POORNIMA

(Analytics)

* **Decision Tree:** The decision tree Algorithm belongs to the family of supervised machine learning algorithms. It can be used for both a classification problem as well as for regression problem.

The goal of this algorithm is to create a model that predicts the value of a target variable, for which the decision tree uses the tree representation to solve the problem in which the leaf node corresponds to a class label and attributes are represented on the internal node of the tree.

→ Assumptions that we make while using the Decision Tree:

↳ In the beginning, we consider the whole training set as the root.

↳ Feature values are preferred to be categorical if the values continue then they are converted to discrete before building the model.

↳ Based on attribute values records are distributed recursively.

↳ We use a statistical method for ordering attributes as a root node or the internal node.

... boundaries.

* Entropy values range from 0 to 1", Less the value of entropy more it is trusting able.

Entropy:

$H(S) = -\text{probability of } \log_2(p+) - -\text{probability of } \log_2(p-)$

Where: $(p+) \rightarrow$ % of positive class

$(p-) \rightarrow$ % of negative class

If we perform Entropy and get equal value like 50% yes or 50% no. Then the splitting will be going on unless and until we get a pure subset.

POORNIMA

Pure subset: The pure subset is a situation where we will get either all yes or all no in the class.

In this case, we take other attributes to reach the leaf node and also have to take the entropy of those values and add it up to do the submission of all those entropy values for that we have the concept of info^n gain.

Information Gain: Info^n gain is used to decide which feature to split on at each step in building the tree. At each step, for small tree, we should choose the split that results in the purest daughter nodes. A commonly used measure of purity is called info^n.

Information Gain:

$$\text{Gain}(S, A) = H(S) \sum \frac{|S_v|}{|S|} H(S_v)$$

The algo. calculates the info^n gain for each split and split which is giving the highest value of info^n gain is selected.

$S_v \rightarrow$ Total Sample after split

$S \rightarrow$ Total Sample

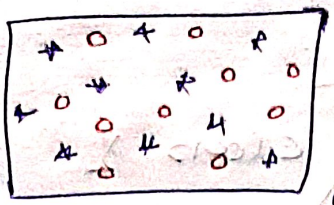
POORNIMA

The higher the value of info^{n} gain of the split the higher the chance of it getting selected for the particulare split.

Gini Impurity:

Gini Impurity is a measurement used to build Decision trees to determine how the features of a data set should split nodes to form the tree. Gini impurity of a data set is a number between 0-0.5.

→ What is Decision Tree?



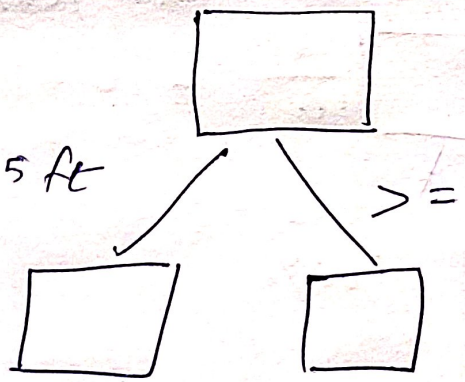
Total no of student = 20
(o) Play cricket = 10 (50%)
(*) Do not play cricket = 10

Features of students :-

- Height
- Performance in class
- class

Using these features we want to train a model and predict wheather they will play cricket or not.

→ Split on Height



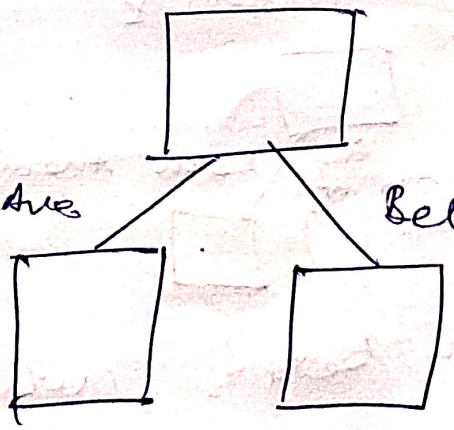
< 5.5 ft

>= 5.5 ft

Students = 8
Play cri = 2
Percentage = 25%

Students = 12
Play cricket = 8
Percentage = 66.67%

→ Split on Performance in class



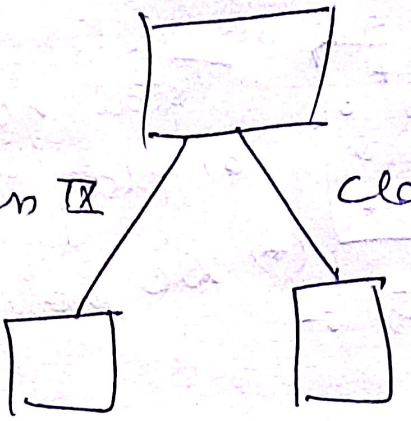
Above Ave

Below Ave.

Students = 14
Play cricket = 8
Percentage = 57.14%

Students = 6
Play cricket = 2
Percentage = 33.33%

Split on class



Students = 10
Play cricket = 8
Percentage = 80%

Students = 10
Play cricket = 2
Percentage = 20%

→ In decision tree we have to separate the classes.

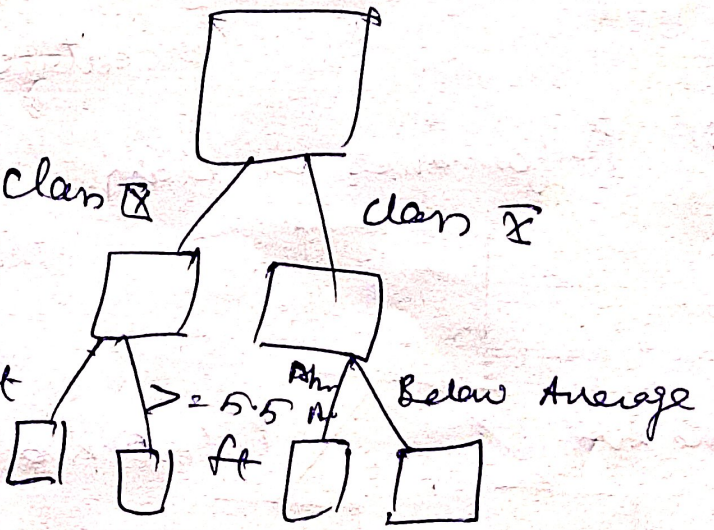
↳ Pure nodes

↳ by which also we split the class.

Split on Played Cricket last year?

100%

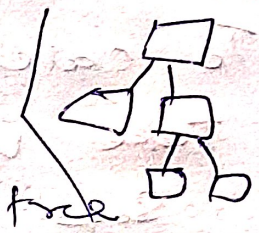
→ What is



Which is split is better, what should be sequence,

Terminologies:

→ Root node
→ Splitting
→ Decision Nodes
→ Left / Terminal Node
→ Branch / Sub Tree
→ Parent and child node
→ Depth of tree (longest path of tree



Select the best split point in Decision Tree:-

- Decision tree split the nodes on all available variables
- Select the split which result in most homogeneous sub-nodes

1. Gini Impurity = 1 − Gini

→ Node split is decided based on the gini impurity

Gini tells purity

Gini impurity tells the impurity of nodes

Selection of two random variable from the node. If node is pure then probability = 1

- Low the gini impurity, higher the homogeneity of nodes.

⑤ Gini works on Categorical target not works on continuous target eg: only perform binary split not in house prediction, etc.

Gini = Sum of Square of probabilities for each class / category

$$Gini = (P_1^2 + P_2^2 + P_3^2 - \cdots + P_n^2)$$

- To calculate the gini impurity for split, take weighted gini impurity of both sub-nodes of that split.

→ Split on Performance in class.

Prob. play = $\frac{8}{14} = 0.57$

not play = $\frac{6}{14} = 0.43$

} Above Average

Play $\frac{2}{6} = 0.33$

not play $\frac{4}{6} = 0.67$

} Below Average

Student play Cricket

Student = 6
Play cricket = 2
Don't play = 4

Gini impurity: Sub node Above Average:

$1 - [(0.57) * (0.57) + (0.43) * (0.43)] = 0.49$

Gini impurity Sub node Below Average:

$1 - [(0.33) * (0.33) + (0.67) * (0.67)] = 0.44$

Weight of node = No node in child / No node in Parent

for both class Above Average ~~word~~ ⇒ 14/20

Below Average = 6/20

Weighted Gini Impurity: Performance in class

(14/20) * 0.49 + (6/20) * 0.44 = 0.475

⟶ Split on class : 0.32

Split Performance in class	weighted Gini Impurity 0.475
Class	0.32