1 What is a Decision Tree?

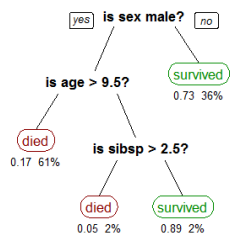
As the name says all about it, it is a tree which helps us by assisting us in decision-making. Used for both classification and regression, it is a very basic and important predictive learning algorithm.

- It is different from others because it works intuitively i.e., taking decisions one-by-one.
- **Non-Parametric**: Fast and efficient.

It consists of nodes which have parent-child relationships

1.1 How it works?

For this let's consider a very basic example that uses titanic data set for predicting whether a passenger will survive or not. Below model uses 3 features/attributes/columns from the data set, namely sex, age and sibsp (number of spouses or children along).



A decision tree is drawn upside down with its root at the top. In the image , the bold text represents a condition/internal node, based on which the tree splits into branches/ edges. The end of the branch that doesn't split anymore is the decision/leaf, in this case, whether the passenger died or survived, represented as red and green text respectively.

Although, a real dataset will have a lot more features and this will just be a branch in a much bigger tree, but you can't ignore the simplicity of this algorithm. The feature importance is clear and relations can be viewed easily. This methodology is more commonly known as learning decision tree from data and above tree is called Classification tree as the target is to classify passenger as survived or died. Regression trees are represented in the same manner, just they predict continuous values like price of a house. In general, Decision Tree algorithms are referred to as CART or Classification and Regression Trees.

1.2 Basic Terminology

- **Root Node**: Entire population or sample, further gets divided into two or more homogeneous (same kind of) sets.
- **Parent and Child Node**: Node which is divided into sub-nodes is called parent node, whereas sub-nodes are the child of parent node.
- **Splitting**: Process of dividing a node into two or more sub-nodes.

- **Decision Node**: A sub-node that splits into further sub-nodes.
- **Leaf/ Terminal Node**: Nodes that do not split.
- **Pruning**: When we remove sub-nodes of a decision node, this process is called pruning. (Opposite of Splitting)
- **Branch/Sub-Tree**: Sub-section of entire tree.

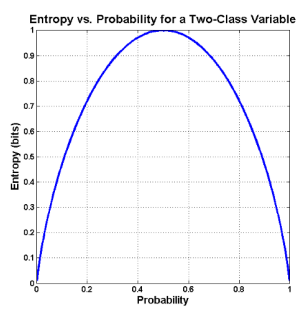
2 Splitting! What is it?

Decision tree considers the most important variable using some criterion and splits dataset based on it. It is done to reach a stage where we have homogenous subsets that are giving predictions with utmost surety. These criteria affect the way tree grows, and thus the accuracy of model. It takes place until a user-defined stopping condition is reached, or perfect homogeneity is obtained. Some of them are:

- **Entropy**: Measure of randomness. More the random data, higher the entropy.

$$S = -p * \log(p) \text{ ; p - probability}$$

- Is a measure of impurity



- **Information Gain**: Decrease in entropy. The difference between the entropy before the split and the average entropy after split is obtained to decide when to split. Formally, the information gain (IG) for a split based on the variable Q.

$$IG(Q) = S_O - \sum_{i=1}^{q} \frac{N_i}{N} S_i,$$

The variable which provides maximum entropy gain is chosen!

where q is the number of groups after the split, Ni is number of objects from the sample in which variable Q is equal to the i-th value.

- **Gini Index**: Measure of variance across all classes of the data. Measures the impurity of the data.

  Ex. Given a binary classification problem, the number of positive cases equals the negative ones. GI = 1/2*(1–1/2)+1/2*(1–1/2) = 1/2. This is maximum GI possible. As we split data, and move towards subtree, GI decreases to zero with increase in depth of tree.

3 Decision Tree Training (ID3 Algorithm)

In decision tree learning, ID3 (Iterative Dichotomiser 3) is an algorithm invented by Ross Quinlan used to generate a decision tree from a dataset.

The ID3 algorithm begins with the original set S as the root node. On each iteration of the algorithm, it iterates through every unused attribute of the set S and calculates the entropy H(S) (or information gain IG(S)) of that attribute. It then selects the attribute which has the smallest entropy (or largest information gain) value. The set S is then split by the selected attribute (e.g. age is less than 50, age is between 50 and 100, age is greater than 100) to produce subsets of the data. The algorithm continues to recurse on each subset, considering only attributes never selected before.

3.1 The algorithm

- Consider the training data and compute the impurity
- At start, all samples are at the root node
- At each step:
  1. Inspect all possible features
  2. Compute the information gain for each
  3. Select the feature that maximizes information gain
  4. Distribute data into child nodes.
  5. Do 1-4 recursively for each child node until pure leaf nodes

4 Decision Tree Algorithms

4.1 Classification and Regression Trees (CART)

Classification and regression trees (CART) are a non-parametric decision tree learning technique that produces either classification or regression trees, depending on whether the dependent variable is categorical or numeric, respectively. Decision trees are formed by a collection of rules based on variables in the modeling data set:

Rules based on variables values are selected to get the best split to differentiate observations based on the dependent variable Once a rule is selected and splits a node into two, the same process is applied to each "child" node (i.e. it is a recursive procedure) Splitting stops when CART detects no further gain can be made, or some pre-set stopping rules are met. (Alternatively, the data are

split as much as possible and then the tree is later pruned.) Each branch of the tree ends in a terminal node. Each observation falls into one and exactly one terminal node, and each terminal node is uniquely defined by a set of rules.

4.2 Recursive Binary Splitting

Growing a tree involves deciding on which features to choose and what conditions to use for splitting, along with knowing when to stop. In this procedure all the features are considered and different split points are tried and tested using a cost function. The split with the best cost (or lowest cost) is selected.

Consider the earlier example of tree learned from titanic dataset. In the first split or the root, all attributes/features are considered and the training data is divided into groups based on this split. We have 3 features, so will have 3 particular splits. Now we will calculate how much accuracy each split will cost us, using a function. The split that costs least is chosen, which in our example is sex of the passenger. This algorithm is recursive in nature as the groups formed can be sub-divided using same strategy.

Due to this procedure, this algorithm is also known as the greedy algorithm, as we have an excessive desire of lowering the cost. This makes the root node as best predictor/classifier.

4.3 Cost Functions

Lets take a closer look at cost functions used for classification and regression. In both cases the cost functions try to find most homogeneous branches, or branches having groups with similar responses. This makes sense we can be more sure that a test data input will follow a certain path.

$$\text{Regression} : sum(y - prediction)^2$$

Lets say we are predicting the price of houses. Now the decision tree will start splitting by considering each feature in the training data. The mean of responses of the training data inputs of particular group is considered as prediction for that group. The above function is applied to all data points and cost is calculated for all candidate splits. Again the split with lowest cost is chosen.

$$\text{Classification} : \text{G} = \text{sum(pk * (1-pk))}$$

A Gini score gives an idea of how good the split is here, pk is proportion of same class inputs present in a particular group. A perfect class purity occurs when a group contains all inputs from the same class, in which case pk is either 1 or 0 and G = 0, where as a node having a 50–50 split of classes in a group has the worst purity, so for a binary classification it will have pk = 0.5 and G = 0.5.

5 Early Stopping and Pruning

Early Stopping: Decision trees are notorius for overfitting. An alternative method to prevent overfitting is to try and stop the tree-building process early, before it produces leaves with very small samples. This heuristic is known as early stopping.

At each stage of splitting the tree, we check the error. If the error does not decrease significantly enough then we stop.

Pruning: As the name implies, pruning involves cutting back the tree. After a tree has been built it may be overfitted. The CART algorithm will repeatedly partition data into smaller and smaller subsets until those final subsets are homogeneous in terms of the outcome variable. In practice this often means that the final subsets (known as the leaves of the tree) each consist of only one or a few data points. The tree has learned the data exactly, but a new data point that differs very slightly might not be predicted well.