# Day 7

Agenda :

1. Support Vector Machine (SVM)
    - Geometrical Interpretation
    - Kernel Trick
2. Decision Tree
    - Entropy
    - Information Gain
    - Gini Index
    - CART, ID3, C4.5
3. Code for classification of iPhone purchase
4. code for churn prediction
5. Hyperparameter Tunning
6. MCQs

## Decision Tree

-- Classification is a two-step process, learning step and prediction step, in machine learning. In the learning step, the model is developed based on given training data.

-- Decision Tree is one of the easiest and most popular classification algorithms to understand and interpret.

-- Decision tree algorithm can be used for solving regression and classification problems too.

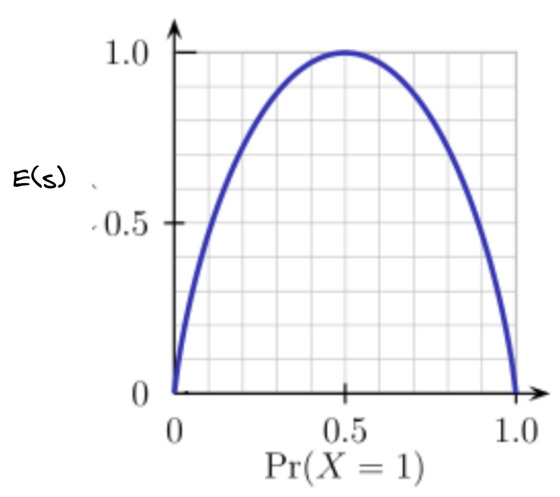
lets discuss some terminologies then will connect the dots:

### 1. Entropy

-- Entropy is a measure of the randomness in the information being processed.
-- The higher the entropy, the harder it is to draw any conclusions from that information.

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Where S → Current state, and Pi → Probability of an event i of state S or Percentage of class i in a node of state S




Example:

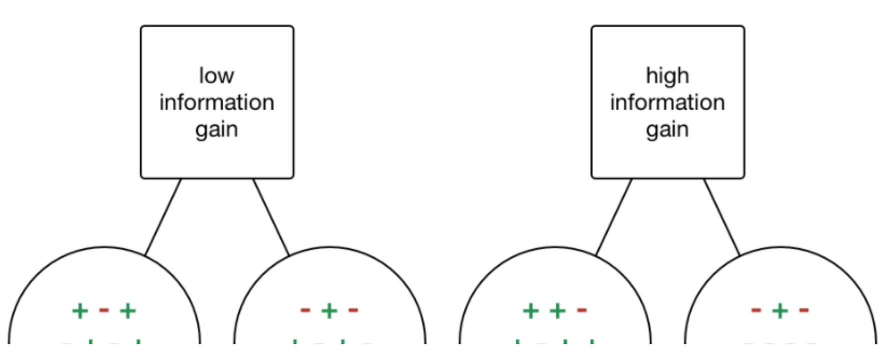
| Play Golf | |
|---|---|
| Yes | No |
| 9 | 5 |

**Entropy(PlayGolf) =** Entropy (5,9)
= Entropy (0.36, 0.64)
= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)
= 0.94

Key Points:
-- If classes are equally distributed then entropy will be minimum.
-- if classes are randomly distributed then entropy will be maximum.

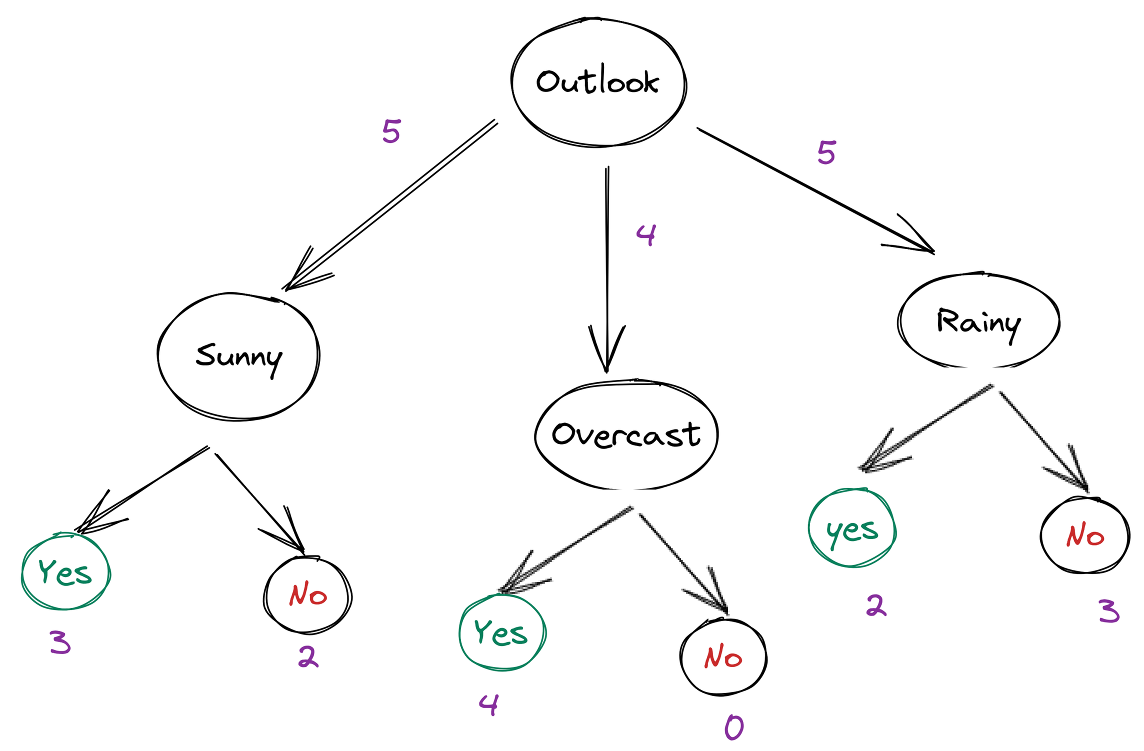
### 2. Information Gain

--- Information gain or IG is a statistical property that measures how well a given attribute separates the training examples according to their target classification.
--- Constructing a decision tree is all about finding an attribute that returns the highest information gain and the smallest entropy.



IG = [Entropy (Parent)] - [Weighted Avg entropy of child nodes]

| | | Play Golf | | |
|---|---|---|---|---|
| | | Yes | No | |
| Outlook | Sunny | 3 | 2 | 5 |
| | Overcast | 4 | 0 | 4 |
| | Rainy | 2 | 3 | 5 |
| | | | | 14 |




**E**(PlayGolf, Outlook) = **P**(Sunny)***E**(3,2) + **P**(Overcast)***E**(4,0) + **P**(Rainy)***E**(2,3)

= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971

= 0.693

where **T→ Current state and X → Selected attribute**

IG(PlayGolf, Outlook) = E(PlayGolf) - E(PlayGolf, Outlook)
= 0.940 - 0.693
= 0.247

## 3. Gini Index/ Impurity

---- its is similar to entropy.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

---- Gini Index works with the categorical target variable "Success" or "Failure". It performs only Binary splits
---- Higher value of Gini index implies higher inequality, higher heterogeneity.

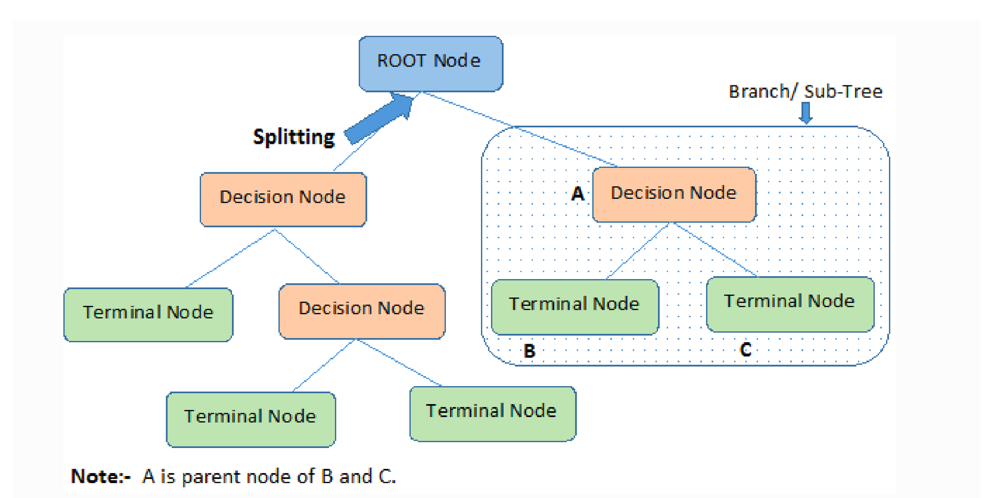
Information gain is biased towards choosing attributes with a large number of values as root nodes.
It means it prefers the attribute with a large number of distinct values

## Gain Ratio:

$$Gain\ Ratio = \frac{Information\ Gain}{SplitInfo} = \frac{Entropy\ (before) - \sum_{j=1}^{K} Entropy(j,\ after)}{\sum_{j=1}^{K} w_j \log_2 w_j}$$

## Some other Terminology related to Decision Trees:

1. Root Node: It represents the entire population or sample and this further gets divided into two or more homogeneous sets.
2. Splitting: It is a process of dividing a node into two or more sub-nodes.
3. Decision Node: When a sub-node splits into further sub-nodes, then it is called the decision node.
4. Leaf / Terminal Node: Nodes do not split is called Leaf or Terminal node.
5. Pruning: When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.
6. Branch / Sub-Tree: A subsection of the entire tree is called branch or sub-tree.
7. Parent and Child Node: A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.




## Assumptions while creating Decision Tree

-- In the beginning, the whole training set is considered as the root.

-- Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model.
-- Order to placing attributes as root or internal node of the tree is done by using some statistical approach

## Steps :

1. It begins with the original set S as the root node.
2. On each iteration of the algorithm, it iterates through the very unused attribute of the set S and calculates the Entropy(H) and Information gain(IG) of this attribute.
3. It then selects the attribute which has the smallest Entropy or Largest Information gain.
4. The set S is then split by the selected attribute to produce a subset of the data.
5. The algorithm continues to recur on each subset, considering only attributes never selected before.

⇒ Decision trees use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that the purity of the node increases with respect to the target variable.

### 1. ID3 :

1. Compute the entropy for the data set
2. for each and every attribute calculate below:
   a) Calculate Entropy
   b) take information gain
3. pick the highest IG value attributes
4. repeat until convergence.
5. Handles only categorical data.

### 2. C4.5:

1. Handles both categorical and numerical data.
2. Error-based punning is used
3. Handle missing values also

### 2. CART :

1. Same steps as ID3
2. Handles both Categorical and numerical data
3. Handles missing values
4. Gini index/gain is being used.

## ⇒ How to avoid/counter Overfitting in Decision Trees?

The common problem with Decision trees, especially having a table full of columns, they fit a lot. Sometimes it looks like the tree memorized the training data set. If there is no limit set on a decision tree, it will give you 100% accuracy on the training data set because in the worse case it will end up making 1 leaf for each observation. Thus this affects the accuracy when predicting samples that are not part of the training set.

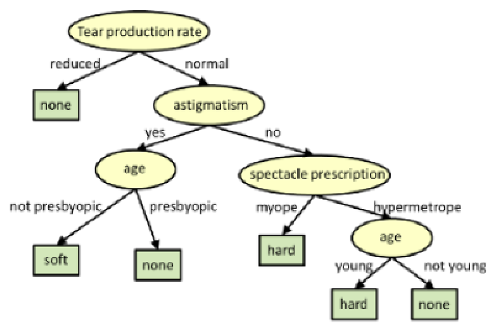
Here are two ways to remove overfitting:

1. Pruning Decision Trees.
2. Random Forest

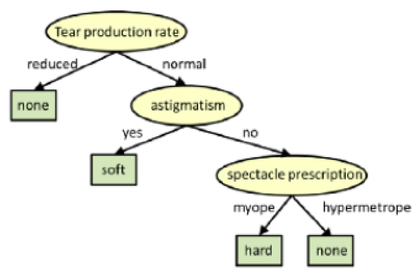
### 1. Pruning :

-- In pruning, you trim off the branches of the tree, i.e., remove the decision nodes starting from the leaf node such that the overall accuracy is not disturbed.

--- This is done by segregating the actual training set into two sets: training data set, D and validation data set
--- Continue trimming the tree accordingly to optimize the accuracy of the validation data set




Original Tree




Pruned Tree