Short & Sweet (Episode 3)

Decision Tree

Decision Tree is particularly **supervised machine learning algorithm** which helps to analyze and learns from the data and predicts the **classes** or **target** based on the conditional trees built by the algorithm.

By conditional trees, I mean decision tree generate different conditions (like you all know **if else** in python) and based on different conditions, it eventually reaches to the answer i.e. our required target class.

Does **Decision Tree** only use if conditions, so why it is called machine learning algorithm. Isn't it being simple python 🤣🤣???

Obviously Not, Let's see how 🌳🌳 works

How It Works

- Let's assume our example on a simple dataset like below -

Income	Age	Class
30000	35	A
45000	45	B
25000	28	A
70000	48	C
60000	40	B
35000	30	A
90000	55	C
40000	38	B
20000	25	A
75000	50	C
28000	32	A

- Let’s learn common terminologies about tree-
  - **Node**: The fundamental building block of a binary tree that contains data and references to its left and right **child** nodes.
  - **Root**: The topmost node of the tree, from which all other nodes are descended. It's the starting point for traversal.
  - **Parent Node**: A node that has child nodes (direct descendants).
  - **Child Node**: Nodes directly connected to a parent node.
  - **Leaf Node**: Nodes that do not have any children (i.e., they are at the ends of the tree).
  - **Internal Node**: Nodes that have at least one child node.

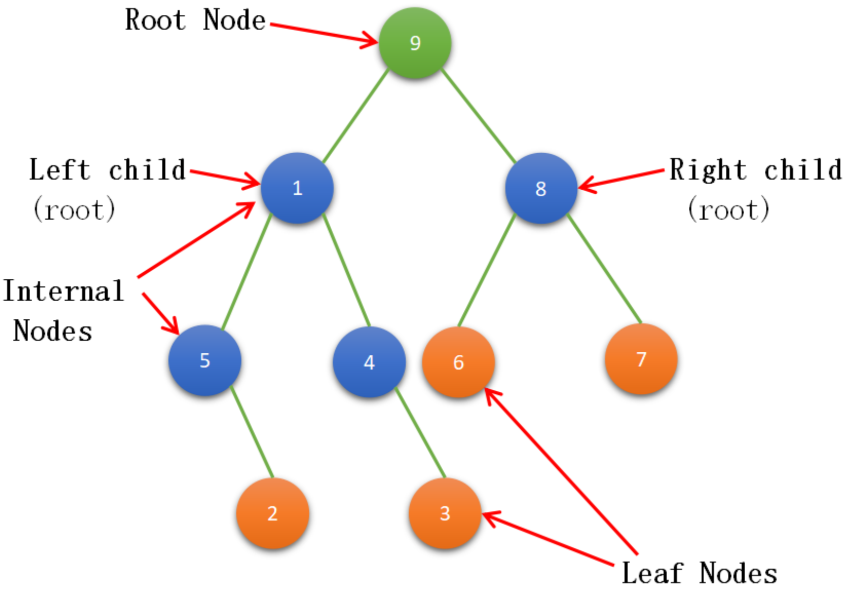
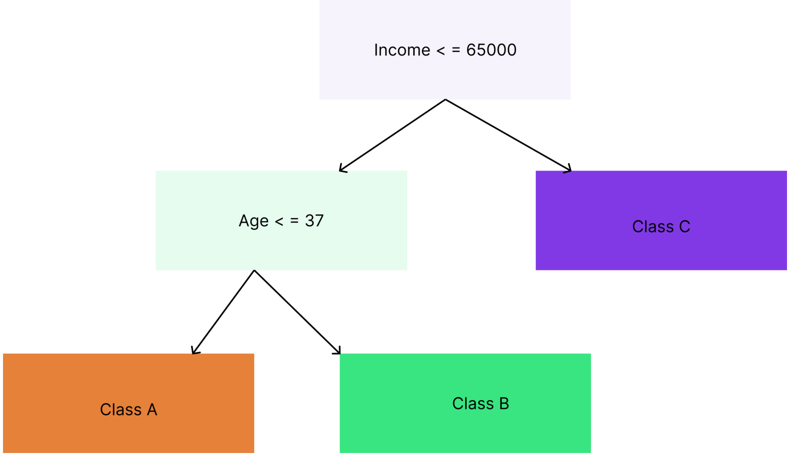


Figure: BINARY TREE

image is credit to tutorialcup.com

- So basically our algorithm tries to create a binary tree (it has many types according to which it can be binary or with more branches) where every node contains a condition like **if (income > 30000)**, then we'll go to class **A** or **C** or **B** then after **age < 45** then finally we get only class **A**. On applying condition, we'll see the probabilities that does this condition leads to any particular class or not which we can called **pure split**.



- Look at the above tree, **root node** and **internal nodes** consist of conditions and all the **leaf node** consist of target class.
- So Decision Tree tries to select the best **condition** (like less than something, greater than something etc.) and select the best **feature**. Here, **information gain** comes into action.
- **Information Gain** measures the effectiveness of a particular attribute in classifying data points within a dataset. With the help of this, we can decide which feature to select. It ranges from 0 to 1.
- **0 indicates no information gain**: If the information gain is zero, it means that the attribute does not provide any improvement in classifying or splitting the data. It doesn't contribute to reducing the uncertainty about the classification of the data points.

- **1 indicates maximum information gain**: An information gain of 1 implies that the attribute perfectly separates or categorizes the data points into their respective classes after the split. It provides the maximum reduction in uncertainty about the classification of the data.
- Formula for **Information Gain (IG)** is :-

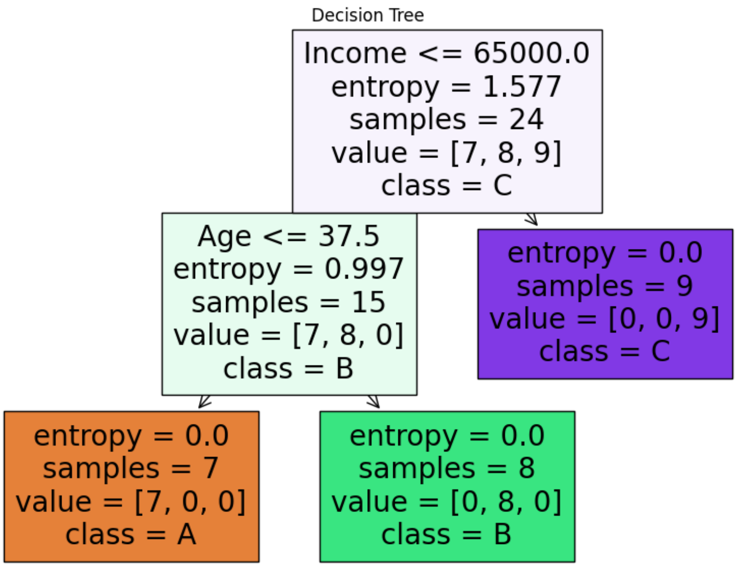
$$\mathrm{IG}(S, A) = \mathrm{Entropy}(S) - \sum_{j=1}^{m} \frac{|S_{v_j}|}{|S|} \cdot \mathrm{Entropy}(S_{v_j})$$

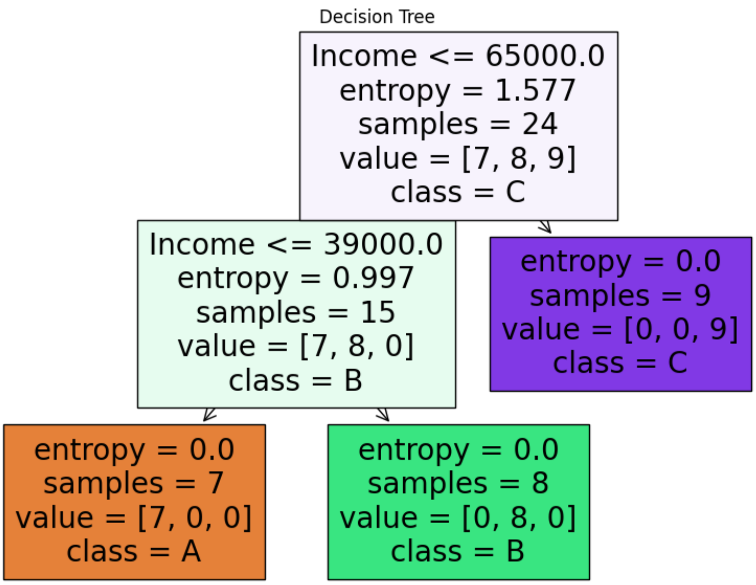
Where:

  - **Entropy(*S*)** represents the entropy of dataset.
  - **IG(*S*,*A*)** represents the information gain of attribute *A* on dataset *S*.
  - ***/ S /*** denotes the total number of samples in dataset *S*.
  - ***Svj*** represents the subset of samples in *S* where attribute *A* has the value ***vj***.
  - **Entropy(*Svj*)** is the entropy of each subset.

Entropy determines the **randomness** of the node in a decision tree. For example, by selecting some suitable condition, if it yields **pure split** means node gives an output of only one particular class, then entropy value is **0** (no randomness) or if it is **1** means it is fully random (it yields output of more classes)

- Look at the different decision tree below :-





- These are the different trees in which information gain is calculated, entropy is calculated.
- Some observations -
  - here in the below leaf nodes, we can see the **entropy** is 0 means pure split - no randomness.
  - **samples** means this much data lies after applying current condition.
  - **value** is a list with length of number of classes at that current node.
- There is one more term - **Gini Impurity** , Go and learn for it.

Assumptions of Decision Tree

The decision tree algorithm operates under several key assumptions:

- **Binary Splits**: Decision trees often perform binary splits on features, meaning they partition the data into two subsets based on a threshold or condition related to a specific attribute. While some variations might allow multi-way splits, many decision tree algorithms default to binary splits.
- **Recursive Partitioning**: Decision trees use a recursive partitioning approach, where they continuously split the dataset based on the selected attributes until a stopping criterion is met. This process involves creating nodes (representing features or attributes) and branches (representing possible outcomes) until reaching a defined depth, purity, or another predefined condition.
- **Feature Independence**: Decision trees assume that features used for splitting at each node are independent of each other. However, some advanced tree algorithms (like Random Forests) can handle correlations between features to some extent.
- **Attribute Relevance**: The algorithm assumes that the attributes or features used for splitting the data are relevant or informative for making predictions or classifications.

- **Noisy Data Handling**: Decision trees can be sensitive to noisy data or outliers, potentially leading to overfitting unless specific precautions, like pruning or using ensemble methods, are taken.
- **Homogeneity within Nodes**: The algorithm assumes that within each node, the data tends to be more homogeneous concerning the target variable (or class label). It aims to partition the data to create more distinct and pure subsets in terms of the target variable.