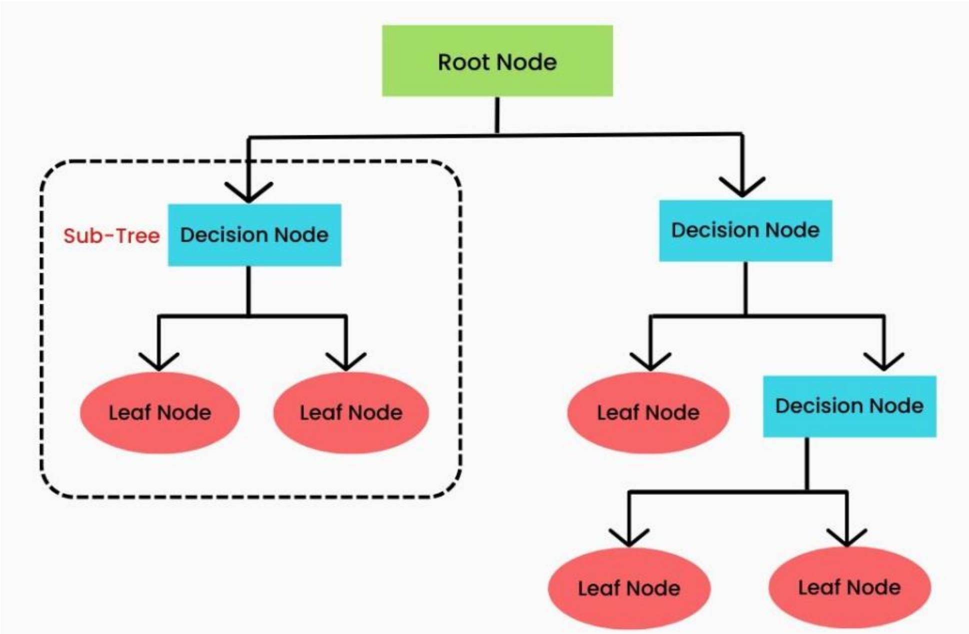
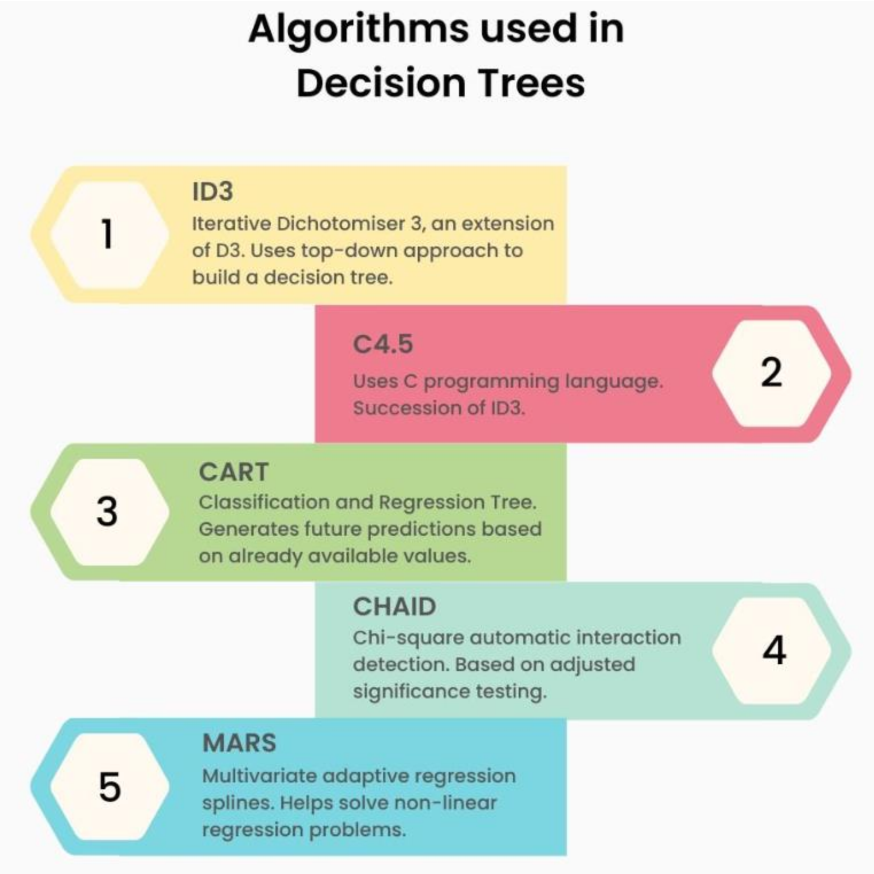
Decision Tree Algorithm

- Decision trees are a **non-parametric supervised learning** algorithm
- They have a hierarchical tree structure consisting of a **root node**, **branches**, **internal nodes**, and leaf **nodes**.
- Decision trees can be used for **classification** as well as **regression problems**.
- Decision trees start with a root node and end with a decision made by leaves.
- **Decisions are made** based on **features** of the given dataset.
- They are build Using **CART Algorithm.**
- Decision trees work by asking questions and splitting the tree into subtrees based on the answers.
- Decision trees are graphical representations of possible solutions to a problem based on given conditions.
- A decision tree can contain **categorical data** (YES/NO) as well as **numeric data**.



Decision tree terminologies:

- **Root node:** Represents the total population or a tiny portion of it. Root nodes can be divided into two or more homogeneous datasets.
- **Decision node:** A sub-node that is further divided into sub-nodes.
- **Pruning:** The process of deleting a sub-node from a decision node.
- **Splitting:** A process to divide a node into different subnodes.
- **Terminal or leaf node:** The nodes that don't split.
- **Parent and child node**: A node divided into sub-nodes. The sub-nodes from the parent node are known as the child node.
- **Branch or sub-tree:** A subset of the entire tree.



ID3 (Iterative Dichotomiser 3)

ID3 or Iterative Dichotomiser 3 is an algorithm used to build a decision tree by employing a top-down approach. The tree is built from the top and each iteration with the best feature helps create a node.

Here are the steps:

- The root node is the start point known as a set S.
- Each iteration of the algorithm will iterate through unused attributes of the root node and calculate the information gain (IG) and entropy (S).
- It will select the attribute with the tiniest entropy or higher information gain.
- We divide set S by choosing the attribute to produce the data subset.
- The algorithm will continue if there is no repetition in the attributes chosen.

Attribute selection measures

- Entropy
- Information gain
- Gini index
- Gain Ratio
- Reduction in Variance
- Chi-Square

Entropy

- Entropy is a measure of the randomness of information.
- Higher entropy means more randomness and harder to solve.
- For example, flipping a coin has high entropy because the outcome is random.
- In ID3, a branch with zero entropy is a leaf node, meaning there is no more information to split on.
- A branch with entropy more than zero will need splitting because there is still randomness to address.

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

where Pi is the probability of an event i from the S state.

Information gain

- Information gain measures an attribute's effectiveness in dividing training instances based on target types.
- Building a decision tree involves finding attributes with the highest information gain and lowest entropy.
- Information gain represents a reduction in entropy.
- It calculates the difference between entropy before and after splitting the dataset based on specific attribute values.
- A higher information gain indicates a more effective attribute for splitting.

$$Information\ Gain\ =\ E(Y)\ -\ E(Y|X)$$

Gini index

- The Gini index measures purity or impurity in decision tree creation using the CART algorithm.
- Comparing attributes with lower Gini indices is only possible against attributes with higher Gini indices.
- The Gini index can only create binary splits, and the CART algorithm utilizes it for the same purpose.
- A cost function based on the Gini index can be used to evaluate splits in the dataset.
- The Gini index is calculated by subtracting the sum of squared probabilities for each class from one.
- It favours large partitions and is simple to implement, but information gain may gain fewer partitions with unique values.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

Gain ratio

- Information gain tends to select attributes with higher values as root nodes, favoring attributes with higher and unique values.
- C4.5, an advancement of ID3, introduces the gain ratio, a modification of information gain that reduces this bias, making it a more balanced choice.
- The gain ratio addresses the issue of information gain by considering the branch count that would result before the split.
- It refines information gain by incorporating the intrinsic information of a split.

$$Gain\ Ratio = \frac{Information\ Gain}{SplitInfo} = \frac{Entropy\ (before) - \sum_{j=1}^{K} Entropy(j,\ after)}{\sum_{j=1}^{K} w_j \log_2 w_j}$$

Reduction in Variance

- Reduction in variance is an algorithm for regression problems that utilizes the standard deviation formula to select the optimal split.
- The split with the lowest variance is chosen as the criterion for dividing the population.

Steps to calculate Variance:

- Calculate variance for each node.
- Calculate variance for each split as the weighted average of each node variance.

$$\text{Variance} = \frac{\Sigma(X - \overline{X})^2}{n}$$

Chi-Square

- CHAID stands for Chi-squared Automatic Interaction Detector, a classification method that identifies statistically significant differences between sub-nodes and parent nodes.
- It measures this significance using the sum of squares of standardized differences between observed and expected frequencies of the target variable.
- CHAID works with categorical target variables like "Success" or "Failure" and can perform multiple splits.
- A higher Chi-squared value indicates stronger statistical significance of differences between sub-nodes and parent nodes.
- CHAID generates a tree structure called CHAID (Chi-square Automatic Interaction Detector) to represent these relationships.

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

Where:

χ^2 = Chi Square obtained
$\sum$ = the sum of
O = observed score
E = expected score

Advantage of Decision Tree

- Easy to interpret and understand
- Relatively robust to outliers and missing data
- Can handle both categorical and numerical data
- Computationally efficient to train and predict with
- Can generate feature importance scores

Disadvantage of Decision Tree

- Can be overfitted
- Sensitive to the order in which the features are split
- Difficult to interpret when they are very deep or complex.
- Need to be careful with parameter tuning.