Decision Trees

Decision Trees is a non-parametric supervised learning method used for classification and regression. It is referred to as CART or Classification and Regression Trees. The CART model builds regression or classification models in the form of a tree structure. It brakes down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with **decision nodes** and **leaf nodes**. Decision trees can handle both categorical and numerical data. It is a greedy algorithm which makes the locally optimal choice at each stage with the hope of finding a global optimum. It does not in general produce an optimal solution, but yield locally optimal solutions that approximate a global optimal solution in a reasonable time.

Let's take the dataset from the Titanic Kaggle competition as an example. Assume we only consider **three features**: sex, age, total number of siblings and spouses. A decision tree can be drawn from top to bottom as follows. Here we have **three splitting conditions** (internal nodes) in bold black text, based on which the trees split into branches. The end of the branch that **doesn't split any more** is the decision (leaf), 'died' or 'survived' labeled in red or green.

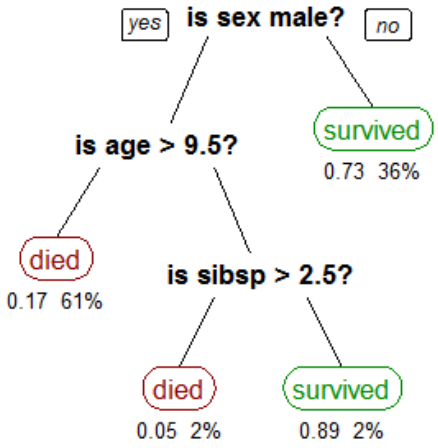

Image taken from wikipedia

The three key factors for the decision trees algorithm are:

1) features (used for splitting)
2) splitting condition
3) when to stop splitting

1) features

All input features are used for splitting. Make sure all features are numerical or categorical before feeding into the decision trees.

2) splitting condition

All input features and possible split points are evaluated and chosen in a greedy manner based on the lowest cost function. For classification, the optimum split is the split with the lowest Gini index gain; for regression, the optimum split is the split with the highest standard deviation reduction.

For classification, Gini index is used to evaluate how pure the nodes are and is mathematically defined as follows:

$$\sum_i p^{(i)}(1-p^{(i)})$$

For example, we have one parent node B split into Node N1 and Node N2. Node B contains 6 C1 and 6 C2, Node N1 contains 5 C1 and 2 C2 while Node N2 contains 1 C1 and 4 C2. The Gini cost of parent and children are calculated as follows:

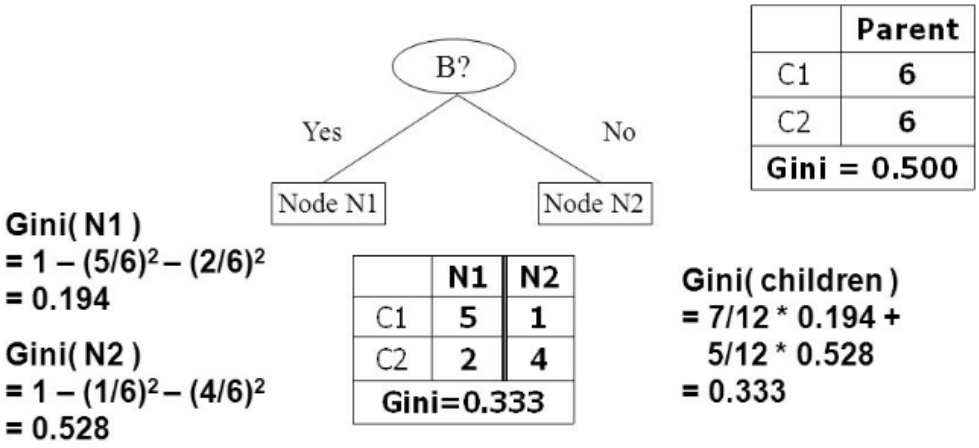

Image from http://slideplayer.com/slide/7696713/, by Jeff Howbert.

For regression, the cost function is the sum squared error (or standard deviation) across all training samples and mathematically defined as follows:

$$\sum_i (y^{(i)} - \hat{y})^2$$

Here $\hat{y}$ is the average y value from the partitioned subgroup.

For example, the squared error of the parent node is calculated as follows:

Hours Played
25
30
46
45
52
23
43
35
38
46
48
52
44
30

$$S = \sqrt{\frac{\sum (x-\mu)^2}{n}}$$

Standard Deviation

S = 9.32

When the training set in the parent node is partitioned based on one feature, the squared error of the child nodes can be calculated as follows:

$$S(T,X) = \sum_{c \in X} P(c)S(c)$$

		Hours Played (StDev)	Count
Outlook	Overcast	3.49	4
	Rainy	7.78	5
	Sunny	10.87	5
			14

S(Hours, Outlook) = **P**(Sunny)***S**(Sunny) + **P**(Overcast)***S**(Overcast) + **P**(Rainy)***S**(Rainy)

= (4/14)*3.49 + (5/14)*7.78 + (5/14)*10.87

= 7.66

If the data has multiple features, multiple S can be calculated and the one with the highest standard deviation reduction (SDR) is chosen for the optimum splitting, as shown in the following example:

		Hours Played (StDev)
Outlook	Overcast	3.49
	Rainy	7.78
	Sunny	10.87
SDR=1.66		

		Hours Played (StDev)
Temp.	Cool	10.51
	Hot	8.95
	Mild	7.65
SDR=0.17		

		Hours Played (StDev)
Humidity	High	9.36
	Normal	8.37
SDR=0.28		

		Hours Played (StDev)
Windy	False	7.87
	True	10.59
SDR=0.29		

Images from http://www.saedsayad.com/decision_tree_reg.htm

3) When to stop

The splitting stops when the minimum number of training example on each leaf or the maximum depth is reached.

Building a decision tree starts from defining terminal nodes (decide when to stop growing a tree), followed by recursively making splits until all nodes are terminal nodes.

The steps for making a split is:

1) calculate the Gini index for a split dataset
2) splitting a node into left child and right child based on a feature and the feature value
3) evaluate all splits and decide the one with smallest Gini score to be the best split

Classification: After building a decision tree using the training set, each sample in the test set is recursively feed into the decision tree until it reaches the leaf node. The node value, which is **the class with maximum counts in that node**, is the predicted class of the test sample.

Regression: The node value, which is **the average value of the training samples** in that node, is the predicted value of the test sample.

References:

1,http://scikit-learn.org/stable/modules/tree.html
2,https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052
3,https://machinelearningmastery.com/implement-decision-tree-algorithm-scratch-python/