Lecture - 04 :

Machine learning algorithms are,

→ Decision Tree Classification.

→ Decision Tree Regression.

→ Practical Implementation.

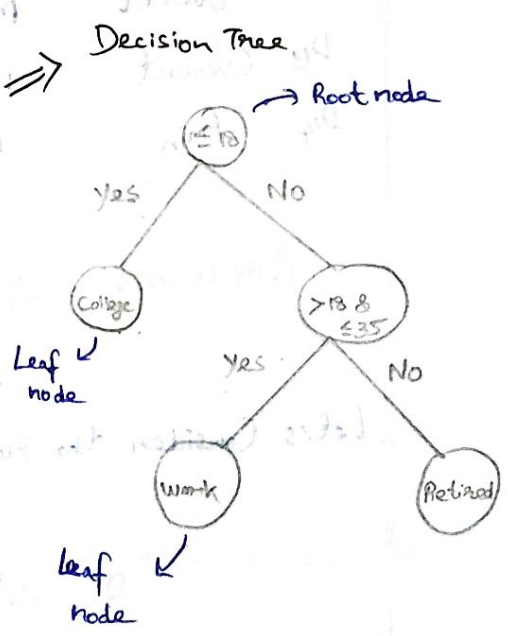
→ Ensemble Techniques.

Decision Tree : [Solving Many Use Cases]

Decision Tree
- ↳ Regression.
- ↳ Classification.

Consider, [If - condition]

```
if (age ≤ 18) :
    print ("college")
elif (age > 18 and age ≤ 35) :
    print ("work")
else :
    print ("Retired")
```



Same way, The regression and classification problem is Solved by Decision Tree.

Nested if else ⇒ Actually solved with the help of Decision Tree (visualized manner).

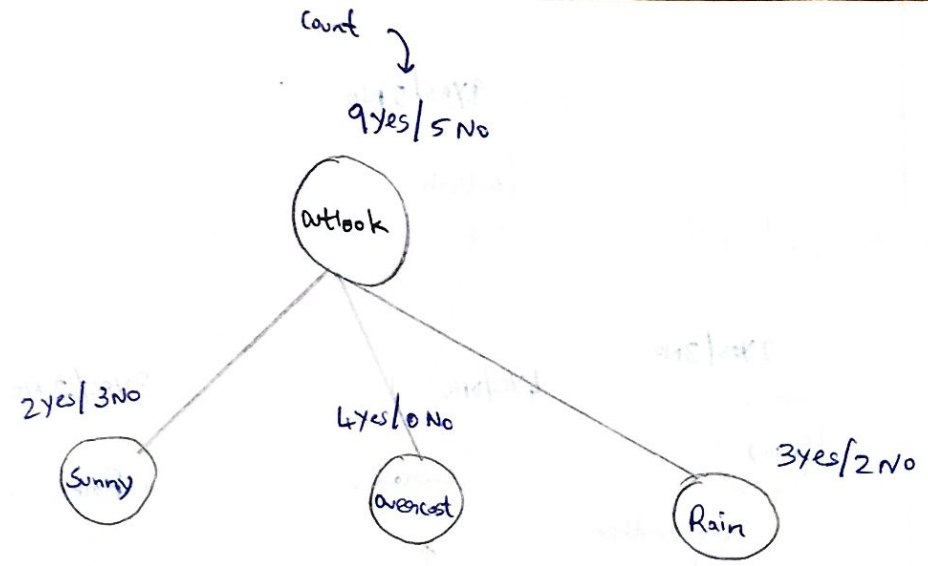
Let's consider the dataset,

DAY	OUTLOOK	TEMPERATURE	HUMIDITY	WIND	PLAY TENNIS
D_1	Sunny	Hot	High	Weak	No
D_2	Sunny	Hot	High	Strong	No
D_3	Overcast	Hot	High	Weak	Yes
D_4	Rain	Mild	High	Weak	Yes
D_5	Rain	Cool	Normal	Weak	Yes
D_6	Rain	Cool	Normal	Strong	No
D_7	Overcast	Cool	Normal	Strong	Yes
D_8	Sunny	Mild	High	Weak	No
D_9	Sunny	Cool	Normal	Weak	Yes
D_{10}	Rain	Mild	Normal	Weak	Yes
D_{11}	Sunny	Mild	Normal	Strong	Yes
D_{12}	Overcast	Mild	High	Strong	Yes
D_{13}	Overcast	Hot	Normal	Weak	Yes
D_{14}	Rain	Mild	High	Strong	No

PLAY TENNIS → O/P [dependent variable]

Let's Consider the random feature,

"OUTLOOK".



Pure split - Pure split ensures all the data in Same Category.

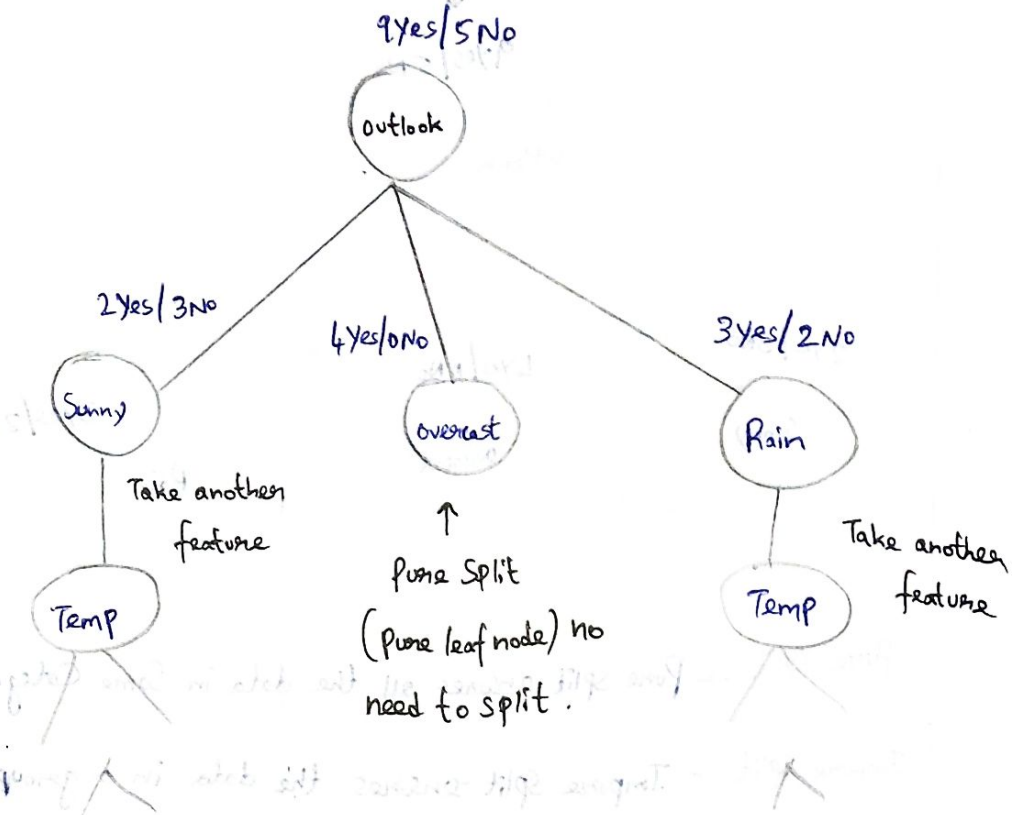
Impure Split - Impure Split ensures the data in a group belongs to different Category.

Let's take the Overcast node,

In this node, 4yes/0No ensures that all are in the Same Category "yes". Suppose the new test data will come, we definitely say Overcast with respect to "yes", you play the Tennis right now.

This type of split is known as the Pure split.

Suppose, The impure split is come, The decision tree take the next feature based on the information Gain.



Split will be happen until the Pure Split is Come. The feature Selection is done by information Gain.

Two Questions,

1. How Purity (or) Pure Split is known by the Machine?

We are just Count and know the Pure Split. But in Machines have Complex Problems to Solve.

Therefore, two techniques used to Say the purity of node are,

→ Entropy.

→ Gini Coefficient (or) Gini impurity,

2. How the features are selected ?

The feature selection with the help of the information Gain.

Entropy:

Formula,

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

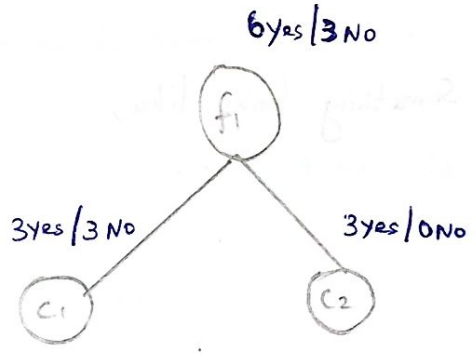
Where,

P_+ => probability of yes.

P_- => Probability of No.

+ } Binary
− } Classification.

Suppose Consider the node,



f → feature

C → Category

Take the node, C_2 to Check the pure split or not

→ probability one check for this node when Calculation.

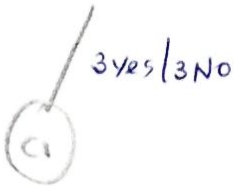
Apply in entrophy formula,

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

$= -1 \log_2 1$

$= -1 (0) = 0 \rightarrow$ Pure Split.

Then Check the another node, C1.

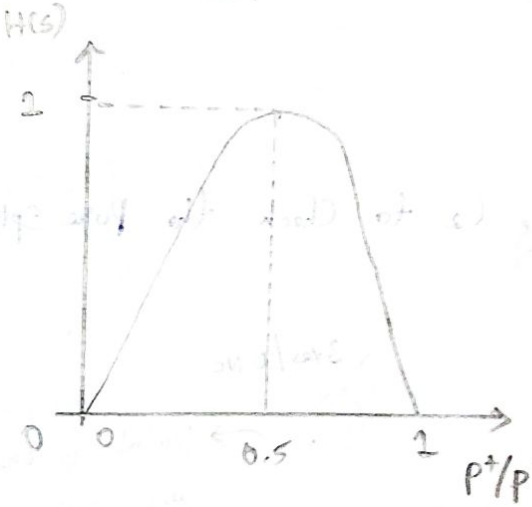


$$H(S) = -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6}$$

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

$= 1 \rightarrow$ Impure Split

The entropy graph Something looks like,



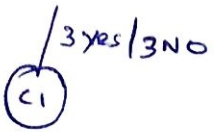
$P^+ = 0.5$

$P^- = 0.5$

$P = 1 - Q$

$Q = 1 - P$

Suppose, The Consider the Second node we are take,



The entropy is 1.

The corresponding the p^+/p^- value is 0.5.

It means 0.5 => 50% of p^+ and 50% of p^-.

The entropy values is always the range of 0 to 1.

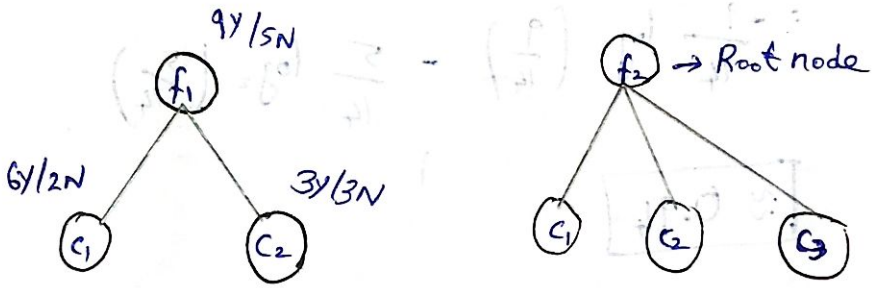
0 → Pure split.

1 → Impure split.

In purity Test, we use the entropy.

Which feature to take to split?

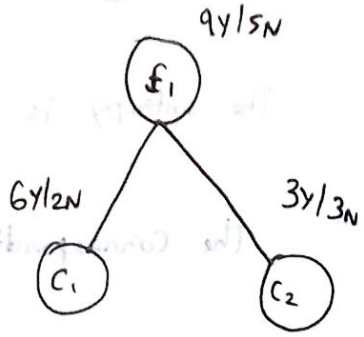
Suppose, we have to take the feature,



Which feature is to take for impure split?

It is responsible for information gain.

Information Gain:



Gain (S, f_1) where $S \rightarrow$ Sample

$f \rightarrow$ feature

$$\text{Gain}(S, f_1) = H(S) - \sum_{v \in \text{val}} \frac{|S_v|}{|S|} H(S_v')$$

- $H(S)$: Entropy for root node
- $\sum_{v \in \text{val}}$: Entropy for how many categories found.
- $S_v \rightarrow$ How many samples each for splitted category.
- $S \rightarrow$ Total Sample Size.

First Find $H(S)$,

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

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

$$\approx 0.94$$

Let's find the $H(S_v')$,

$$H(C_1) = -\frac{6}{8} \log_2 \frac{6}{8} - \frac{2}{8} \log_2 \frac{2}{8}$$

$$H(C_1) = 0.81$$

Same like, Suppose we get

$$H(C_2) = 1$$

9Y|5N
f_1 → Total Sample = 14 (S)

6Y|2N
C_1 → Total sample = 8 (S_{V_1})

3Y|3N
C_2 → Total sample = 6 (S_{V_2})

S_V

$$Gain(S, f_1) = 0.94 - \sum_{i=val}^{2} \frac{|S_v|}{|S|} H(S_v)$$

$$= 0.94 - \left[\frac{8}{14} \times 0.81 + \frac{6}{14} * 1\right]$$

$$= 0.049$$

$$Gain(S, f_1) = 0.049$$

Suppose, we get

$$Gain(S, f_2) = 0.051$$

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

↳ This is more information.

So, take feature (f_2) to split the decision Tree when impure split is happened.

Not only for this two features, Calculate all of the feature and make the decision Tree Split.

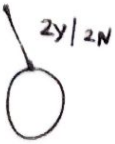
GINI IMPURITY:

n → no of output

Output { Yes / No

$$G.I = 1 - \sum_{i=1}^{n} (P)^2$$

$$= 1 - \left[(P_+)^2 + (P_-)^2 \right]$$



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

$$= 1 - \left[\frac{1}{2} \right]$$

P_+ → probability of yes

$P_+ = \frac{2}{4} = \frac{1}{2}$

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

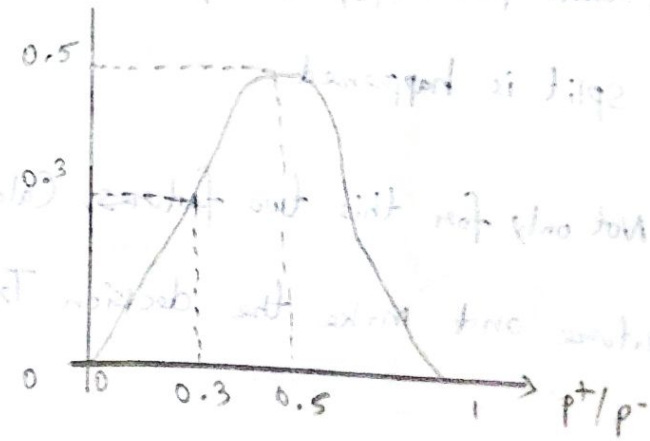
$$G.I = 0.5$$

Gini impurity is faster than entropy. But entropy is also important for calculating information gain.

Large set of datasets => Go Gini impurity.

Minimum or low number of dataset records

└> Go Entropy.



Suppose, In Classification problem we have numerical values in features. How decision tree work?

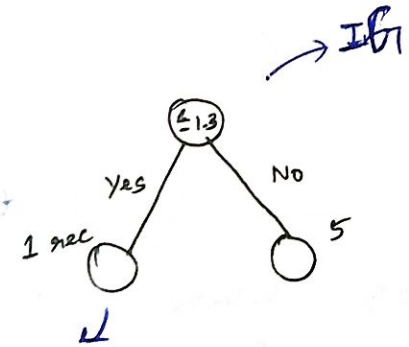
Suppose we have the Continuous feature,

f_1	O/P
2-3	-
1-3	-
4	-
5	-
7	-
3	-

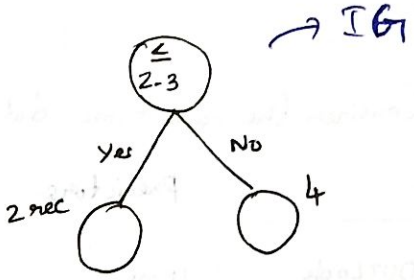
The decision tree Sort the features ⇒

f_1	O/P
1-3	-
2-3	-
3	-
4	-
5	-
7	-

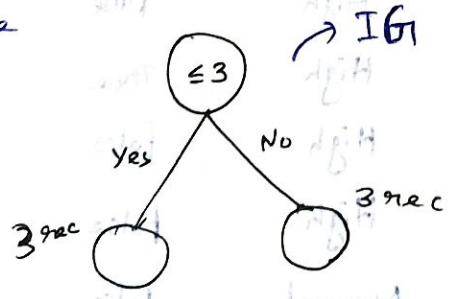
Then take first value,



leaf node



IG1 → Information Gain



Highest IG1 Selected and make the split.

Likwise, Computing the all information gain for every value. Then, Compare and Select for Splitting.

The main thing is that how decision tree work for Regression ?. Because it is Continuous value. Then we move to,

Decision Tree Regressor:

Suppose,

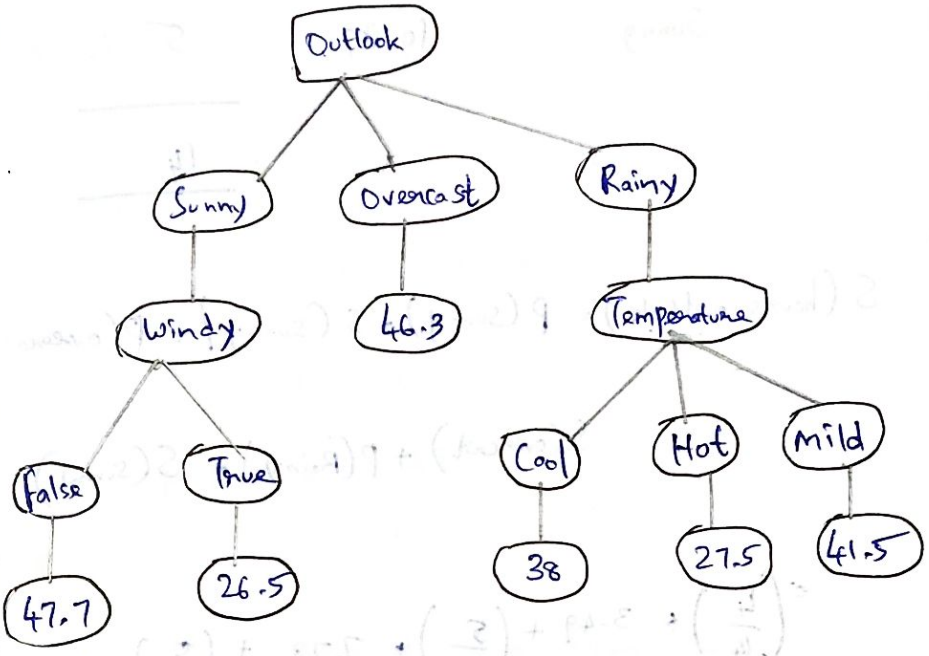
f_1	f_2	o/p	
2.3	5.6	-	Continious values
5.2	7.8	-	
6.9	6.5	-	
4.4	8.4	-	
3.3	4.3	-	

Consider the real time dataset,

Predictors				target
OUTLook	TEMP	HUMIDITY	WINDY	HOURS PLAYED
Rainy	Hot	High	false	26
Rainy	Hot	High	True	30
Overcast	Hot	High	false	48
Sunny	mild	High	false	46
Sunny	Cool	Normal	false	62
Sunny	Cool	Normal	True	23
Overcast	Cool	Normal	True	43
Rainy	Mild	High	false	36
Rainy	Cool	Normal	false	38
Sunny	Mild	Normal	false	48

Rainy	mild	Normal	True	48
Overcast	mild	High	True	62
overcast	Hot	Normal	false	44
Sunny	mild	High	True	30

Decision Tree
Converts to



How?

Firstly, Calculate mean and sd for target variable,

Mean = 39.8

Sd = 9.32

Coefficient of variation $CV = \frac{S}{\bar{x}} * 100 = 23\%$

↳ variation in percentage.

Secondly, Take one feature and Calculate ~~Mean~~ Sd along with target variable,

		Hours played (Std)	Count
Outlook	Overcast	3.49	4
	Rainy	7.78	5
	Sunny	10.87	5
			14

$S(\text{hours, outlook}) = P(\text{Sunny}) * S(\text{Sunny}) + P(\text{overcast}) * S(\text{overcast}) + P(\text{Rainy}) * S(\text{Sunny})$

$= \left(\frac{4}{14}\right) * 3.49 + \left(\frac{5}{14}\right) * 7.78 + \left(\frac{5}{14}\right) * 10.87$

$= 7.66.$

Standard deviation is high, so reduce it,

Something called as Standard deviation reduction.

		Hours played (Std)	~~Count~~
Outlook	Overcast	3.49	SDR = 1.66
	Rainy	7.78	
	Sunny	10.87	

$$SDR(T, x) = S(T) - S(T, x)$$

$$= 9.32 - 7.66 = 1.66$$

↓

$$SDR(\text{Hours}, \text{outlook}) = S(\text{Hours}) - S(\text{Hours}, \text{Outlook})$$

· outlook ⇒ **SDR = 1.66**

Same like,

Humidity SDR = 0.28

Temp SDR = 0.48

Mild SDR = 0.29

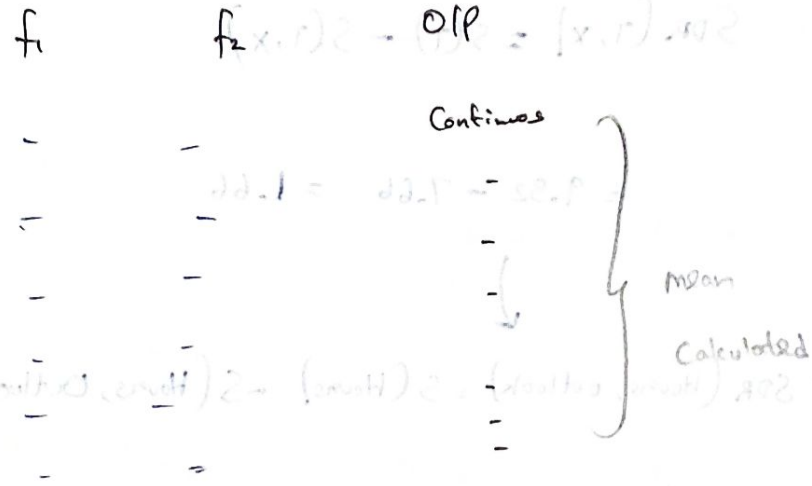
Then set one threshod using hyperparameter,

and too many processes.

Refer: https://www.Saedsayad.com/decision_tree_reg.htm

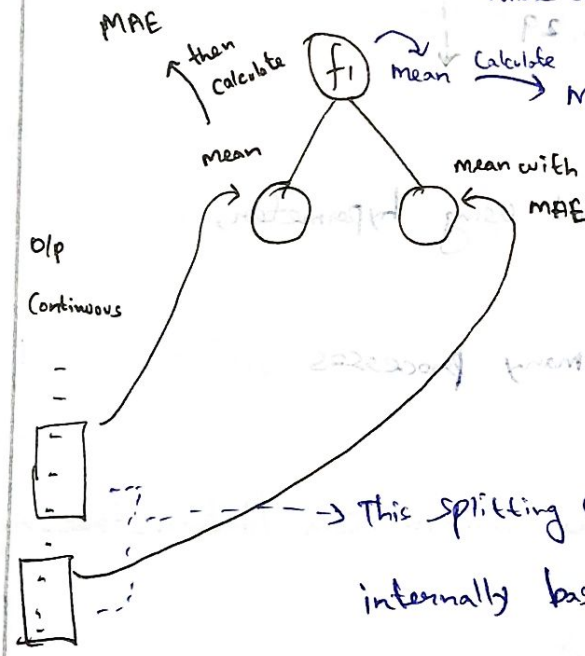
Another method (or)

Decision Tree Regression:



In this method, The Cost function is same as linear regression. Not Entropy ~~and~~ (or) Gini Coefficient.

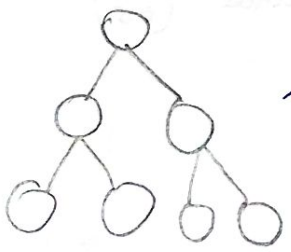
Suppose, In f_1 feature, we assign mean value and Calculate MSE



$$MAE \rightarrow \frac{1}{2m} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$$

(Then, MAE value is something)

→ This splitting Category is applied internally based on the dataset.



→ Likewise Go, until the mse is reduced.

Finally, which mse is low, that Corresponding mean is Output.

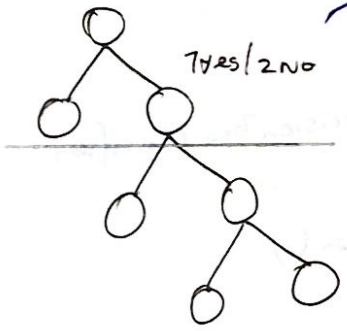
The decision tree have more depth (or) split, The model becomes Complex. It well working for training data and fit. But unable to work testing data because we have Only Some amount of test data so it may be unfit and Causes Overfitting. In Order to avoid that we use, two techniques,

→ Post-pruning.

→ Pre-pruning.

Post pruning method set the after construction of decision tree.

Example,



→ In this case we know 7yes it means 80% of yes category. So, we decide it. So, Cut the remaining split.

The advantages are time complexity is reduced and Prevent Overfitting.

Pre-pruning method pruning the tree before Constructing Using hyperameter.

Hyperparameter:

Max depth, Max_leaf } Inside Grid Search CV

Simple Tree fit Example Problem:

```
import Pandas as pd
import matplotlib.Pyplot as plt
% matplotlib inline

from sklearn.datasets import load_iris
iris = load_iris()
iris.data

from sklearn.tree import DecisionTreeClassifier
Classifier = DecisionTreeClassifier()
Classifier.fit(iris.data, iris.target)

from Sklearn import tree
plt.figure(figsize=(15,10))
tree.plot_tree(Classifier, filled=True)
```

#output: visualize tree for get more information.