# Decision Trees

**Abstract**

This report consists of my basic understanding of one of the popular Ml methods ”Decision Trees”.

## 1 Decision Trees

Decision trees is a supervised machine learning algorithm (where the techniques developed use labelled dataset to classify the data or predict the outcome) used to solve general classification and regression problems.As the name suggests, it uses a tree classifier in which internal nodes represent the features/attributes to be tested and leaf nodes represents the class labels. The method is quite similar to how a normal person do decision making.

## 2 How it is different from Logistic/Linear Regression?

Well as you can see, in logistic/linear regression we are trying to find a best line/curve which fits the data,i.e., we are finding suitable weights which can fit the data. The model starts with assuming that there is linear relationship between independent and dependent variables.The machine learning techniques of this sort are generally known as **Parametric machine learning algorithms**.This is where the red line comes, Decision trees will not assume any kind of relationship between the independent and dependent variables.These kind of techniques are known as **”Non-parametric machine learning algorithms”**. They often used when we don’t have any prior knowledge about training data.

## 3 Understanding Decision Trees

As we said earlier, decision trees uses a tree-structured classifier which means at each node we are asked to choose one path/sub-branch, and the decision we make split the tree into sub trees. This process continues until we end up hitting leaf nodes which are nothing but labels.In the decision trees, we need to know some basic terminology, those are

1) **Root node:**The first node that we will encounter in a decision tree,which further divides into many sub trees.
2) **Leaf node:** The leaf nodes are the final nodes, the tree cannot be segregated further after encountering a leaf node.

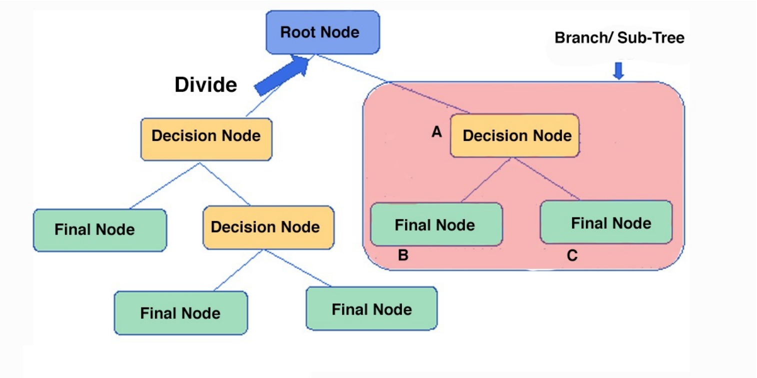

Fig. 3.1: A tree example

3) **Decision node:** At each decision node, we are asked to make a decision, according that we will travel through nodes.

To get better idea you can see Fig 3.1, So basically in decision trees each node represents a feature/attribute of the data and the data splits at every node based on the decision we make until we reach the leaf nodes which are basically the labels predicted by decision trees for the data.It is as simply as it sounds but the only task we have here is to construct a proper decision tree since we can't randomly choose an attribute as a root node.The method we are using to select a attribute over other is known as **"Attribute Selection Measure"**. Now before going to ASM, as we said that it used for Regression it means that labels at leaf nodes are just real numbers so as you might have guessed it can only play some cards when it comes to regression. So it is more preffered for Classification problems than regression.

## 4 Attribute Selection Measure:

In Attribute Selection Measure, we will give a value to each attribute, which represents how important/reliable the attribue is to predict the outcome.The value is known as **"Information Gain"**. Based on this value, the hierarchy of the nodes/attributes is decided.So the attribute with highest information gain will be considered as "root node"and it will be the first split in the tree. Splitting continues until all childeren nodes are **pure** or the information gain is 0.

### *4.1 Information Gain*

Based on the value of information gain we will decide whether some attributes are necessary or not,i.e., we will remove some attribute which has low or 0 information gain.This process is known as **”Pruning”**.Now we will see how to calculate information gain,

$$I\left(T, A\right) = S\ \left(T\right) - \sum_{v \in A} \frac{|T_v|}{T} . S\ \left(T_v\right) \tag{4.1}$$

where,

- T = Target attribute, the feature based on which the data is classified.
- A = The attribute choosen to calculate information gain.
- v = The each unique value in A.
- S($T$) = Entropy of the target attribute.

To understand the (4.1),first we will see what ”Entropy” means in this context.
**Entropy :** With our prior knowledge, we can say that entropy is degree of randomness.To understand that in this scenario, let us take an example, let say we choose 10 persons in that 3 people love anime and 7 people think that it is cartoon. If someone them did they love anime then there will be a ”mixed” answers. In information theory, entropy gives us a way to quantify this ”mixed” thing.It is a measure of how random a data is.If all the 10 people think it is cartoon then the entropy will be 0 since there is no ”mixed” responses.If it is equally distributed then the entropy would be 1 since there is fair chance for one of them to be true.The entropy is formulated as,

$$S\ = -\sum_{i=1}^{c} \Pr\left(X_i\right) \log_2 \Pr\left(X_i\right) \tag{4.2}$$

where,

- The $i$ will iterate over the each unique value in the data.
- $\Pr\left(X_i\right)$ = The probability of the unique value occuring in the data.

The equation (4.1),can also be written as,

$$I = \textbf{Original Entropy}\left(T\right) - \textbf{Weighted sum of Entropies}\left(T_v\right) \left(\text{after the split}\right) \tag{4.3}$$

If $I$ is positive, then overall entropy decreased, which means the randomness is decreased and now we can predict more accurately.So we need check $I$ for every attribute and choose which has more value.

### *4.2 Gini Index*

Similar to what we have discussed in entropy,gini index is another measure by which we split the tree based on randomness/impurity(We will also use impure term in place random).The value of gini index ranges from 0 to 0.5.The algorithm which used gini

index to derive decision trees is known as CART algorithm.
The gini index can be calculated as,

$$Gini\,(T) = 1 - \sum_{i} P_i^2\,(T) \tag{4.4}$$

where, $T$ is the target attribute.
Same as in previous case gini index is calculated for every attribute and we will choose the node with lowest gini index.

## 5 Advantages of decision trees

1) It is easy to understand as it similar to how a human thinks while making a decision
2) Since it will not assume any relationship between independent and dependent variables, it can fit for any data.
3) It is very useful algorithm for decision making

## 6 Disadvantages of decision trees

1) It requires more computation power as the class labels increases.
2) Often has overfitting problem.

## 7 Questions:

1) Explain the structure of Decision trees?
2) List out the different techniques used for ASM?
3) How to select the root node for decision tree using IG ASM?
4) Do we require feature scaling in Decision Trees?
5) How can avoid overfitting in Decision trees?

## 8 Solutions:

1) Decision tree is tree-like classifier in which,
    a) Each internal node is an attribute and we are being asked to make a decision to go to the child nodes.
    b) Each branch from a internal node represents the decision that you made.
    c) Each leaf node is the class label.
2) There are two popular techniques used for ASM,
    a) Using Information Gain and the name of the algorithm is ID3(Iterative Dichotomiser 3).
    b) Using Gini index and the name of the algorithm is CART(Classification and Regression trees)
3) The node which has more information gain value will be considered as root node.

4) In decision trees, we don't require feature scaling because it is a rule based approach and not sensitive to the variance of the data.
5) We can avoid overfitting by doing pre-pruning and post-pruning.
   a) **Pre-pruning:**Pruning the branches before generating the decision tree.
   b) **Post-pruning:** As the name suggests first it will allows the tree to grow to its full depth, then removes the tree branches.