Decision Tree

- Decision tree can be used for both classification and regression problem.

Types of Decision Tree

- D.T.C (Decision Tree Classifier)
- D.T.R (Decision Tree Regression)

D.T.C ⟶ feature [cat/Num] ⟶ o/P [cat]

D.T.R ⟶ feature [cat/Num] ⟶ o/P [Num]

Cat : Categorical Num : Numerical o/P : Output.

ID3 : Interactive Decotomiser → Entropy.

CART : Classification And Regression Tree
↳ Gini Impurity.

Entropy: Measure of randomness or disorder of a system.

$$\boxed{= - \sum_{i=1}^{n} P_i \times \log_2 (P_i)}$$ [P_i : Probability.]

Ex:

For two class with yes/No (Y/N)

Entropy $\Rightarrow$ $-P_Y \times \log_2(P_Y) - P_N \times \log_2(P_N)$

Gini Coefficient: Measure the inequality among values of a frequency distribution.

Ex: $\boxed{1 - \sum_{i=1}^{n} P_i^2}$ [P_i : Probability]

level of income inequality.

For two class Yes/No (Y/N)

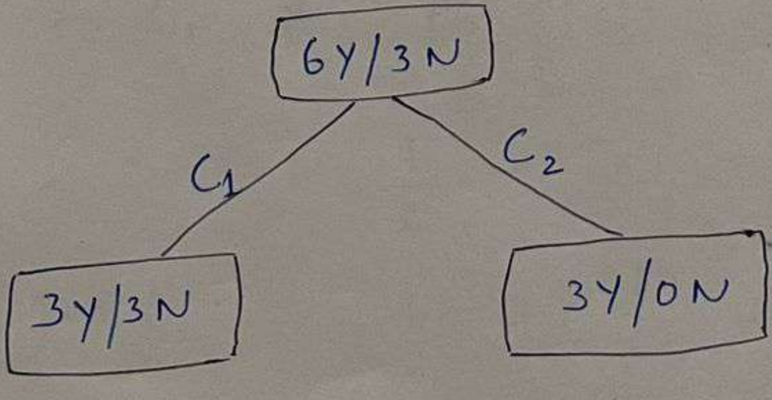
Gini coefficient : $1 - [P_Y^2 + P_N^2]$

Note: we find Entropy / Gini coefficient to find Purity of feature / randomness of feature.

Consider data for example:

f_i	O/P
C_1	Y
C_1	N
C_2	N
C_1	Y
C_2	Y
C_2	Y
C_1	N
C_1	Y
C_2	Y

we can draw following tree:



Calculating Entropy: $-\sum_{i=1}^{N} P_i \times \log_2(P_i)$

C_1

$$= -P_Y \times \log_2(P_Y) - P_N \times \log_2(P_N)$$

$$= -\frac{1}{2} \times \log_2 \frac{1}{2} - P_2 \times \log_2 \frac{1}{2}$$

$$= \frac{1}{2} + \frac{1}{2} = 1$$

Entropy C_2 : $-\sum_{i=1}^{n} P_i \times \log_2(P_i)$

$$= -1 \times \log_2 1 + 0 = 0$$

From above we have following condition:

$P = \frac{1}{2}$, $H = 1$ | Pure split

$P = 1$, $H = 0$ | Impure split

- - - - - - - - - - -

P : Probability | H : Entropy.

Calculating Gini Coefficient:

for $C_1 = 1 - \left[\frac{1}{2}^2 + \frac{1}{2}^2\right] = 1 - \frac{1}{2}$

$$= \frac{1}{2}$$

for $C_2 = 1 - [1^2] = 0$

ID_3 approach $\longrightarrow$ H $\longrightarrow$ I.G

CART approach $\longrightarrow$ Gini-Coef. $\longrightarrow$ I.G

- - - - - - - - - - -

H : Entropy | I.G : Information Gain.

$$\boxed{I.G = H(S) - \frac{\sum |S_v|}{|S|} \times H(S_v)}$$

H(S) : Root feature Entropy

S_v : Sample value

S : Tatal feature sample value

$H(S_v)$: Entropy of sample value.

Considering earlier example we Have:

6Y / 3N for root feature.

Root feature Entropy:

$$= -P_Y \times \log_2(P_Y) - P_N \times (\log_2(P_N))$$

$$= \frac{-6}{9} \times \log_2\left(\frac{6}{9}\right) - \frac{3}{9} \times \log_2 \frac{3}{9}$$

$$= -.66 \times -.585 - .33 \times -1.585$$

$$= .3861 + .523 = 0.909$$

we have entropy of samples as

$H(S_v) = 1$ for 3Y/3N & 0 for 3Y/0N

Calculating I.G:

$$.909 - \left[\frac{6}{9} \times 1 + \frac{3}{9} \times 0\right]$$

$$= .909 - .666 = .243$$

Consider numeric value for regression.

For regression problem we need to first arrange dataset in order and then find average value of adjacent value.

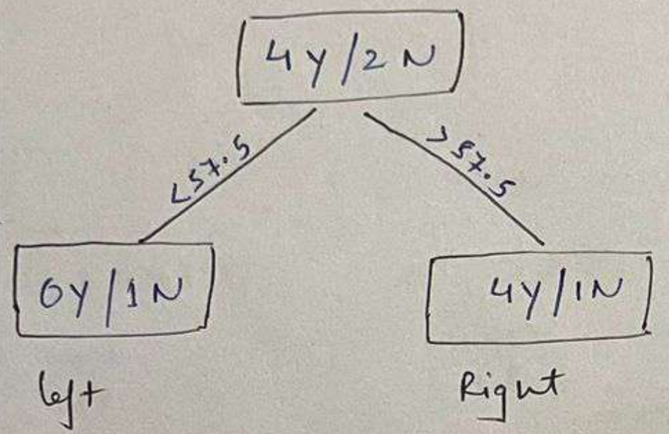
weight (kg)	Heart disease
65	Y
70	Y
55	N
60	N
80	Y
100	Y

Considering average value of adjacent weight and building decision tree for Greater and lower than the value.

Arranged

	weight (kg)	Heart disease
	55	N
57.5		
	60	N
62.5		
	65	Y
67.5		
	70	Y
75		
	80	Y
90		
	100	Y

1st Tree :



Calculating Gini-coefficient for tree:

left : 0

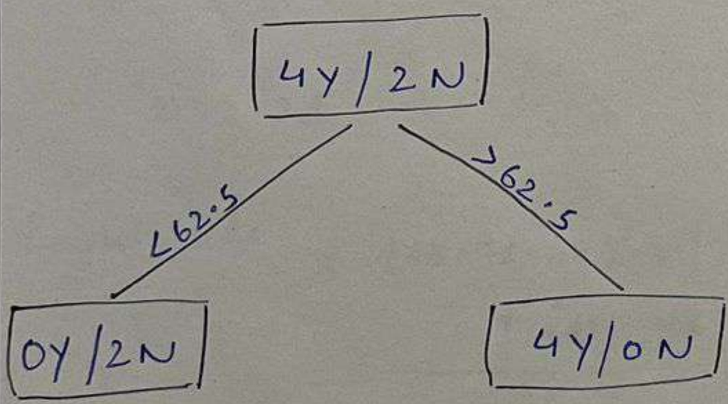
Right :

$$1-\left[\left(\frac{4}{5}\right)^2+\left(\frac{1}{5}\right)^2\right]$$

$$I.G = G.I[Root] - \sum \frac{|S_v|}{|S|} \times G.I(child) = 1 - .68 = .32$$

$$= .45\left[\frac{1}{6} \times 0 + \frac{5}{6} \times .32\right] = .42 - .26 = .19$$

2nd Tree :

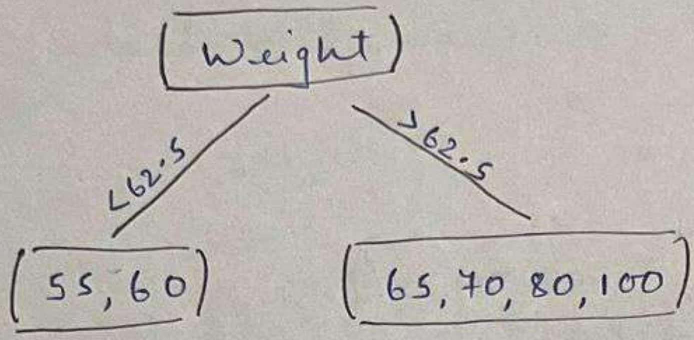


Calculating Gini-coefficient for tree:

Both nodes are Pure node so value is max = 1

So, I.G is Max = root node = .45

→ Considering above tree for M.S.E (variance) and Reduction in variance calculation:



$\mu = 57.5$ mean left

$\mu = 78.75$ mean Right

Mean of weight

$$= \frac{55 + 60 + 65 + 70 + 80 + 100}{6}$$

$$= 71.5$$

$$\boxed{\text{Variance} = \frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2}$$

$$= \frac{(71.5-55)^2 + (71.5-60)^2 + (71.5-65)^2 + (71.5-70)^2 + (71.5-80)^2 + (71.5-100)^2}{6}$$

$$= \frac{(16.5)^2 + (11.5)^2 + (6.5)^2 + (1.5)^2 + (-8.5)^2 + (-28.5)^2}{6}$$

$$= 222.29$$

Right side variance $= \frac{(78.75-65)^2 + (78.75-70)^2 + (78.75-80)^2 + (100-78.75)^2}{4}$

$$= \frac{(13.75)^2 + (8.75)^2 + (1.25)^2 + (21.25)^2}{4}$$

$$= 179.5$$

Left side Variance $= \frac{(57.5-55)^2 + (60-57.5)^2}{2}$

$$= 6.25$$

Reduction in variance

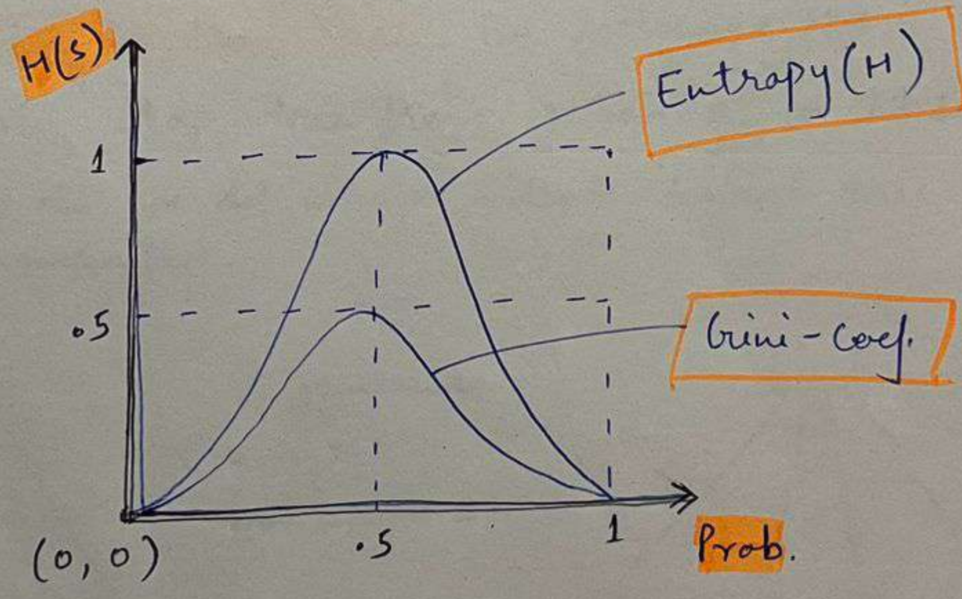
$$= Var(root) - \sum_{i=1}^{n} w_i \times Var(child)$$

$$= 222.3 - \left[\frac{2}{6} \times 6.25 + \frac{4}{6} \times 179.5 \right]$$

$$= 222.3 - [2.08 + 119.55]$$

$$= 222.3 - 121.63 = 100.67$$

Graph of Entropy & Gini-Coefficient



Entropy w.r.t Probability.

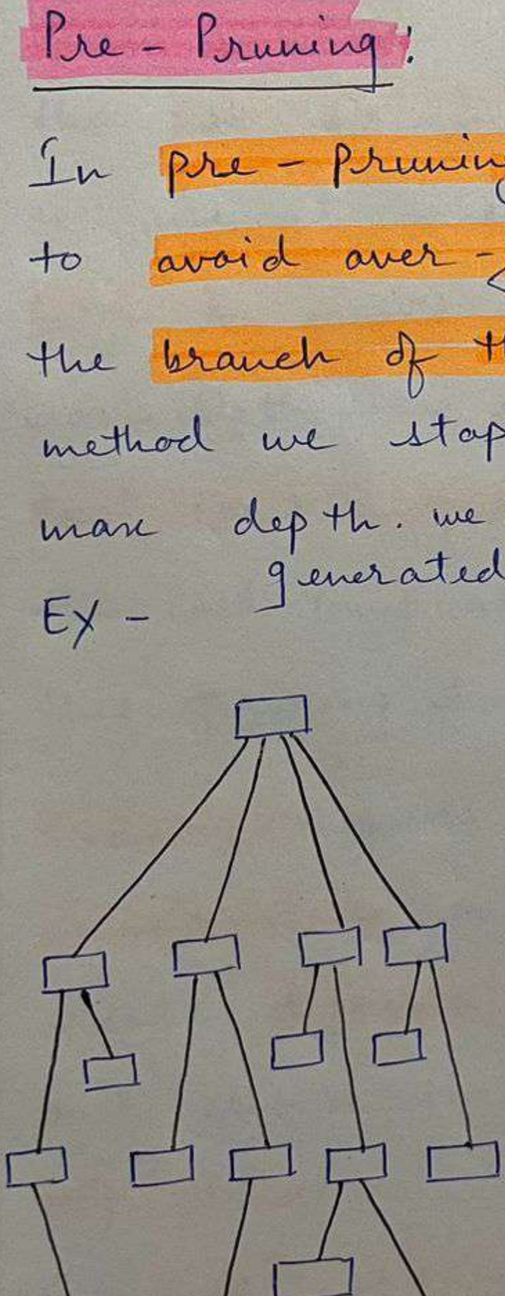
Pre-Pruning and Post-Pruning.

when a dataset is huge and we allow it to grow to it's max branches or depth it always led to overfitting (well perform with training data but poor with test data).

Pre-Pruning:

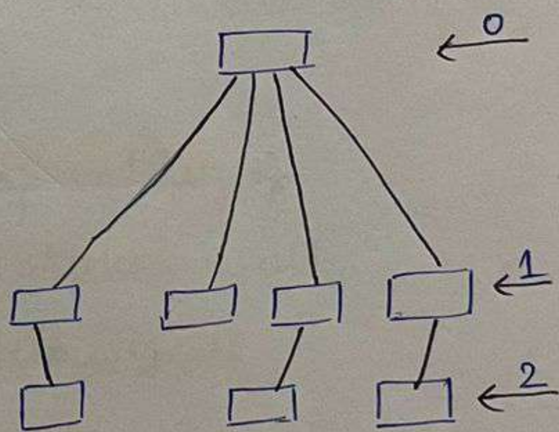
In pre-pruning we tweak same parametess to avoid over-fitting by defining depth of the branch of the tree. By fallowing the method we stop tree fram growing to it's max depth. we define max-depth before tree is generated.

Ex -



Decision tree with max depth.

Suppose we keep:
max_depth=2



Decision tree with:
max_depth = 2

Prepruning stops the non-significant branches from generating and we apply the technique before the construction of a decision tree.

Post-Pruning:

Post pruning is called as backward pruning. Decision tree is generated first then the non significant branches are removed. we use it when decision tree has very large or infinite depth. Instead of using max-depth like in pre-pruning we use cost complexity pruning techniques. ccp_alpha, the cost complexity parameter, parameterizes this pruning technique.

ccp_alpha gives minimum leaf value of decision tree and each ccp_alpha will create different-different classifier and choose the best out of it. More the ccp-alpha value will be more the no.s of nodes are pruned.