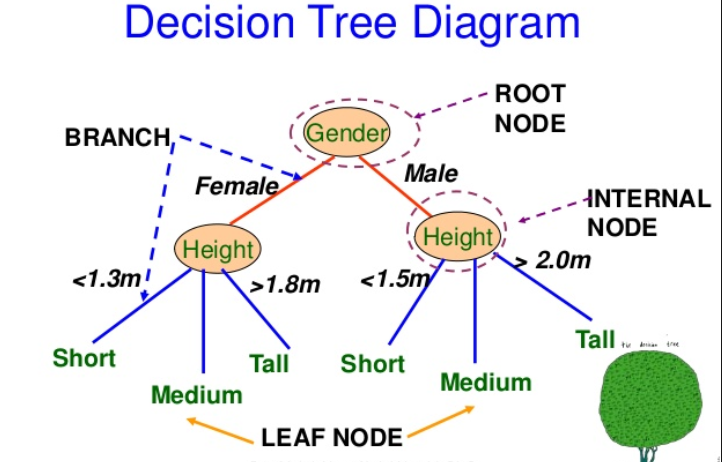
**Decision Tree Modeling in Machine Learning**

A decision tree is a visual representation of a decision situation (and hence aids communication).The branches of a tree explicitly show all those factors that are considered relevant to the decision (and implicitly those that are not).



Courtesy : Google images

Internal node = Decision node; Leaf node = Terminal node

The general motive of using Decision Tree is to create a training model which can use to predict class or value(prediction in regression tree) of target variables by **learning decision rules** inferred from prior data (training data).

### **Key Concepts – Decision Trees**

* **Decision tree topology**: Decision tree consists of following different types of nodes:
  + Root node: Top-most node of the tree from where tree starts
  + Decision nodes: One or more decision nodes which result in splitting of data in multiple data segments. The goal is to have the children nodes with maximum homogeneity (purity).
  + Leaf nodes: The node representing data segment having highest homogeneity (purity).
* A decision tree is built in the top-down fashion.
* **Decision Nodes represent Features/Attributes**: Decision nodes represent the features/attributes based on which data is split into children nodes.
* **Valid decision node or feature**: A decision node or a feature can be considered to be suitable or valid when the data split results in children nodes having data with higher homogeneity or lower entropy (weighted entropy of children nodes added to determine overall entropy after the split).

What is Entropy in decision tree?

* **Entropy**: is lack of order or predictability with data; In other words, data with the high disorder can be said to be data with high entropy, and, homogenous or puredata can be termed as data with very low entropy. Entropy can, thus, be defined as a measure of impurity of data. Higher the entropy, higher impure the data is.

A decision node or a feature can be considered to be suitable or valid when the data split results in children nodes having data with higher homogeneity or lower entropy

* **The leaf node is reached when no significant information gain happens**:

What is Information gain?

**Information Gain**: Information gain is the difference between the entropy of a data segment before the split and after the split. The high difference represents high information gain. Higher the difference implies the lower entropy of all data segments resulting from the split. Thus, higher the difference, higher the information gain and better the feature used for the split.

* A data segment is said to be **pure** if it contains data instances belonging to just one class and not the mix of class. For example, a segment of data consisting of only females can be said to be pure. However, a data segment consisting of 50-50 split can be said to be most impure or entropy of 1. Leaf nodes in the decision tree is reached when further splits do not result in any further information gain. This implies that resultant entropy of partitioned data segments remains almost equal to entropy before the split.

A data segment is said to be ***pure*** if it contains data instances belonging to just one class. The goal while building decision tree is to reach to a state where leaves (leaf nodes) attain pure state.

* **Feature selection in decision trees**: The goal of the feature selection while building a decision tree is to find features or attributes (decision nodes) which lead to split in children nodes whose combined entropy sums up to lower entropy than the entropy value of data segment before the split. This implies higher information gain.

The goal of the feature selection is to find the features or attributes which lead to split in children nodes whose combined entropy sums up to lower entropy than the entropy value of data segment before the split.

* **Pre-pruning vs Post-pruning decision tree**: Once you have understood feature selection technique in relation decision tree, you can end up creating a tree which results in leaf nodes having very high homogeneity (properly classified data). However, this may lead to overfitting scenario. Alternatively, if you have not done enough partitioning, there is a chance of underfitting. This aspect of stopping partitioning the tree early enough is also called as **pre-pruning**. The challenge is to find the optimal tree which results in the appropriate classification with acceptable accuracy. Alternatively, one can let the tree grow enough and then use error rates to appropriately prune the trees.

**Algorithm C5.0** is one of the most used implementations for building machine learning model based on the decision tree.

**Decision tree Vs Linear Regression**

Decision Tree and Linear Regression are both supervised learning algorithms. While decision tree is easy to interpret, linear regression is good when relationships between variables are linear and also when you need to find the marginal effect.

What if relationship between variables are not linear? That is, one variable does not increase or decrease in relation to another variable. Let’s imagine that you fit a line with the classification data points you have. You will not be able to find a suitable straight line that expresses the cause-effect relationship. Hence, linear regression isn’t good for classification models.

**Decision Tree Vs Logistic Regression**

Logistic Regression and trees differ in the way that they generate decision boundaries i.e. the boundaries that are drawn to separate different classes.

Decision Trees bisect the space into smaller and smaller regions, whereas Logistic Regression fits a single line to divide the space exactly into two. Of course for higher-dimensional data, these lines would generalize to planes and hyperplanes. A single linear boundary can sometimes be limiting for Logistic Regression.

When a tree consists of a large number of nodes, it can require a significant amount of mental effort to comprehend all the splits that lead up to a particular prediction. In contrast, a Logistic Regression model is simply a list of coefficients.Because Logistic Regression models are fully described by their coefficients, they are attractive to users who have some familiarity with their data, and are interested in knowing the influence of particular input fields on the objective.

Let’s see benefits and draw backs of decision trees

Benefits:

Decision trees are compatible with both types of tasks — regression as well as classification.

Since decision trees have in-memory classification models, they do not bring in high computation costs (when running individually - not ensemble), as they don’t need frequent database lookups.

1. Decision Trees are easy to explain. It results in a set of rules.
2. It follows the same approach as humans generally follow while making decisions.
3. Interpretation of a complex Decision Tree model can be simplified by its visualizations. Even a naive person can understand logic.

They are extensively used by banks for loan approvals just because of their extreme transparency of rule-based decision-making.

Decision trees have relatively lesser error rates when compared to linear regression for handling non-linear scenarios. They have low bias.

Decision trees can handle data with both numeric and nominal input attributes.

Decision trees are well-known for making no assumptions about the classifier’s structure.

One of the most useful aspects of decision trees is that they force you to consider as many possible outcomes of a decision as you can think of.

1. **Less data cleaning required:**It requires less data cleaning compared to some other modeling techniques. It is not influenced by outliers and missing values to a fair degree.
2. **Data type is not a constraint:**It can handle both numerical and categorical variables.
3. **Non-Parametric Method:**Decision tree is considered to be a non-parametric method. This means that decision trees have no assumptions about the space distribution and the classifier structure.

**Drawbacks:**

There is a high probability of **[overfitting](https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/" \t "_blank)**in Decision Tree. It is susceptible to high variance.

With decision trees working in batches, they model one group of training observations at a time. Hence, they are unfit for incremental learning.

1. Decision tree learners create biased trees if some classes dominate. It is therefore recommended to balance the data prior to fitting with the decision tree.
2. Calculations can become complex when there are many **class labels**.

DT algorithms often tend to produce wrong results if complex, humanly intangible factors are present.

Practical decision-tree learning algorithms are based on heuristic algorithms such as the greedy algorithm where locally optimal decisions are made at each node. Such algorithms cannot guarantee to return the globally optimal decision tree. Can be unstable because small variations in the data might result in a completely different tree being generated.

A drawback of using decision trees is that the outcomes of decisions, subsequent decisions and payoffs may be based primarily on expectations of how things may happen. When actual decisions are made, the payoffs and resulting decisions may not be the same as those you've planned for. It is impossible to plan for all contingencies that can arise as a result of a decision. This can lead to an unrealistic decision tree that could guide you toward a bad decision.

Use cases of decision trees:

* Loan approval
* General business decision-making
* Product planning
* Customer’s willingness to purchase a given product in a given setting, i.e. offline and online both

## 14. To gain more information, refer below:

<https://rstudio-pubs-static.s3.amazonaws.com/27179_e64f0de316fc4f169d6ca300f18ee2aa.html>

<https://github.com/SanyTiger/Decision-Tree-Using-R/blob/master/bank.r>

<https://github.com/tingtingting118/decisiontree-for-income-in-SF/blob/master/8dataset.R>

Interview questions:

<https://www.analyticsvidhya.com/blog/2017/09/30-questions-test-tree-based-models/>

Reference:

<https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/>

https://blog.bigml.com/2016/09/28/logistic-regression-versus-decision-trees/