

| Component | Description | Alternate terms |
|---|---|---|
| Root node | Has no incoming edges and zero or more outgoing edges | Parent node |
| Inner node | Each has exactly one incoming edge and two or more outgoing edges | Decision nodes / Child nodes |
| Leaf node | Each has exactly one incoming edges and no outgoing edges | Terminal nodes |
| Branches | Categories of attributes | Edges |

# Decision Tree – Basic Components



* Class labels show observations belong to which class. The leaf node also shows Number of observations and Error rate (Actual classification vs classification given by the tree)

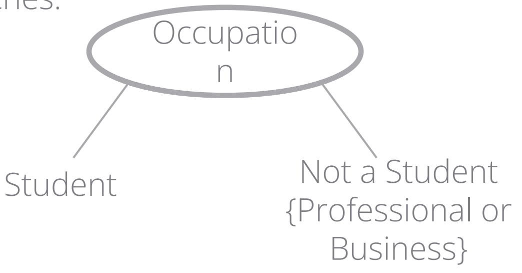
# Binary and Non Binary Trees

Consider the same example illustrated earlier.

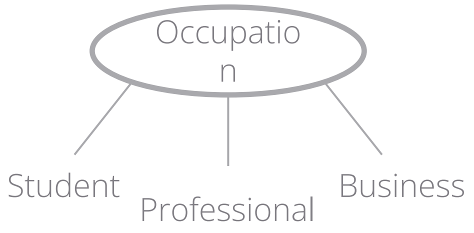
Occupation is the nominal attribute under consideration, with three distinct categories.

| Customer No. | Occupation | Buys Credit Card |
|---|---|---|
| 01 | Student | No |
| 02 | Professional | No |
| 03 | Business | Yes |
| 04 | Business | Yes |
| 05 | Professional | Yes |

There are two ways in which a decision node can be split into further branches:



Binary Tree



Non-Binary Tree

## Introduction to CHAID

The acronym **CHAID stands for Chi Square Automatic Interaction Detection**

CHAID is ,

- Developed in South Africa and was published in 1980 by Gordon V. Kass, who had completed a PhD thesis on this topic
- A type of decision tree technique used for classification problem
- An **exploratory method used to study the relationship between a dependent variable and a series of independent variables**
- **Builds non-binary trees** (trees where more than 2 branches can attach to the root node or any node)
- Can be effectively applied in case of large data sets

# Introduction to CHAID

- Chi-square test of independence is used at each step to determine the strength of relationship between dependent and independent variables.
- The independent variables should be categorical so if data contains continuous predictors then they must be converted to categorical variables.
- Chi-square test is used to measure association between dependent and independent variables as well as amongst independent variables.
- Each pair is assessed to identify the least significantly different via a Bonferroni adjusted p-value which is calculated for merged cross table.

?

Why are we doing this?
The CHAID algorithm is based on Chi-square test of independence, which essentially is a test for checking association between two categorical variables.

# Case Study – Employee Churn Model

## Background

- A company has comprehensive database of its past and present workforce, with information on their demographics, education, experience and hiring background as well as their work profile. The management wishes to see if this data can be used for predictive analysis, to control attrition levels.

## Objective

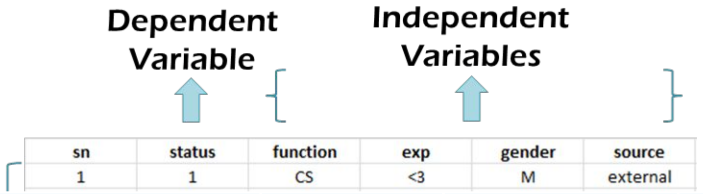
- To develop an Employee Churn model via Decision Tree

## Available Information

- Sample size is 83
- **Gender, Experience Level** (<3, 3-5 and >5 years), **Function** (Marketing, Finance, Client Servicing (CS)) **and Source** (Internal or External) are independent variables
- **Status** is the dependent variable (=1 if employee left within 18 months from joining date)

# Data Snapshot

EMPLOYEE CHURN DATA

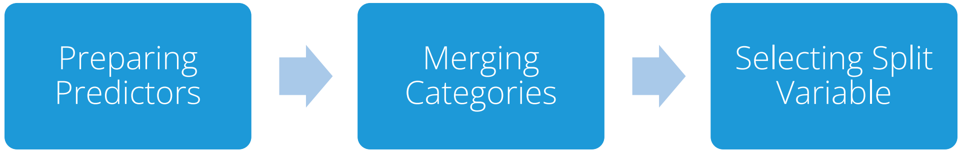

Dependent Variable
Independent Variables

| sn | status | function | exp | gender | source |
|---|---|---|---|---|---|
| 1 | 1 | CS | <3 | M | external |

| Columns | Description | Type | Measurement | Possible values |
|---|---|---|---|---|
| sn | Serial Number | - | - | - |
| status | = 1 If the Employee Left Within 18 Months of Joining<br>= 0 Otherwise | Binary | 1,0 | 2 |
| function | Employee Job Profile | Categorical | CS, FINANCE, MARKETING | 3 |
| exp | Experience in Years | Categorical | <3,3-5,>5 | 3 |
| gender | Gender of the Employee | Categorical | M,F | 2 |
| source | Whether the Employee was Appointed via Internal or External Links | categorical | external, internal | 2 |

# CHAID Algorithm

- CHAID algorithm has three main steps:



## Step 1: Preparing Predictors

- All continuous predictor variables are converted to categorical variables. Example: Experience Level (<3, 3-5 and >5 years)
- The categorical variables are considered with the natural categories
- Example: Function (Marketing, Finance, Client Servicing)

# CHAID Algorithm

Step 2: Merging Categories

- In this step categories of predictors are assessed for possible merging
- For every predictor all possible pairs of categories are considered
- Example: For predictor "function", Marketing-Finance, Marketing-CS and Finance-CS are assessed pairwise
- A chi-squared test of independence is used to decide whether categories in the pair are to be combined or not
- The level of significance value used in the testing is named as alpha-to-merge
- If the adjusted p-value (Bonferroni) for a particular pair is greater than alpha-to-merge, then the categories are merged together
- This merging is continued until no categories can be merged further

# CHAID Algorithm

Step 3: Selecting the Split Variable

- Chi-square test of independence between dependent variable and each of predictors is performed.
- Predictor variable with the smallest adjusted p-value (Bonferroni) is considered as a first split variable and appears at first node.
- If there is tie between 2 predictors then the predictor with largest chi-square statistic value is considered as the split variable.
- The level of significance value is named as alpha-to-split  is decided apriori.
- The process is repeated at each node using subset of data defined by the node.
- If the smallest (Bonferroni) adjusted p-value for any predictor is greater than some alpha-to-split value, then no further splits will be performed, and the respective node is a terminal node.

This process is continued until no further splits can be performed

# Bonferroni Adjustment in CHAID

- Bonferroni adjustment acts as a safeguard against false decisions in case of multiple hypothesis testing problems.
- If n dependent or independent tests are performed simultaneously at level of significance α, then there is a chance of false rejection or acceptance of null hypothesis.
- To control the false rejection/acceptance, α (Probability of type I error) is adjusted to α/n, making the decision rule stricter .
- Instead of adjusting the α value , one can also adjust the p-value obtained for each test .
- $p_{adj}$= min($p_{raw}$*n ,1) , $p_{raw}$ is the p-value for individual test , tested at α.

In CHAID algorithm, both the steps (Merging /Splitting) , include multiple chi-square tests, hence Bonferroni adjusted P-values are used .

## Package “CHAID” in R

```
# Importing the Data
```

```
empdata<-read.csv("EMPLOYEE CHURN DATA.csv",header=T)
```

```
# Decision Tree Using Package "CHAID"
```

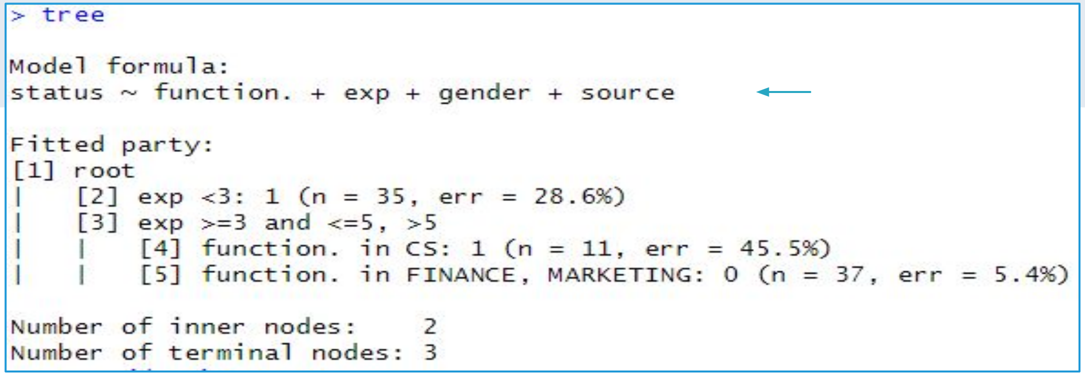
```
install.packages("partykit")

install.packages("CHAID",
                 repos="http://R-Forge.R-project.org",type="source")
library(CHAID)

library(partykit)
```

Package **“CHAID“** is no longer available on CRAN. **“CHAID“** can be downloaded from R-Forge Repository. We also install package **“partykit”** because CHAID is built upon **“partykit”**.

# Package “CHAID” in R

```
# Coverting all variables into factor variables
empdata$status<-as.factor(empdata$status)
empdata$function.<-as.factor(empdata$function.)
empdata$exp<-as.factor(empdata$exp)
empdata$gender<-as.factor(empdata$gender)
empdata$source<-as.factor(empdata$source)
```

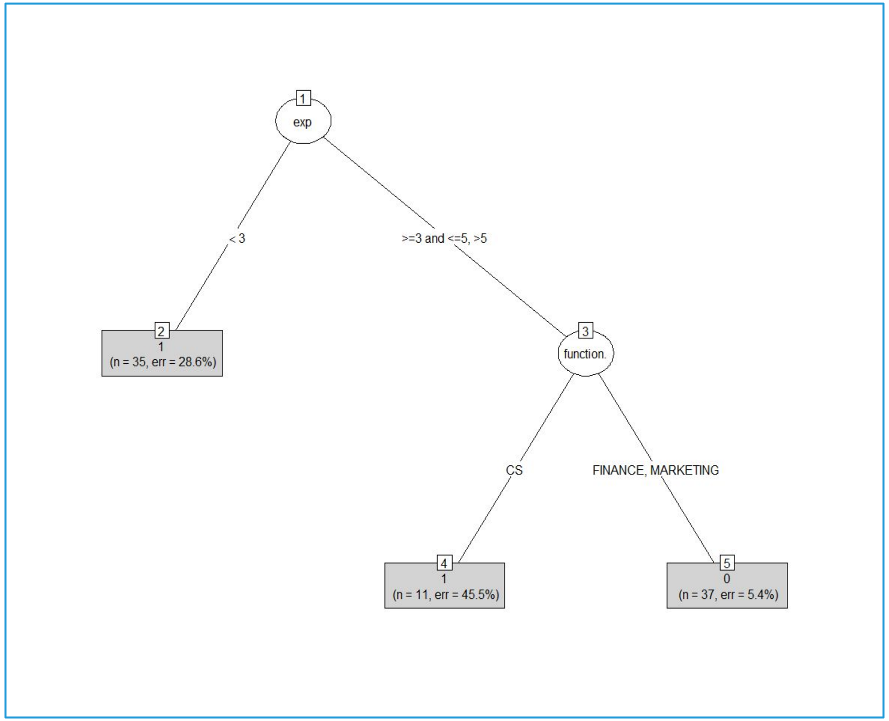
```
# Decision Tree Using Package "CHAID"
tree<-chaid(formula=status~function.+exp+gender+source,data=empdata)
tree
```



```
> tree

Model formula:
status ~ function. + exp + gender + source

Fitted party:
[1] root
|   [2] exp <3: 1 (n = 35, err = 28.6%)
|   [3] exp >=3 and <=5, >5
|   |   [4] function. in CS: 1 (n = 11, err = 45.5%)
|   |   [5] function. in FINANCE, MARKETING: 0 (n = 37, err = 5.4%)

Number of inner nodes:    2
Number of terminal nodes: 3
```

# Decision Tree Using Package “CHAID”

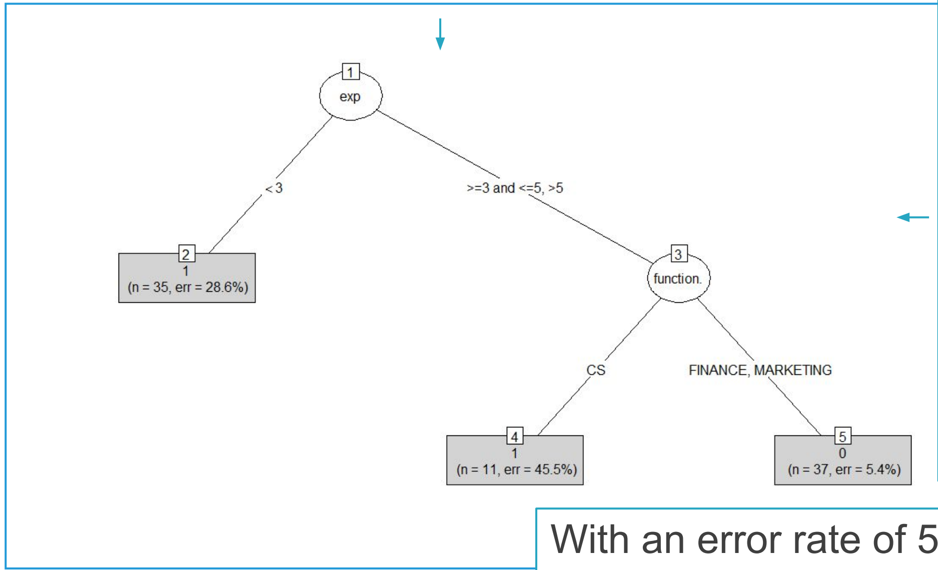
```
# Plotting chaid output
plot(tree,type="simple")
```



There are 35 employees with exp<3. Out of which 25 Left within 18 months. 25/35=0.714. Err=28.6%

## Decision Tree Interpretation – Case Study

Exp is the first split variable. 71% of employees with less than 3 years of experience left the organisation within the first 18 months of joining.



Next split variable is Function. Out of the more experienced employees those working in Client Servicing are more likely to leave.
55% of client servicing employees left the organisation despite having more than 3 years of overall work experience.

With an error rate of 5%, employees with more than 3 years of experience working in the finance and marketing departments are least likely to leave within first 18 months of joining

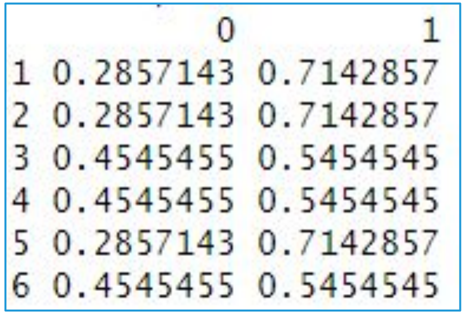
# Predicted Probability and ROC Curve

```
# Predicted Probability
```

```
predtree<-predict(tree,empdata,type="prob")



head(predtree)
```

❑ **predict()** returns predicted values.

```
# Output
```



```
          0         1
1 0.2857143 0.7142857
2 0.2857143 0.7142857
3 0.4545455 0.5454545
4 0.4545455 0.5454545
5 0.2857143 0.7142857
6 0.4545455 0.5454545
```

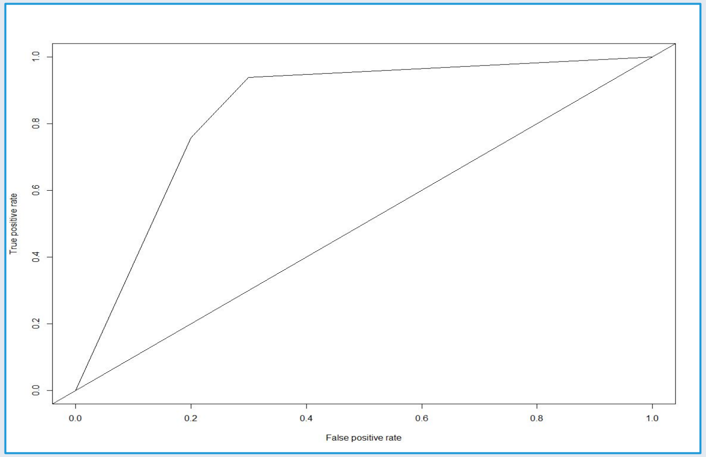
- First column gives probability for Y=0 and second column gives probability of Y=1

# Predicted Probability and ROC Curve

```
#ROC Curve and Area Under ROC Curve
install.packages("ROCR")
library(ROCR)

pred<-prediction(predtree[,2],empdata$status)
perf<-performance(pred,"tpr","fpr")

plot(perf)

abline(0,1)
```

- **prediction()** function prepares data required for ROC curve.
- **performance()** function creates performance objects, "tpr" (True positive rate), "fpr" (False positive rate).
- **plot()** function plots the objects created using performance
- **abline()** adds a straight line to the plot.

# Predicted Probability and ROC Curve



```
auc<-performance(pred,"auc")
auc@y.values
[[1]]
[1] 0.8393939
```

- **The estimated area under ROC curve is 0.8394**

# Quick Recap

| | |
|---|---|
| Decision Tree | • Breaks down a data set into smaller subsets and presents association between target variable(dependent) and independent variables as a tree structure. • A decision tree is made of root node, internal nodes and terminal nodes, joined by branches. |
| Types of Decision Tree | • The term Classification And Regression Tree (CART) analysis is an umbrella term used to refer to predictive decision tree procedures • Classification Tree □ Categorical dependent variable • Regression Tree □ Continuous dependent variable |
| Decision Tree Algorithms | • ID3 (Iterative Dichotomiser 3) • C4.5 (Successor of ID3) • CART (Classification And Regression Tree) • CHAID (CHi-squared Automatic Interaction Detector) • MARS (Multivariate Adaptive Regression Splines) • Conditional Inference Trees |
| CHAID Tree | • **`chaid()`** in package "CHAID" fits a classification tree by CHAID algorithm. |