

37. Decision Tree (Regression)

Decision Tree

- Decision Tree is a **Supervised Learning** technique that can be used for both classification and regression problems, but mostly it is preferred for solving **classification problems**
- In order to build a tree, we can use the **CART algorithm**, which stands for Classification and Regression Tree algorithm
- it splits your data (Binary splitting)
- It works on non-linear splitting data
- It works as a conditional statement

Important Terminology related to Decision Tree


- **Root Node:** It represents the entire population or sample and this further gets divided into two or more homogenous sets
- **Splitting:** It is a process of dividing a node into two or more sub-nodes
- **Decision Node:** When a sub-node splits into further sub-nodes
- **Leaf/Terminal Node:** Nodes do not split further
- **Pruning:** When we remove sub-nodes of a decision node, this is an opposite process of splitting. Some time tree become too big, so the chances of over-fitting. So to avoid over-fitting, we use pruning
- **Branch/Sub-Tree:** A subsection of the entire tree
- **Parent and child node:** A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node

 No description has been provided for this image

In below example, we can split the tree from:

1. company
2. Job
3. Degree

- However, we will consider the factors (which are explained below) to decide from which node, we should start splitting

 No description has been provided for this image

Absolute Selection Measures

This measurement, we can easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are:

1. Information Gain
2. Entropy / Gini Index

Entropy: Entropy is a metric to measure the impurity in a given attribute. it specifies randomness in data.

$$\text{Entropy}(s) = - P(\text{yes}) \log_2 P(\text{yes}) - P(\text{no}) \log_2 P(\text{no})$$

Where:

- S = Total number of samples
- P(yes) = Probability of yes
- P(no) = Probability of no

Information Gain: Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute. It calculates how much information a feature provides us about a class.

$$\text{Information Gain} = \text{Entropy}(S) - [(\text{Weighted Avg}) * \text{Entropy}(\text{each feature})]$$

Gini Index: Gini index is a measure of impurity or purity used while creating a decision tree in the CART (Classification and Regression Tree) algorithm. An attribute with the low Gini index should be preferred as compared to the high Gini Index

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

Where:

- (D) is the dataset
- (p_i) is the proportion of class (i) in the dataset
- (n) is the number of classes

In []:

- It means that your lowest impure data will become decision node
- We prefer less impure data for next splitting
- We will make root node (for example company or Degree) in below example which will have:
 - **Low entropy**
 - **High information gain**



No description has been provided for this image

- In the above example, the node that contains distinct number of either 1 or 0, it is **less impure**
- So we will choose company as a root/parent node because it has high number of 1 and low number of 0
- In other case i.e. Degree, number of 1 and 0 are equal, so we are not sure which value is true, so it is **more impure**
- In above example we have low entropy in case of splitting through company, and
- high entropy in case of splitting through Degree
- So we will choose company as parent/root node for further splitting

So we will calculate entropies of:

1. company
2. Job
3. Degree

- And then decide to start splitting from node which should have **Lowest entropy** (impurity).
- Low entropy means, **high information gain**.
- And vice versa

Algo for doing above task:

- **1st step:** We will calculate entropies of company, job, and degree. In this example, company has lowest entropy, so it will be first decision node, and we will split company into Amazon, Boat, Flipcard
- **2nd step:** We will again calculate entropies of job and degree, and choose the node for further splitting which have lowest entropy, which is degree in this case
- **3rd step:** Only one node left i.e., Job. This would be terminal/leaf node

In []: