# Decision Tree Algorithms - I

# Contents

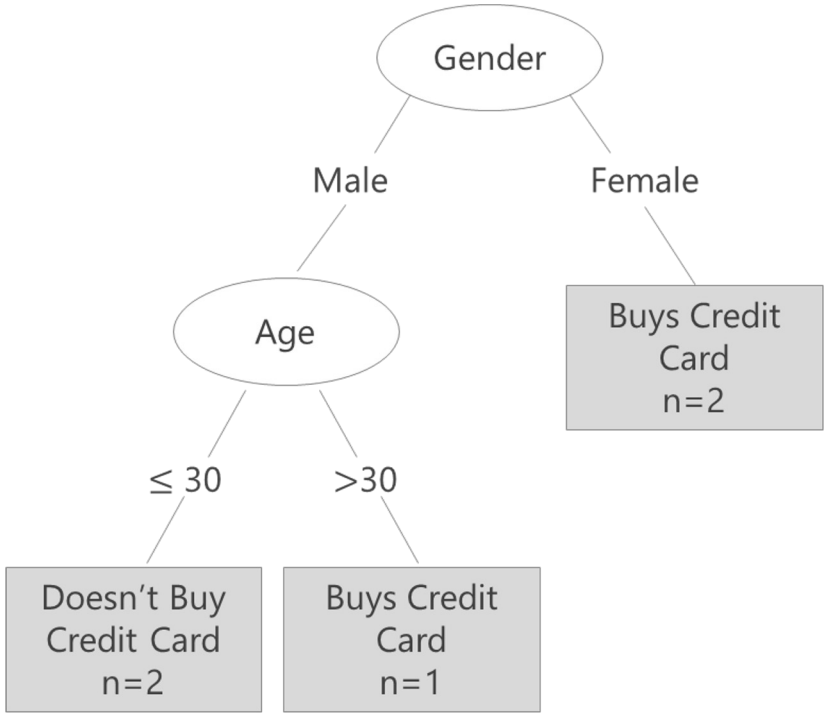
# Introduction to Decision Tree

- One of the most robust predictive modeling techniques, **Decision Tree** uses data mining techniques for model building.
- 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 ?

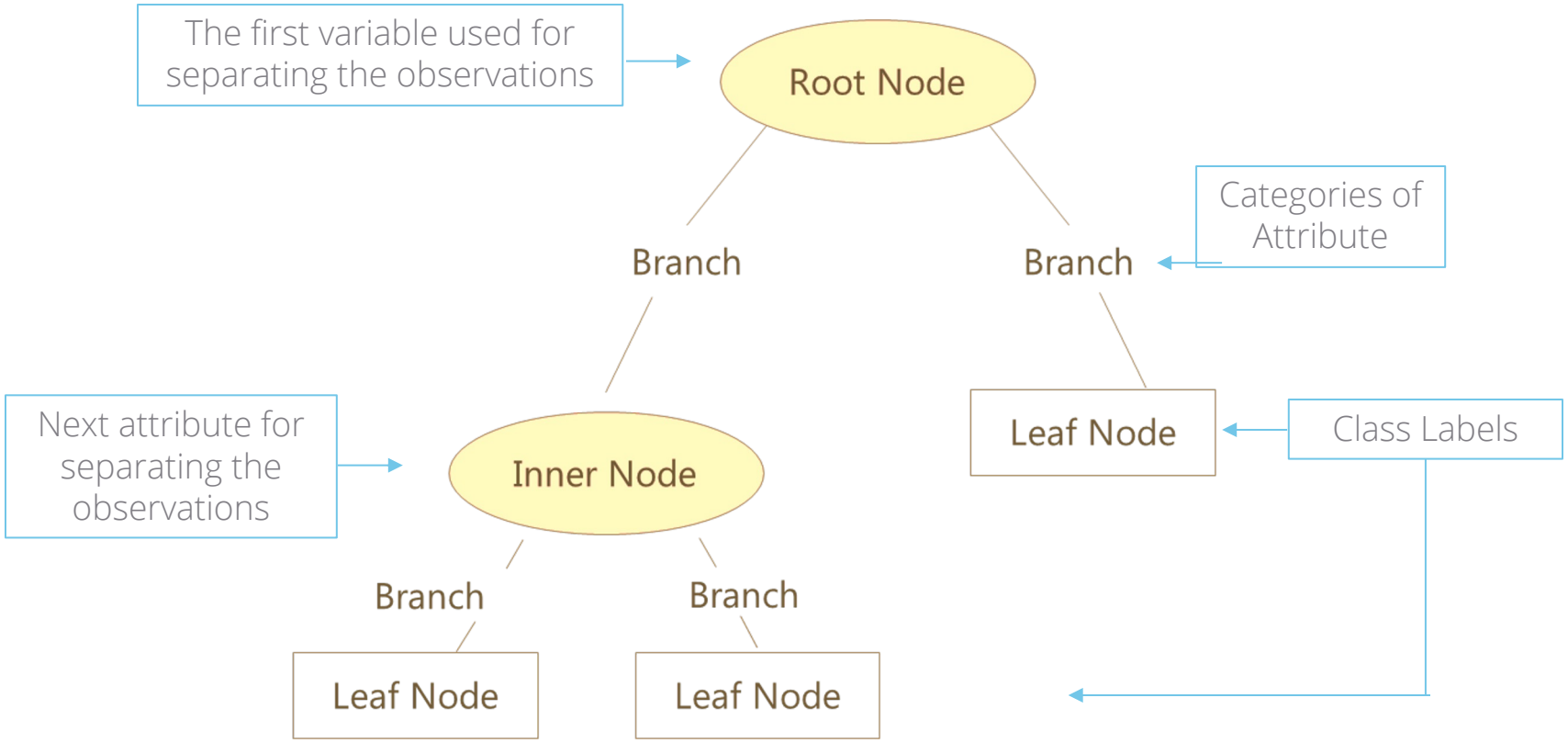
- The term **Classification And Regression Tree (CART)** analysis is an umbrella term used to refer to predictive decision tree procedures, first introduced by Breiman et al.

| Classification Tree | Regression Tree |
| --- | --- |
| When the predicted outcome is the class to which the data belongs | When the predicted outcome can be considered a real number |
| In such cases, **dependent variable is categorical** Independent variables can be either continuous or categorical or both | In such cases, **dependent variable is continuous** Independent variables can be either continuous or categorical or both |

# Decision Tree – Basic Components

| Component | Description | Alternate terms |
|---|---|---|
| Root node | Has no incoming edges and zero or more outgoing edges | Parent node |
| Internal nodes | 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)

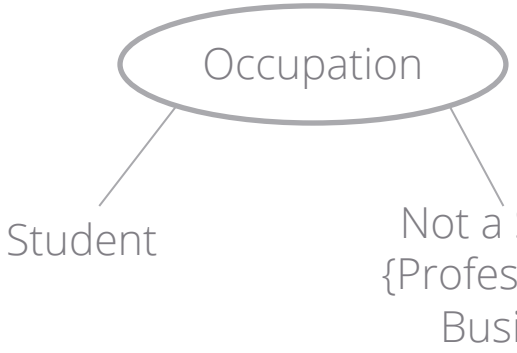
# Binary and Non Binary Trees

Consider the same example illustrated earlier.

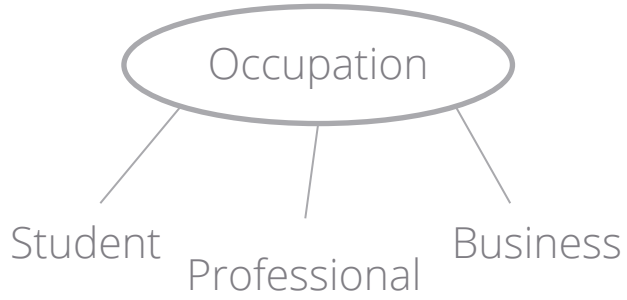
Occupation is the nominal attribute under consideration, with three distinct categories.

| 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

* Binary and non-binary branches can be generated for ordinal and continuous variables as well

# 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

# ID3 Algorithm

- The **Iterative Dichotomiser 3 (ID3)** algorithm is invented by J. R. Quinlan

Uses a **top-down, greedy search method to build a classification decision tree**

Considers a fixed set of examples (training data) to build decision tree and the results are used to classify future observations

Precursor to C4.5 algorithm

# ID3 Algorithm

ID3 searches through the attributes of the fixed set of observations and extracts the attribute that best separates the given examples

If the attribute perfectly classifies the training sets then ID3 stops; otherwise it recursively operates on the n (where n = number of possible values of an attribute) partitioned subsets to get their "best" attribute

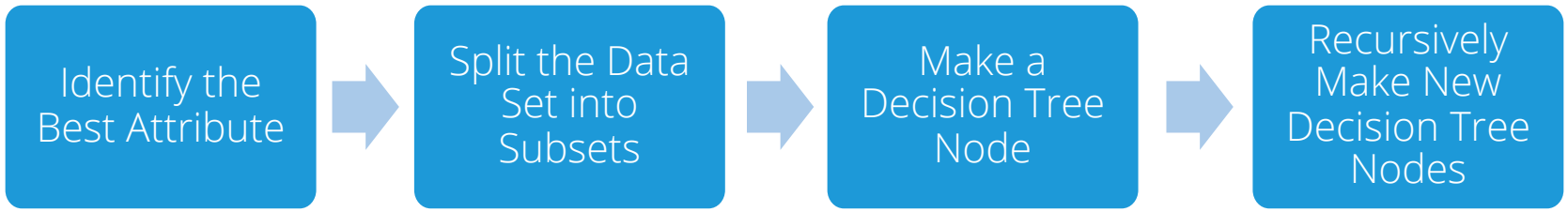
The algorithm uses a greedy search, that is, it picks the best attribute and never looks back to reconsider earlier choices

# ID3 Algorithm

The training data set must have the following properties:

1. Attributes used to describe each observation must be uniform
2. Classification must be predefined for all observations
3. Classes should be discrete
4. The data set should have sufficient observations

Once such data set is obtained, the ID3 employs the following broad procedure




In order to decide which attribute should be included in the decision node, ID3 uses

Entropy and Information Gain

# Entropy

- Entropy measures the homogeneity of a sample. It is used as a parameter for checking the amount of uncertainty associated with a set of probabilities.
- Entropy lies between 0 and 1

  If the sample is completely homogeneous the entropy is 0 and if the sample is equally divided it has entropy of 1
- Entropy can be of two types, for each category and at the variable level
- Entropy of a category is calculated as:

$$-\ P1 * \log_2(P1) -\ P2 * \log_2(P2)$$

where,

P1 is the proportion of class 1

P2 is the proportion of class 2

# Entropy of a Category

Let us consider survey data from three cities depicting shopper's preferred brand

| City | Brand A Voters | Brand B Voters | Number of Voters | | |
|---|---|---|---|---|---|
| Manchester | 90 | 310 | 400 | 22.5% | 77.5% |
| Birmingham | 10 | 90 | 100 | 10% | 90% |
| London | 100 | 100 | 200 | 50% | 50% |

Entropy for each city is calculated as:

Manchester: $- 0.225 * \log_2(0.225) - 0.775 * \log_2(0.775) = \mathbf{0.76919}$

Birmingham: $- 0.1 * \log_2(0.1) - 0.9 * \log_2(0.9) = \mathbf{0.46900}$

London: $- 0.5 * \log_2(0.5) - 0.5 * \log_2(0.5) = \mathbf{1}$

# Entropy at the Variable Level

- Entropy at the variable level can be derived by adding weighted averages of all category level entropy values
- Weights are the proportion of respondents in each category (here in each city)

In the example under consideration,

Weights for the categories are

Manchester: $\mathbf{400/700 = 0.5714}$

Birmingham: $\mathbf{100/700 = 0.1428}$

Chennai: $\mathbf{200/700 = 0.2857}$

Entropy at the variable level is

$$\mathbf{0.57 * 0.76919 + 0.14 * 0.46900 + 0.29 * 1 = 0.79225}$$

# Information Gain

- Information Gain is based on the decrease in entropy after a dataset is split on an attribute
- Constructing a decision tree is about finding attribute that returns the highest information gain

$$\text{Information Gain} = \text{Entropy of Sample (Dependent Variable)} - \text{Average Entropy of Any of the Independent Variable}$$

- Information gain can be interpreted as ability of reducing the uncertainty (Entropy) and hence increase predictability

# Information Gain

| City | Brand A Voters | Brand B Voters | Number of Voters | | |
|---|---|---|---|---|---|
| Manchester | 90 | 310 | 400 | 22.5% | 77.5% |
| Birmingham | 10 | 90 | 100 | 10% | 90% |
| London | 100 | 100 | 200 | 50% | 50% |

Entropy for complete sample is calculated as follows:

P1 = (Total Brand A Voters/Total Voters)

P2 = (Total Brand B Voters/Total Voters)

$$\textbf{Entropy} = -(0.286) * \log_2(0.286) - (0.714) * \log_2(0.714) = \mathbf{0.86312}$$

Information Gain

Entropy at the variable level (Weighted average)

$$0.86312 - 0.79225 = \mathbf{0.070868}$$

# Information Gain and ID3 Algorithm

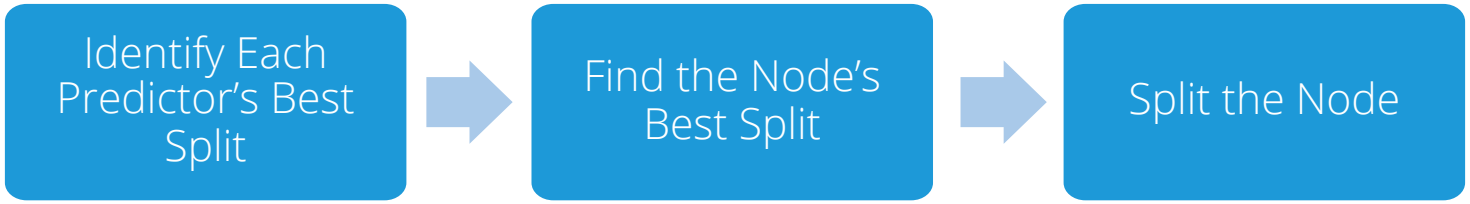
- Let us now go back to the basic ID3 algorithm; Step 1 of which is 'Identify the Best Attribute'


- Information Gain value is used to determine which attribute is the "best" – the attribute with most information gain is chosen
- Information gain for a variable is high when that variable has the low entropy at the variable level (Weighted average)
- Low entropy for a variable implies the classification based on that attribute is fairly homogenous , hence this attribute is selected as the first best attribute
- The same process is repeated till all attributes are used as split variables

# CART Algorithm

- Classification and Regression Tree (CART) algorithm generates a binary decision tree by splitting a node into two branches
- Root node contains the complete sample (training data)
- The splits are univariate – each split depends on the value of only one predictor variable

The algorithm can be divided into three steps:




Gini impurity is used as the splitting criteria in classification problems

# Gini Impurity

- Gini Impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset

$$\text{Gini}\,(t) = 1 - \sum_{i=1}^{c} [p(i|t)]^2$$

where

p(i|t) : Fraction of records belonging to class i at node t

- It reaches its minimum (zero) when all cases in the node fall into a single target category
- Attribute with the smaller Gini Impurity is considered for the split

# Quick Recap

| | |
|---|---|
| Decision Tree Algorithms | • ID3 uses top-down, greedy search method to build a classification decision tree<br>• CART algorithm generates a binary decision tree, by splitting a node into two branches. Root node contains the complete sample. |
| Entropy, Information Gain and Gini Impurity | • Entropy measures the homogeneity of a sample<br>• Information Gain is based on the decrease in entropy after a dataset is split on an attribute<br>• Information Gain= Entropy of Sample − Average Entropy of Any of the Independent Variable<br>• Gini Impurity measures how often a randomly chosen element from the set would be incorrectly labeled |