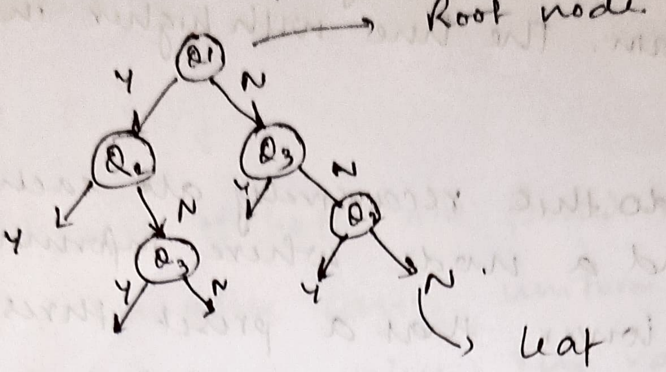
Decision Trees

→ Decision tree classify / regress (?) based on nested questions



→ Q1, Q2, Q3 can be from any of the provided features.

→ How to choose best feature / question.

INFORMATION GAIN (Classification)

$$\text{Information Gain}_{(S)} = EG_{(S)} - \overbrace{\left[(\text{Weighted Avg}) \times EG_{(\text{each feature})}\right]}^{\text{Information}}$$

(S) → for whole collection. (At root node it is out entire dataset)

(EG) → Entropy or Gini Index

Entropy

$$\text{Entropy} = -\sum_{i=1}^{n} p_i * \log_n (p_i) \qquad (E)$$

So for yes or no,

$$\underline{\underline{E}} = -P(yes) \log_2 (P(yes)) - P(no) \log_2 (P(no))$$

Gini Index

$$\text{Gini} = 1 - \sum_{i=1}^{n} p_i^2$$

so here

$$G = 1 - (P_{yes}^2 + P_{no}^2)$$

Information.

→ Finding weighted averages of entropies for all possible features / questions and subtract them from root / previous entropy. This gives us the information gain. The One with higher information gain stands.

→ We have to do this recursively at each stage until we find a node where information gain below it is lower that a preset threshold value. We make that node a leaf.

→ At the end, for prediction of class just flow down the data throught the tree to find its class / till it hits a ~~f~~ leaf. Majority of classes present there is assigned to it.

→ In case of regression the following thus is done.

Variance Reduction (For Regression)

Variance $= \frac{1}{n}\sum(y_i - \bar{y})^2$, $\bar{y}$ = mean

Variance ↓ Impurity ↓

1. Find variance for ~~Root~~ whole dataset at root
2. Find variance for all possible divided datasets individually.

$$\boxed{\text{Var Reduction} = \text{Var (root)} - \sum w_i \,\text{Var (child)}}$$

Higher Variance Reduction is chosen.

→ In regression at the end class take average of all existing values or weighted averages.

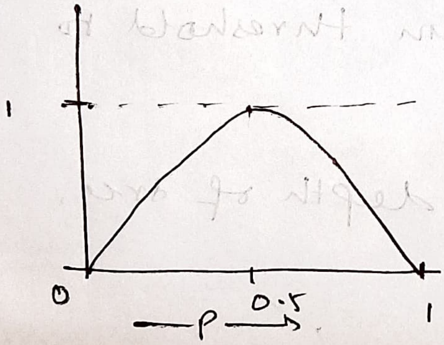
Possible Questions

1. Gini Vs Entropy.

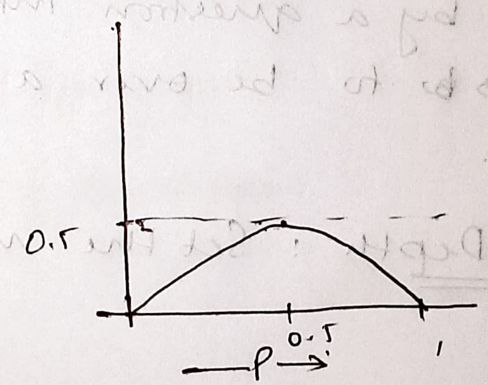
A. Highest values of both are found at 1/2 (for yes no). For other cases, uniform probabilities make the highest value. The values are

Entropy - 1
Gini - 1/2

Graphs:



Entropy



Gini

Entropy takes 1 as max rather than 0.5. Gini is used more in State-of-the-art algos because, it is more computationally efficient. This is because Entropy uses log and logarithmic computation take more time. And for same conditions, Gini only maxed at 0.5 while entropy at 1. Increases till 0.5 & decreases till 0.5 to 1.

→ Decision trees are normally robust to outliers, ~~So~~ but they are prone to overfitting. To overcome this; pruning is done.

PRUNING

- Pre-pruning
- Post-pruning

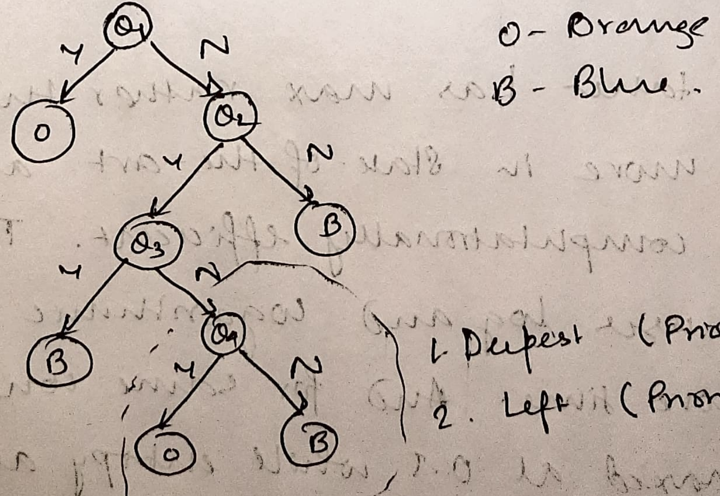
Pre-pruning

1. Min-samples: Save a threshold that the minimum number of data does not exceed the threshold. i.e. the number of data points classified by a question into different sub-nodes/leaf has ~~to~~ to be over a certain threshold to be valid.

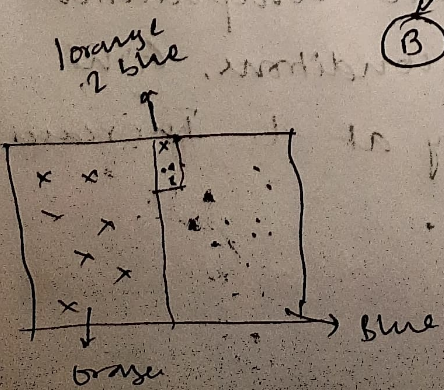
2. Max-Depth: Set the max depth of tree.

~~Beo~~

Post-Pruning



1. Deepest (Priority-1)
2. Left (Priority-2)



→ We start with the deepest node and left / yes entry.

→ In genaral, choose the max number of classes present in the decision node area and replace it by that class. Thus blue replaces Q4 here. The highlighted area has 3 blue and 1 orange.

→ Check errors in prediction for both case. Node & Leaf wrt to validation set. Less error stands,

→ Recursively do this while going up.

→ Thus eliminates overfitting questions.

Normally.

→ Post pruning > Pre pruning.

→ Post pruning is more computationally expensive. Hence slower.

→ Normally both are done.