**Decision Tree:**

* Decision Tree are a non-parametric supervised learning and one of the most popular and powerful algorithm that can be used for classification as well as regression problems in machine learning.
* The decision tree from the name itself signifies that it is used for making decisions from the given dataset. In other words it starts with a root node and ends with a decision made by leaves.
* The concept behind the decision tree is that it helps to select appropriate features for splitting the tree into subparts and the algorithm used behind the splitting is ID3.
* When the target variable takes a discrete set of values are called classification trees and when the target variable takes continuous values (typically real numbers) are called regression trees. Classification And Regression Tree (CART) is general term for this.
* A decision tree is drawn upside down with its root at the top.

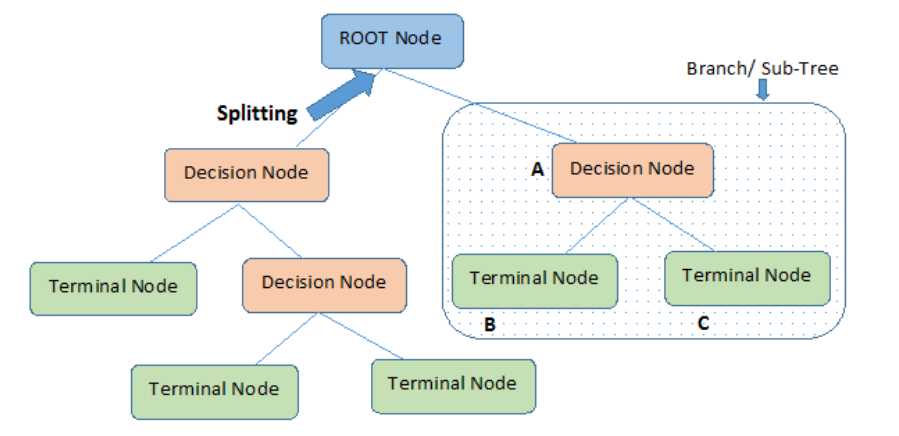
**Terminologies in decision tree:**

**Root Nodes** – It is the node present at the beginning of a decision tree from this node the population starts dividing according to various features.

**Decision Nodes** – the nodes we get after splitting the root nodes are called Decision Node

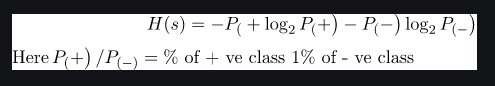
**Leaf Nodes** – the nodes where further splitting is not possible are called leaf nodes or terminal nodes

**Sub-tree** – just like a small portion of a graph is called sub-graph similarly a sub-section of this decision tree is called sub-tree.



**Entropy:**

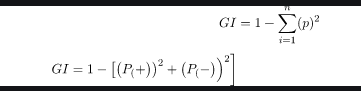
* Entropy helps us to build an appropriate decision tree for selecting the best splitter. Entropy can be defined as a measure of the purity of the sub split.
* Entropy always lies between 0 to 1. The entropy of any split can be calculated by this formula.



The algorithm calculates the entropy of each feature after every split and as the splitting continues on, it selects the best feature and starts splitting according to it.

**Gini Impurity:**

* The internal working of Gini impurity is also somewhat similar to the working of entropy in the Decision Tree. In the Decision Tree algorithm, both are used for building the tree by splitting as per the appropriate features but there is quite a difference in the computation of both the methods.
* Gini Impurity of features after splitting can be calculated by using this formula.



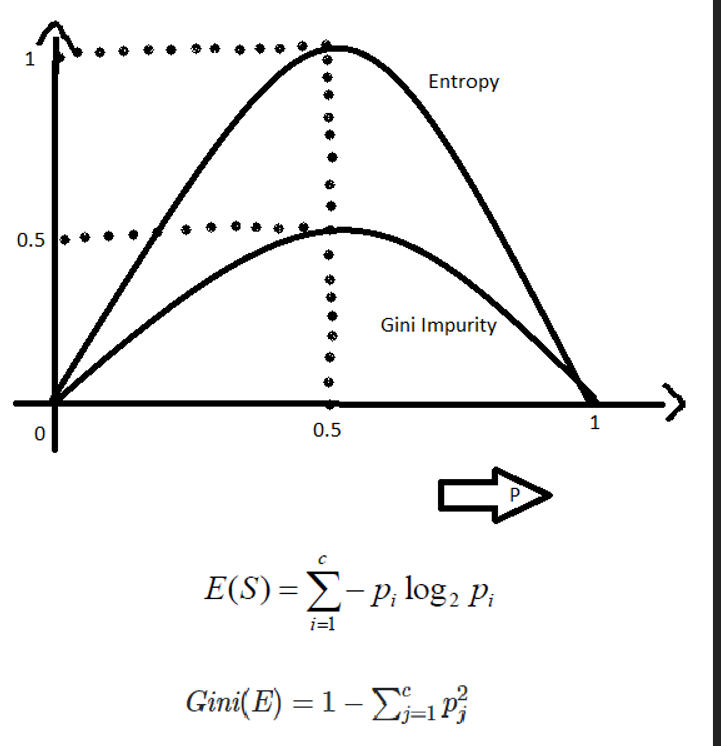
**Information Gain:**

* The information gained in the decision tree can be defined as the amount of information improved in the nodes before splitting them for making further decisions.
* To measure the information gain we use the entropy.



**Gini Vs Entropy:**

|  |  |
| --- | --- |
| Gini Impurity | Entropy |
| * Gini impurities tell us the probability of misclassification of an observation. * Gini has values inside the interval [0, 0.5] * Computationally the calculation of the Gini Index will be faster. | * Entropy is the measure of randomness in the information gain being processed higher the entropy, the harder is to draw conclusions from that information. * The interval of the Entropy is [0, 1]. * Computationally, entropy is more complex since it makes use of **logarithms** |



**Cost of a split:**

* 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.
* **Regression : sum(y — prediction)² - MSE**
  + Let’s say, we are predicting the price of bikes. Now the decision tree will start splitting by considering each feature in 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.
* **Classification : G = sum(pk \* (1 — pk))**
  + A Gini score gives an idea of how good a split is by how mixed the response classes are in the groups created by the split. 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.

**When to stop splitting:**

* One way of doing this is to set a **minimum number of training inputs** to use on each leaf.
* Another way is to **set maximum depth** of your model. Maximum depth refers to the length of the longest path from a root to a leaf.

**Pruning** – is nothing but cutting down some nodes to stop over fitting. There are two types of pruning.

**1. Pre pruning techniques**

* Pre pruning is nothing but stopping the growth of decision tree on an early stage. We can limit parameters
  + **max\_depth**: maximum depth of decision tree
  + **min\_sample\_split**: The minimum number of samples required to split an internal node:
  + **min\_samples\_leaf**: The minimum number of samples required to be at a leaf node.

**2. Post pruning techniques**

* **Cost complexity pruning** is one of the important technique in postpruning.
  + Decision trees can easily overfit. The most effective way is to use post pruning methods like cost complexity pruning.
  + Cost complexity pruning is all about finding the right parameter for alpha.

**Advantages:**

* Simple to understand, interpret, visualize.
* Decision trees implicitly perform variable screening or feature selection.
* Can handle both numerical and categorical data. Can also handle multi-output problems.
* Decision trees require relatively little effort from users for data preparation.
* Nonlinear relationships between parameters do not affect tree performance.

**Disadvantages:**

* Decision-tree learners can create over-complex trees that do not generalize the data well. This is called over fitting.
* Decision trees can be unstable because small variations in the data might result in a completely different tree being generated. This is called variance, which needs to be lowered by methods like bagging and boosting.
* Greedy algorithms cannot guarantee to return the globally optimal decision tree. This can be mitigated by training multiple trees, where the features and samples are randomly sampled with replacement.
* Decision tree learners create biased trees if some classes dominate. It is therefore recommended to balance the data set prior to fitting with the decision tree.

**Resources:**

<https://www.kaggle.com/arunmohan003/pruning-decision-trees-tutorial>

<https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/>

<https://www.xoriant.com/blog/product-engineering/decision-trees-machine-learning-algorithm.html>

<https://www.youtube.com/watch?v=1IQOtJ4NI_0&list=PLZoTAELRMXVPBTrWtJkn3wWQxZkmTXGwe&index=50> 37,38,39 & 40

<https://www.youtube.com/watch?v=mzW66DB48oM>

<https://analyticsindiamag.com/a-complete-guide-to-decision-tree-split-using-information-gain/#:~:text=The%20information%20gained%20in%20the,them%20for%20making%20further%20decisions>.

<https://www.youtube.com/watch?v=SLOyyFHbiqo>

<https://github.com/krishnaik06/Post_Pruning_DecisionTre/blob/master/plot_cost_complexity_pruning.ipynb>

<https://gdcoder.com/decision-tree-regressor-explained-in-depth/>

<https://www.geeksforgeeks.org/gini-impurity-and-entropy-in-decision-tree-ml/>

<https://thatascience.com/learn-machine-learning/gini-entropy/>

<https://victorzhou.com/blog/information-gain/>

<https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8>

# Decision Tree Split For Numerical Feature

* Sorting all values in ascending order
* Set threshold values and calculate entropy, informationgain and gini
* With highest value of Information gain, Threshold value will be selected.

# Decision Tree Split For Continuous variable as Target :

* Decision trees work with continuous variables as well. The way they work is by principle of reduction of variance.
* Let us take an example, where you have age as the target variable. So, let us say you compute the variance of age and it comes out to be x.
* Next, decision tree looks at various splits and calculates the total weighted variance of each of these splits. It chooses the split which provides the **minimum variance**.