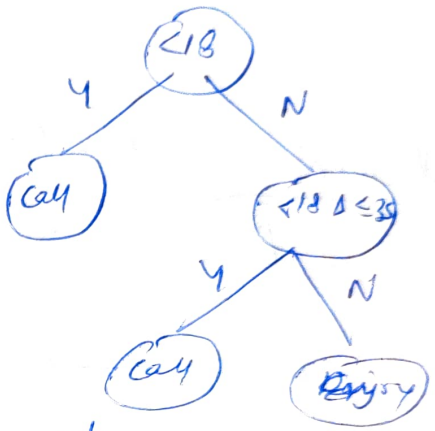
Decision Tree

- Regression
- Classification

Decision tree

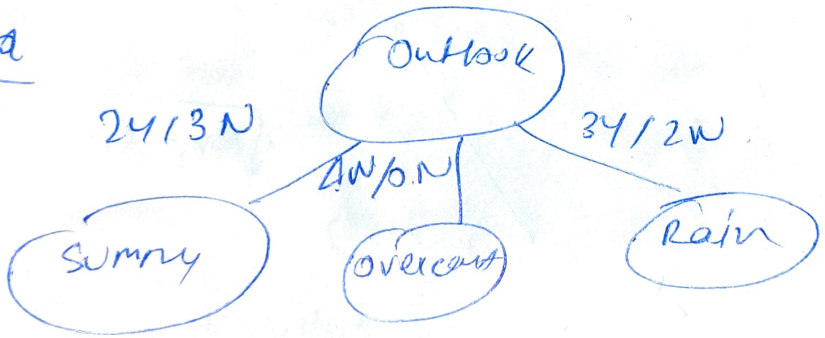
Example :-

```
If (age<=18):
    print ("College")
elif (age<35):
    print ("Work")
else:
    print ("Enjoy")
```



So, Using nesting we solve regression & classification problem.

Over data



How we determine Y/N/ow

↳ ① Purity → Pure split {
↳ Entropy
↳ Gini coefficient }

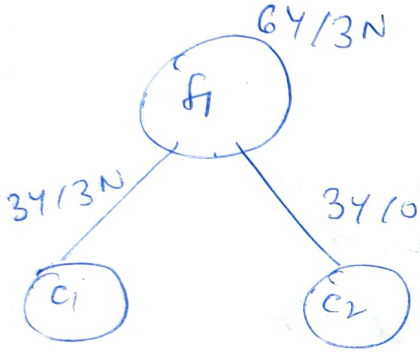
② How the features are selected
↳ Information gain? }

① Entropy → For purity check

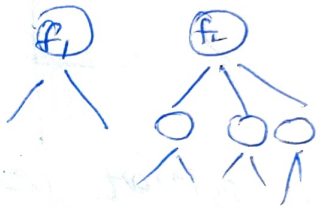
$H(S) = -P_+ \log_2 P_+ - P_- \log_2 P_-$

② Gini Impurity

$Gini = 1 - \sum_{i=1}^{n} (P_i)^2$



P_+ = Prob. of Yes.
P_- = Prob of No.



→ Which we select

That depend on information gain → Root node

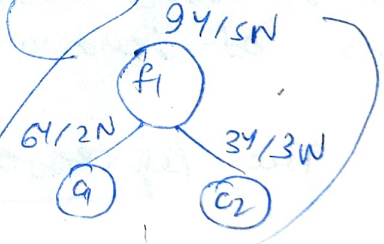
$H(S) = -\frac{3}{3}\log_2\frac{3}{3} - \frac{0}{3}\log_2\frac{0}{3}$

$= -1\log_2 1$

$= 0$ → Whenever we are having pure split.

$Gain(S, f_1) = H(S) - \sum_{i\,level} \frac{|S_v|}{|S|} H(S_i)$

↳ Category node



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

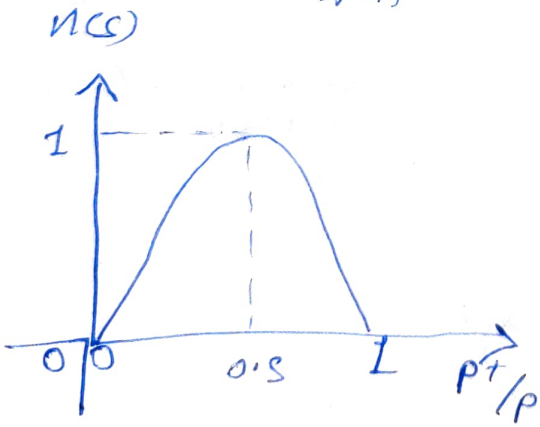
$= -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right)$

$= 0.94$

So,

$H(C_2) = -\frac{6}{8}\log_2\frac{6}{8} - \frac{2}{8}\log_2\frac{2}{8}$

$H_{C_1} = 0.81$ & $H_{C_2} = 1$



Purity Test → Entropy rule

Entropy = 0-1

So, $Gain(S, f_1) = 0.94 - \left[\frac{8}{14} \times 0.81 + \frac{6}{14} \times \right.$

$= 0.049$ For feature f_1

$Gain(S, f_2) = 0.051$ → F_2

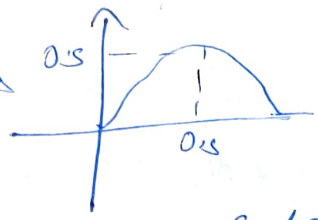
So, Gain(S, f_2) >> Gain(S, f_1)

↳ we will go with f_2

Gini Impurity

$$G.I = 1 - \sum_{i=1}^{n} (p_i)^2 = 1 - \left[(P_+)^2 + (P_-)^2\right]$$

$$= 1 - \left[\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right] = 1 - \left[\frac{1}{2}\right] = 0.5$$



(*)

We can either use Entropy or Gini.

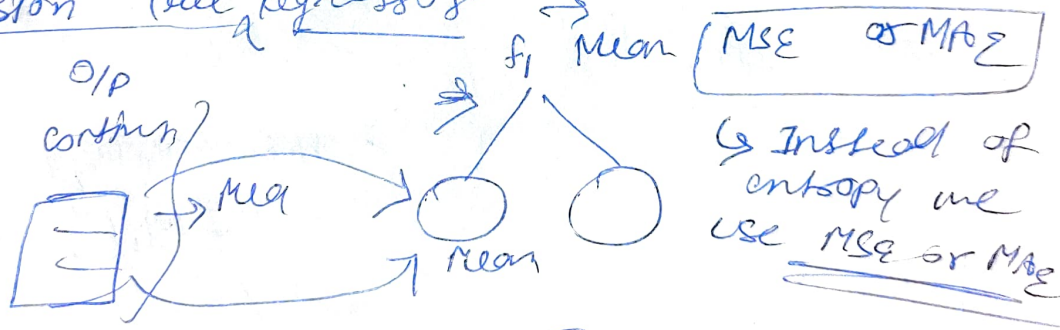
↳ But Entropy → log → more time

Gini → simple math → Less time.

In lots of features use Gini

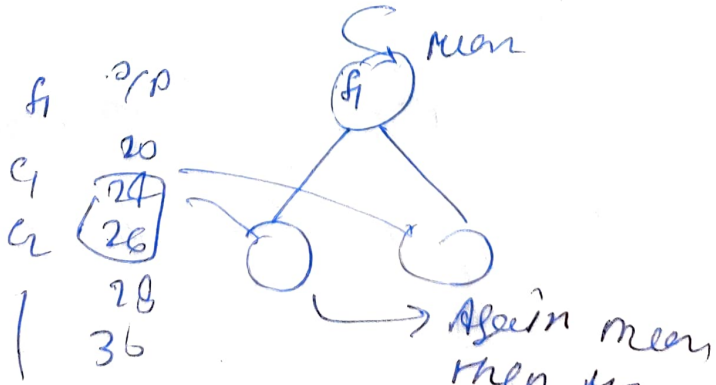
In small no. of features use Entropy.

Decision Tree Regressor

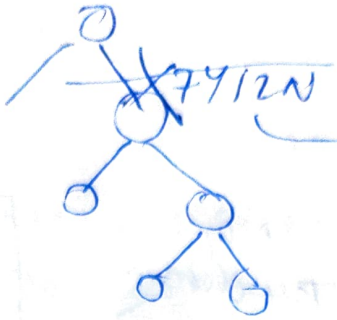


↳ Instead of entropy we use MSE or MAE

Hyperparameters



⇒ Decision-tree lead to overfitting.



Pre-pruning → Post-pruning.

↳ Max depth
Max leaf.
↳ Grid Search CV.

Code