**Decision Tree:**

Decision tree also refer as Classification and regression tree is a decision base algorithm that uses tree like structure in which internal node represents the decision based on the feature and each leaf nodes represent a class label where as the path from leaf node to root is called as the rules. Decision tree is also called as nested if else classification/regression model.

**Types of decision trees:**

Decision tree type is based on the type of target feature.

1. **Categorical Variable**: Decision tree is considered to be classifier when the target variable is categorical variable. Ex. YES/NO, LOW/MEDIUM/HIGH.
2. **Continuous Variable**: Decision tree is considered to be regressor when the target variable is continues variable.

**Splitting techniques:**

**Information Gain –**

Information theory is measure to define uncertainty in the data/impurity in a system known as Entropy. Where as the information gain will be low as the uncertainty of a respective dataset/feature increase. If the sample is equally divided (50-50) it has entropy of one and if the data set is completely homogenous then the entropy is zero. Following is the formula to calculate the information gain after splitting any feature.

**Information Gain = Entropy before splitting – Entropy after splitting**

Entropy- Entropy is the uncertainty of a data or Measure of impurity.

Steps to calculate entropy for a split:

1. Calculate entropy of parent node
2. Calculate entropy of each individual node of split and calculate weighted average of all sub nodes.

Derivation of information gain for entropy is done as 1 – entropy.

**Pruning technique:**

Extraction rules with respect to decision tree:

* Support: How frequently the item-set appears in the database. (should be above 20%)
* Confidence: Confidence is an indication of how often the rule found to be true. (should be above 80%)
* Lift: Ratio of the observed support to that expected if X and Y were independent. (1<)

**Gini Index (Gini Impurity) –**

Selection will be done where Gini will be low and perform only binary split.

Steps to calculate the Gini for the split:

1. Calculate Gini for sub nodes using sum of square of probability formula.
2. Calculate Gini for the split using weighted Gini score of each note of that split.

Overfitting & Underfitting:

As the depth of a tree increase the possibility of overfitting increase gradually. Where as if depth is too low the algorithm will be underfit and the change id bias increase.

Algorithm Complexity:

Train time complexity – approx. O (nlog(n)\*d)

Where: n – number of observations

d – number of features

Runtime Complexity - O(D)

Where: D – Depth of a tree

Pros:

* Easy to understand.
* Less fata cleaning required.
* Non linear relation between parameter does not affect tree.
* Support multiple classification.

Cons:

* Easy to overfit.
* Imbalance data impact the entropy calculation.
* Decision tree should not be large.