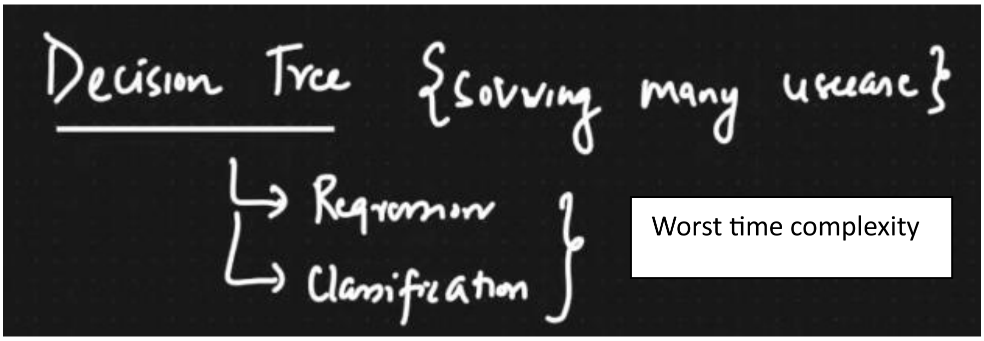
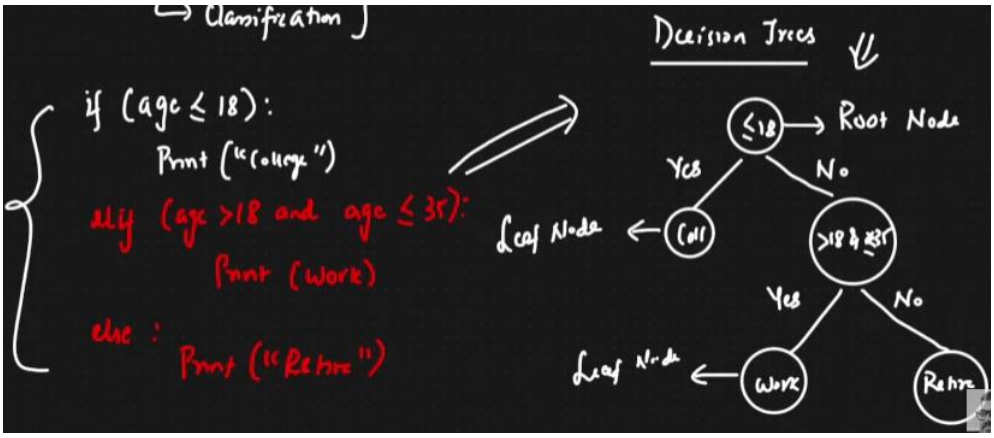
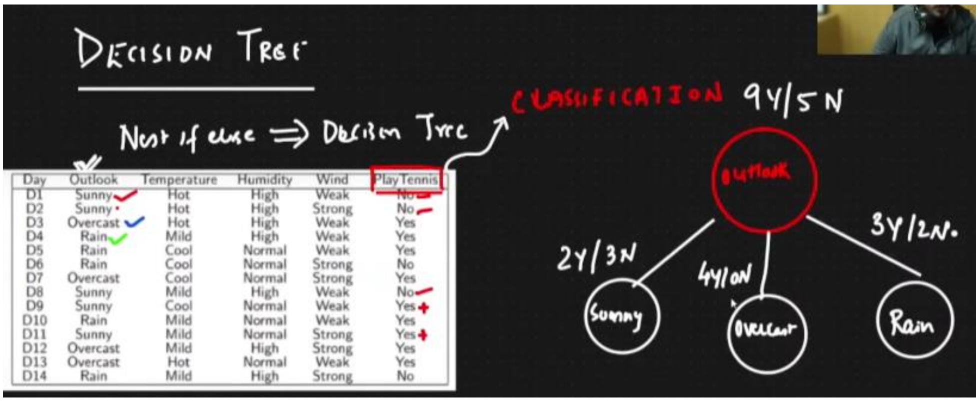


i) Decision Tree is like a flowchart of the python, if-else code!

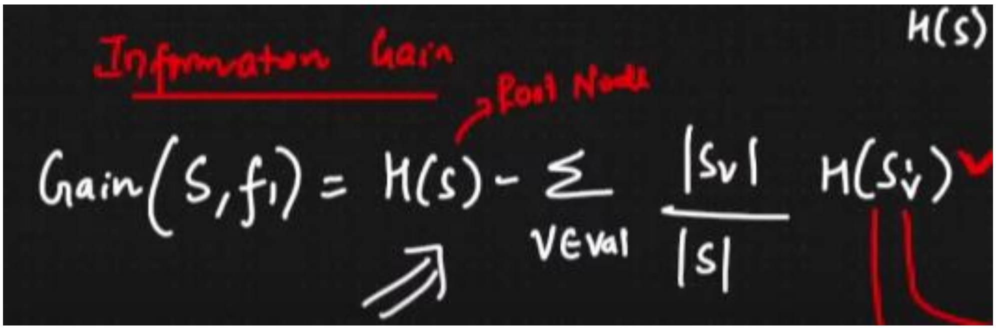


ii) Classification: (here, outlook is randomly selected!)



Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

a) How are the features selected? → through: **Information Gain**



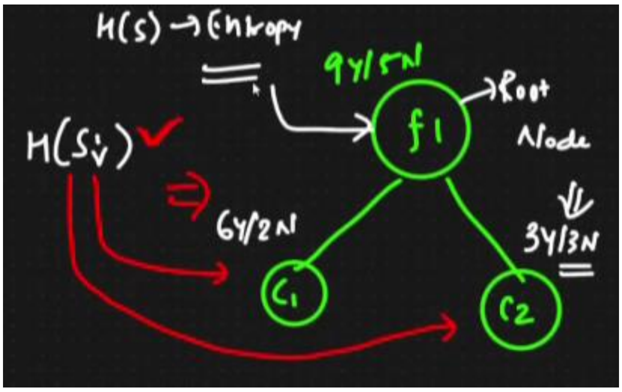
Were,

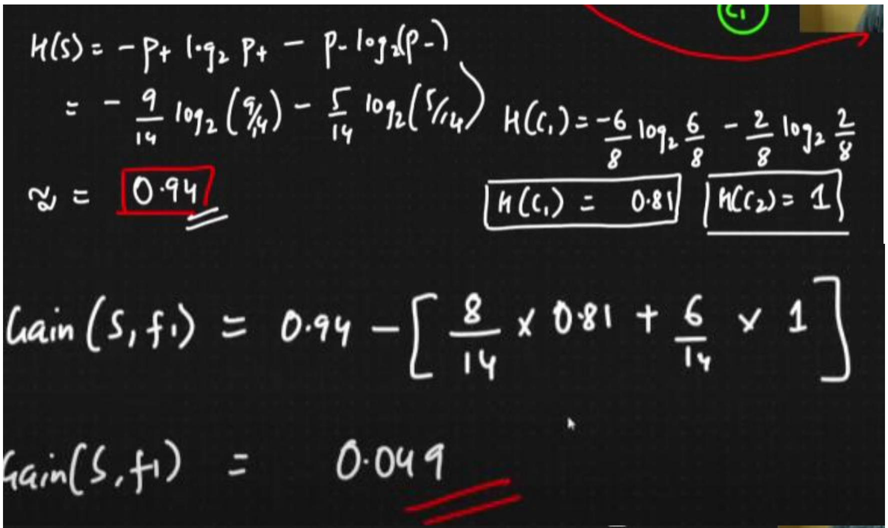
H(S): Entropy of root node(f1)**,**

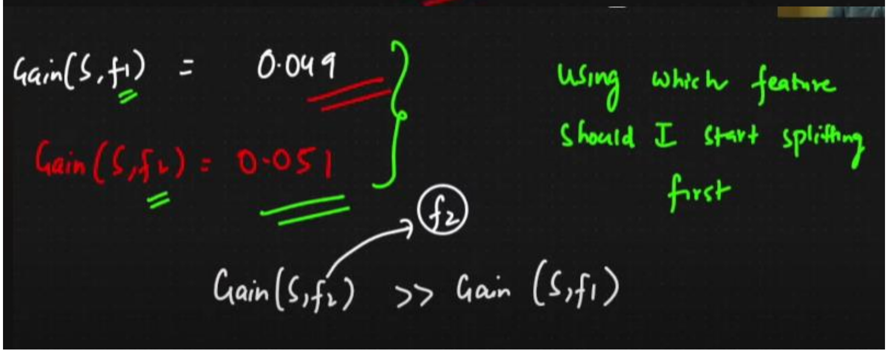
H(S'V): Entropy of c1 & c2**,**

$\frac{SV}{S}$: $[\frac{\text{total c1}}{\text{total f1}} + \frac{total\ c2}{total\ f1}]$

Example:







1) we took root node as f1, found the H(S) value.

2) went inside the f1's child root & found c1 & c2.

3) we got all the necessary data, now we place the value in place of the formula.

4) we got: "Gain (S, f1)"

5) similarly, find for other nodes (f2, f3, f4, etc.)

6) the node, having highest Gini value, is taken 1st!

b) Later, **Split** in **2 categories:**

i) Pure (100% yes or 100% no): Overcast is the pure node, having 100% strike rate, 4 yes & 0 no!

ii) Impure: not a 100% strike rate!

Once you get a pure node, stop else keep going on with the next features(columns)

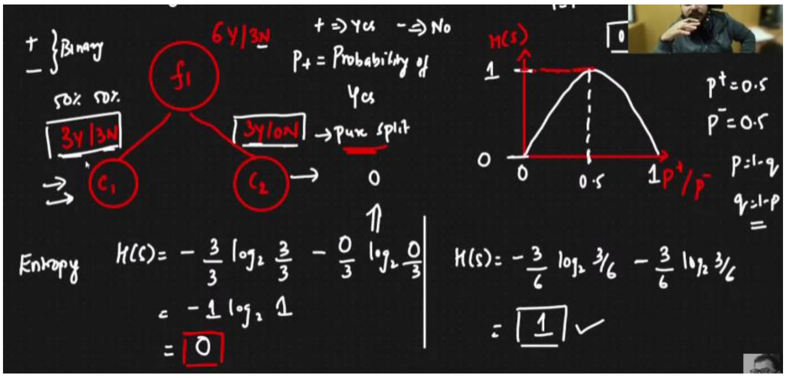
c) How to know wheatear it's a **pure fit or impure fit? → Entropy or Gini Impurity**

i) Entropy:

$$H(S) = -P_+ \log_2 P_+ - P_- \log_2 P_-$$

-: no, +: yes, p+: prob. of yes, p-: prob. of no

Example:

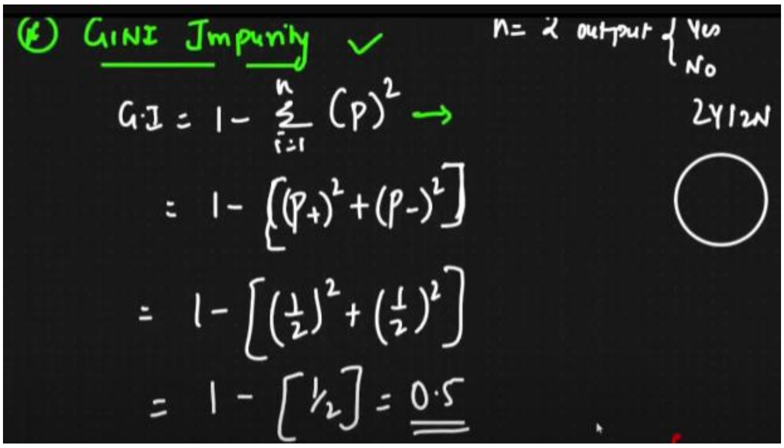


Entropy is always going to be between 0 to 1

0 = pure & 1 = impure

As you can see, at the right side we got H(S) = 1; split it further, taking other features/columns.

ii) Gini Impurity/Coefficient:



1) took random no. of output, that's 2 each (yes & no)

2) placed the value inside the formula.

3) got the answer, 0.5 an impurity!

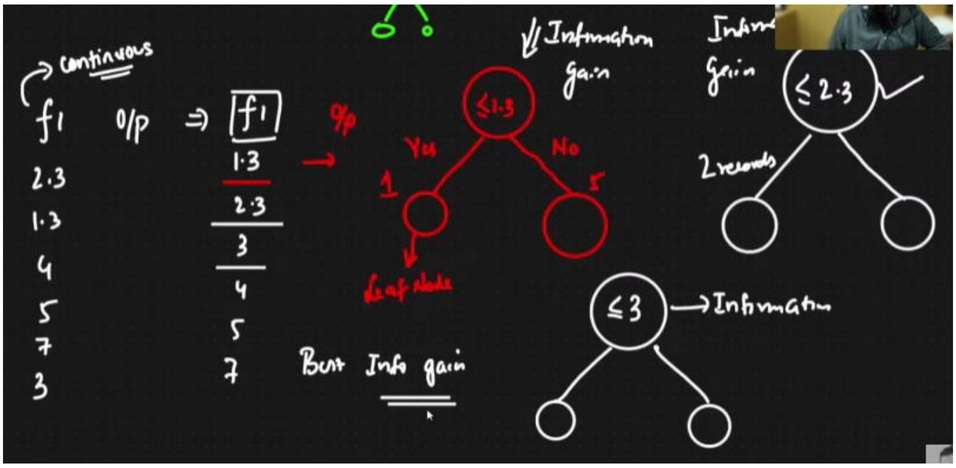
4) entropy has only 0 or 1 but Gini doesn't!

iii) Question may arise, which should be taken & when?

- ➔ When there's more than 100 nodes, take Gini, due to its simple maths & faster solution;
- ➔ As, entropy contains log & does takes lots of time!

iii) Regression:

If we have 1 feature:

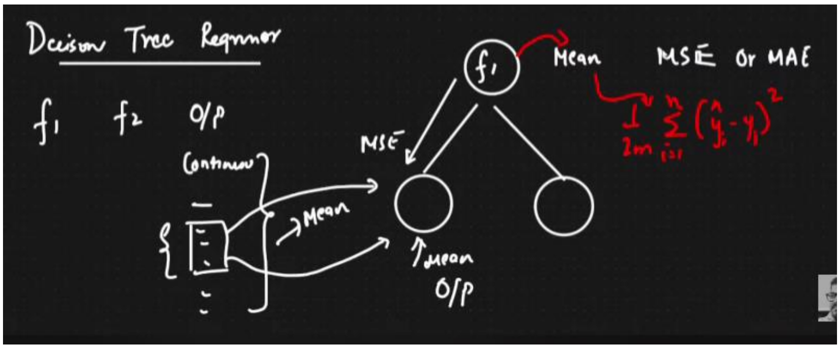


a) Here, will first arrange the f1 in ascending order

b) Later, take the first value, that's 1.3 & find their information gain.

c) Simultaneously, get information gain for all the remaining values & the one with highest will be taken to perform purity – impurity (entropy or Gini impurity)

When we have multiple features:



1) take all the mean of the output, will get assign to f1 & will use MSE or MAE, instead of Entropy or Gini impurity

3) based on the f1 feature, assign the mean value & later find the MSE, MAE; in the end, split!

4) during the split, some records will go under the f1, becoming c1, later find the mean of those value & find the mse or mae!

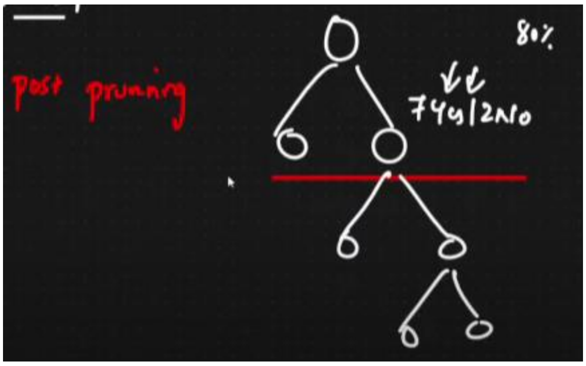
5) as the mse gets reduced, that means we are reaching to leaf score!

6) and the same thing will happen for c2.

7) the mean value present at c1, c2, etc. will be the output!

iv) Hyperparameters:

Has **overfitting**, to fix this, will do: **post-pruning** & **pre-pruning**



If we know that, there's an 80% of a node to be yes; will cut the further part of the tree.

In the pre-pruning, will know the max_depth, max_leaf, we can get applying gridsearchcv.

Reference:

[1) Decision Tree & Ensemble ML Algorithm]