05 Oct 2023

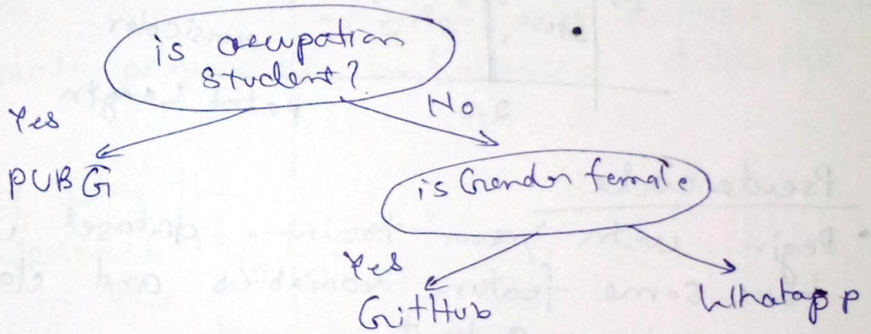
Decision Trees

example-1.

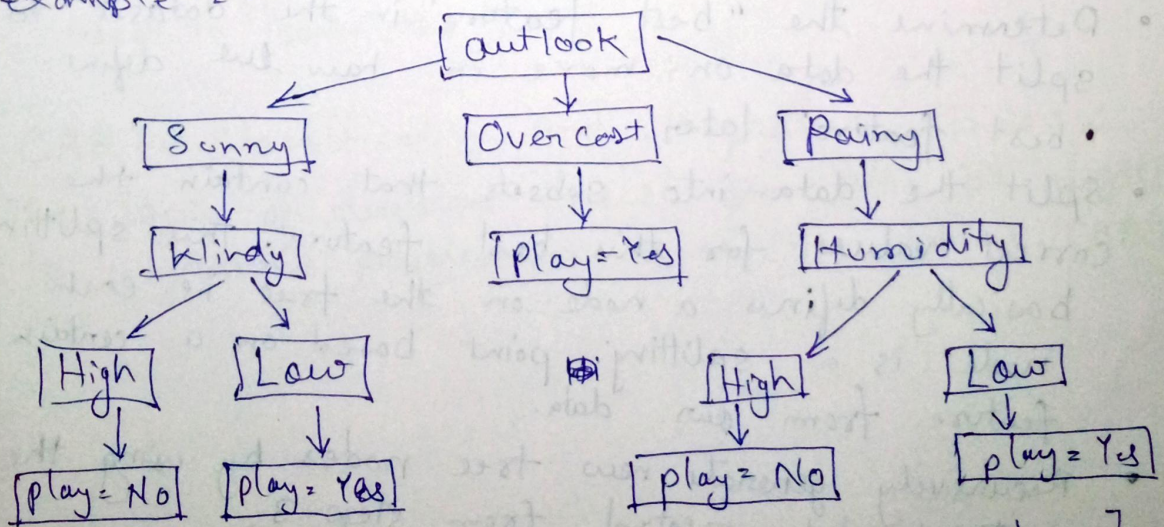
Gender	Occupation	Suggestion
F	Student	PUBG
F	Programmer	GitHub
M	Programmer	Whatsapp
F	Programmer	GitHub
M	Student	PUBG
M	Student	PUBG

Table-1

```
if occupation == student
    print(PUBG)
else
    if gender == female
        print(GitHub)
    else
        print(Whatsapp)
```



example-2.



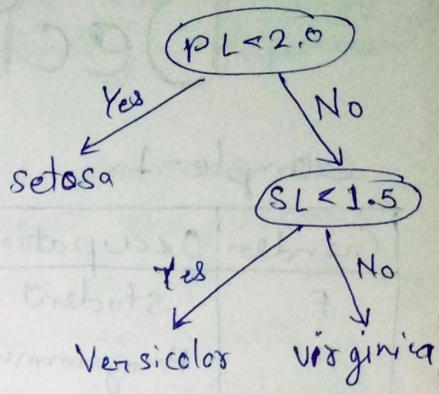
input query point: [Rainy, Mild, High, strong]

output: play=No

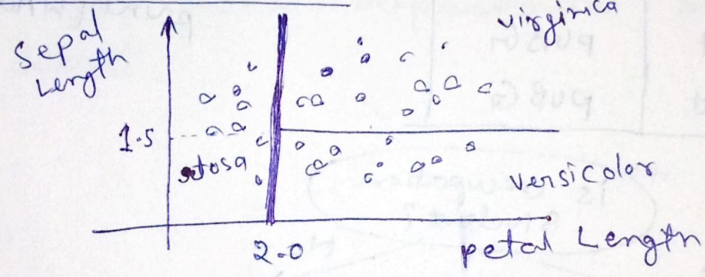
What if we have numerical data?

Petal Length	Sepal Length	Type
1.34	0.34	Setosa
3.45	1.45	Versicolor
1.69	0.98	Setosa
2.56	1.79	Virginica
3.00	1.13	Versicolor
1.3	0.88	Setosa

Table-2



Geometric intuition.



Pseudo code

- Begin with your training dataset which should have some feature variables and classification or regression output.
- Determine the "best feature" in the dataset to split the data on; more on how we define "best feature" later.
- Split the data into subsets that contain the correct values for this best feature. This splitting basically defines a node on the tree i.e. each node is a splitting point based on a certain feature from our data.
- Recursively generate new tree nodes by using the subset of data created from step-3.

Conclusions

Programatically speaking, Decision trees are nothing but a giant structure of nested if-else condition.

Mathematically speaking, Decision trees use hyperplanes which run parallel to any one of the n axes to cut your coordinate system into hyper cuboids.

Advantages and Disadvantages:

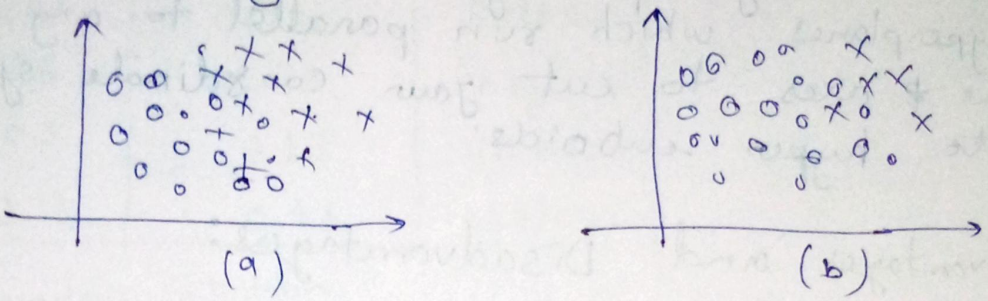
Advantages	Disadvantages
- intuitive and easy to understand - minimal data preparation is required - The cost of using the tree for inference is logarithmic in the number of data points used to train the tree	- overfitting - prone to errors for imbalanced datasets

CART - Classification and Regression Trees:

The logic of decision trees can also be applied to regression problems, hence the name CART

What is Entropy

Entropy is nothing but the measure of disorder. Or you can also call it the measure of purity/impurity



(a) (b)

more knowledge less entropy.
(a) has high impurity.

Calculating Entropy

$$E(S) = \sum_{i=1}^{n} -P_i \log_2 P_i$$

where P_i is simply the frequentist probability of an element/class 'i' in our data.
example: if our data have only 2 class labels Yes and No.

$$E(D) = -P_{Yes} \log_2(P_{Yes}) - P_{No} \log_2(P_{No})$$

Salary	Age	Purchase
20000	21	Yes
10000	45	No
60000	27	Yes
15000	31	No
12000	18	No

Table-3

High entropy

$H(d) = -\frac{2}{5}\log_2 2/5 - 3/5 \log_2 3/5$

$H(d) = 0.97$

Salary	Age	Purchase
34000	31	No
15000	25	No
69000	57	Yes
25000	21	No
32000	28	No

Table-4

$H(d) = -\frac{1}{5}\log_2 \frac{1}{5} - \frac{4}{5}\log_2(4/5)$

$H(d) = 0.72$

calculating entropy for a 3 class problem:

Salary	Age	Purchase
20000	21	Yes
10000	45	No
60000	27	Yes
15000	31	No
30000	30	Maybe
12000	18	No
40000	40	Maybe
20000	20	Maybe

Table- 5

$H(d) = P_Y \log_2(P_Y) - P_N \log_2(P_N) - P_M \log_2(P_M)$

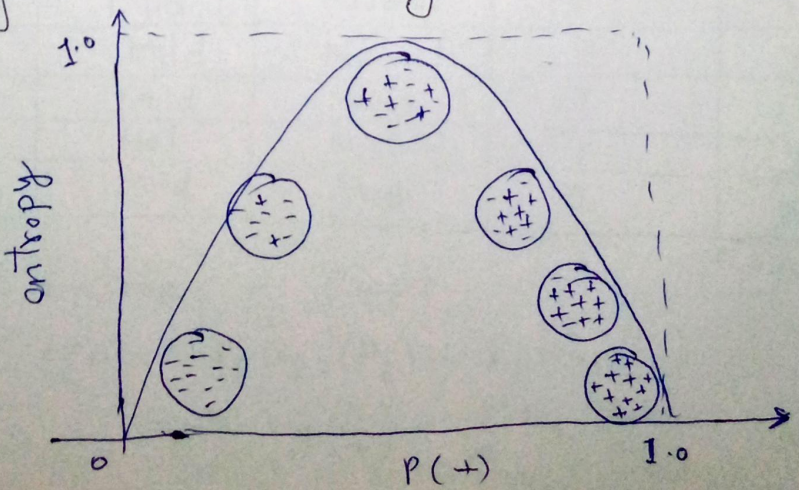
$H(d) = \frac{-2}{8}(\log_2(\frac{2}{8})) - \frac{3}{8}(\log_2(\frac{3}{8})) - \frac{3}{8}\log_2(\frac{3}{8})$

$H(d) = 1.56$

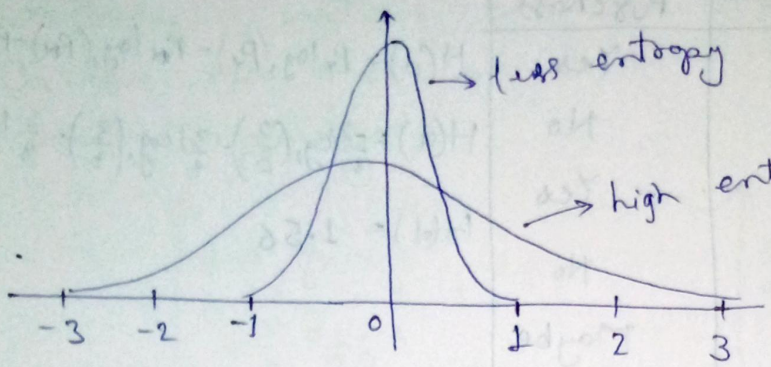
observation:

- more the uncertainty more is entropy
- for a two class problem the min entropy is 0 and the max is 1
- For more than 2 classes the min entropy is 0 but the max can be greater than 1
- Both $\log_2$ or $\log_e$ can be used to calculate entropy. but mostly use $\log_2$

entropy Vs Probability



entropy for continuous variable



if for any data set plot of KDF and less peak of that data set in the graph then that data set have high entropy

Information Gain

Information Gain is a metric used to train Decision Trees. Specifically, this metric measures the quality of a split.
The information gain is based on the decrease in entropy after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain.

$$IG = E(\text{parent}) - \{\text{weighted average}\} * E(\text{children})$$

example:-

S.N.	outlook	Temperature	Humidity	Windy	Play Tennis
1	Sunny	Hot	High	F	No
2	Sunny	Hot	High	T	No
3	Overcast	Hot	High	F	Yes
4	Rainy	Mild	High	F	Yes
5	Rainy	Cool	Normal	F	Yes
6	Rainy	Cool	Normal	T	No
7	Overcast	Cool	Normal	T	Yes
8	Sunny	Mild	High	F	No
9	Sunny	Cool	Normal	F	Yes
10	Rainy	Mild	Normal	F	Yes
11	Sunny	Mild	Normal	T	Yes
12	Overcast	Mild	High	T	Yes
13	Overcast	Hot	Normal	F	Yes
14	Rainy	Mild	High	T	No

Table-6

step-1. Entropy of Parent

$$E(P) = -P_y \log_2(P_y) - P_n \log_2(P_n)$$

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

$$E(P) = 0.94$$

Step-2. Calculate Entropy for children.

outlook

Sunny | Overcast | Rain

$E(S) = -\frac{2}{5}\log\frac{2}{5} - \frac{3}{5}\log\frac{3}{5}$ = 0.97

$E(O) = 0$

$E(R) = \frac{3}{5}\log\frac{3}{5} - \frac{2}{5}\log\frac{2}{5}$ = 0.97

Step-3 Calculate weighted entropy of children

weighted entropy = $\frac{5}{14} \times 0.97 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.97$

W E(children) = 0.69

P(overcast) is a leaf node as it's entropy is 0

Step-4 Calculate Information gain.

IG = {E(parent) − {weighted average}} * E(children)

= 0.97 − 0.69

= 0.28

So the information gain (or the decrease in entropy/impurity) when you split this data on the basis of outlook condition/column is 0.28

Step-5 Calculate information gain for all columns. whichever column has the highest Information gain. (max decrease in entropy) the algorithm will select that column to split the data.

Step-6 Find Information gain recursively

Decision tree then applies a recursive greedy search algo in top bottom fashion to find information gain at every level of the tree.

Once a leaf node is reached (Entropy = 0), no more splitting is done.

Gini Impurity

it is used to measure purity/impurity of nodes after split.

for 2 class classification problem

$E = -P_y(\log_2(P_y) - P_n \log_2(P_n)$

Gini impurity $(G_I) = 1 - (P_y^2 + P_N^2)$

eg. from table-3 and table-4

for table-3

$G_1 = 1 - (P_y^2 + P_n^2) = 1 - (4/25 + 9/25) = 0.48$

for table-4

$G_2 = 1 - (P_y^2 + P_n^2) = 1 - (1/25 + 16/25) = 0.32$

here $G_1 > G_2$ Hence G_2 provide more Information gain.

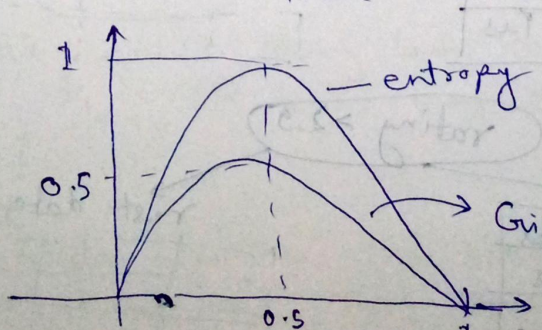
* gini impurity and entropy both are work as same. for selecting root node and their sub-node.
* but the differen is that for Binary classification data $P_y = 1$, $P_N = 0$

then $E = 0$ and $G = 0$

But if $P_y = 0.5$ $P_N = 0.5$

then $E = 1$ $G = 1 - (0.5^2 + 0.5^2)$

$G = 0.5$

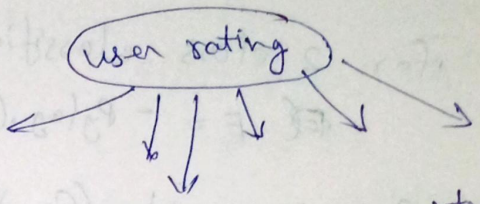


* Gini computational faster than entropy for large data use it but in some cases Gini gives overfitting

Handling Numerical (continuous) Data in DT

SN.	User Rating	Downloaded
1	3.5	Yes
2	4.6	Yes
3	2.2	No
4	1.6	Yes
5	4.1	No
6	3.9	No
7	3.2	No
8	2.9	Yes
9	4.8	Yes
10	3.3	No
11	2.5	Yes
12	1.9	Yes

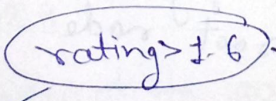
data['user_rating'].unique()=n



we will then have n subtrees

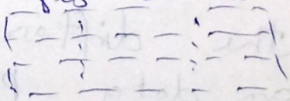
Step-1 sort the data on the basis of numerical column.

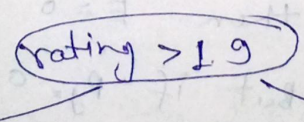
Step-2 split the entire data on the basis of every value of user_rating



1	1.6	Yes

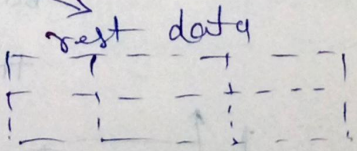
rest data.

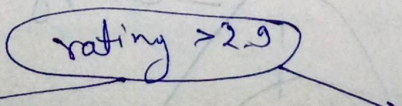




1	1.6	Yes
2	1.9	Yes

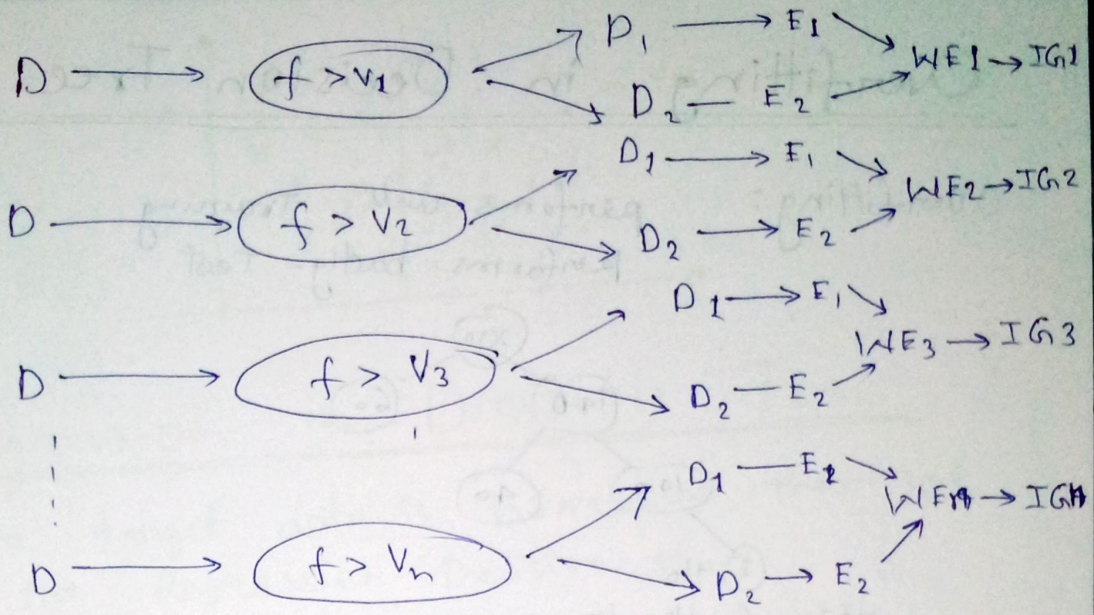
rest data





1	1.6	Yes
2	1.9	Yes
3	2.2	No
4	2.5	Yes
5	2.9	Yes

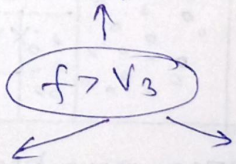
rest data



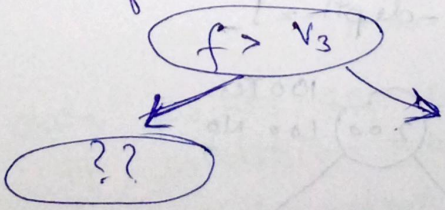
- where D is the dataset
- f is the column user rating.
- $V_1, V_2, \ldots V_n$ are the values of various raws of user Rating

Step-5 $\text{Max}\{IG_1, IG_2, IG_3, \ldots IG_n\}$

let us assume IG_3



Step--6 Do this recursively until you get all the leaf nodes.



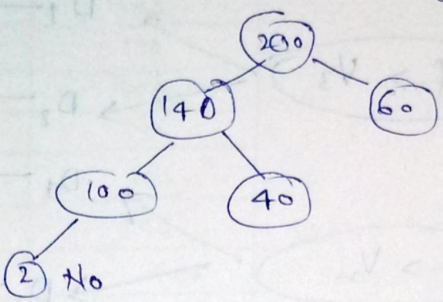
Question - This entire thing looks computationally too expensive. Is is the right way of finding the splitting criteria?

Answer - Yes

For this particular algorithm the train time complexity is higher than the Test time complexity ($\log n$)

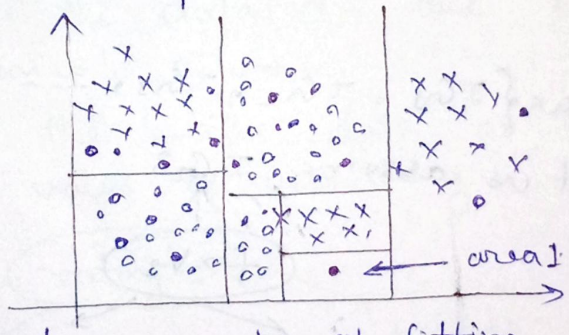
Overfitting in Decision Tree:

Overfitting: performs well = Training
performs badly - Test



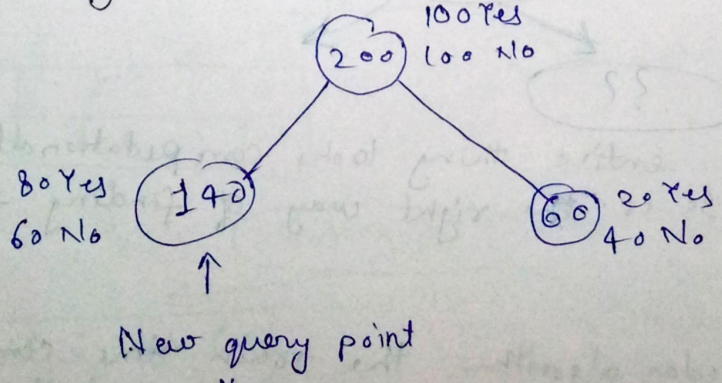
noisy / outlier / Erroneous

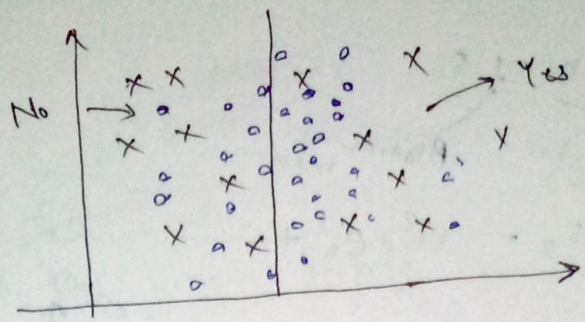
and if we not select the max-depth parameter then there is a huge chance there is overfitting occur. Or max_depth = None



very large number of fitting due to this if near future 'x' is occur / land in area1 then there is overfitting occur.

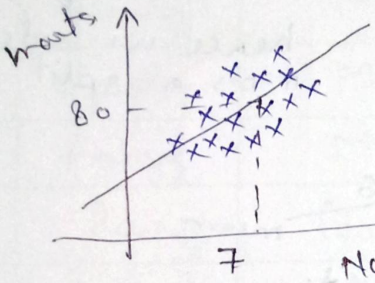
underfitting: max-depth = 1.





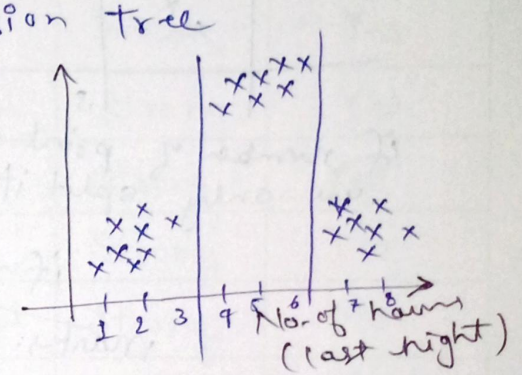
Regression in Dicision Tree

if the target data is numerical then we use the Regression Decision tree

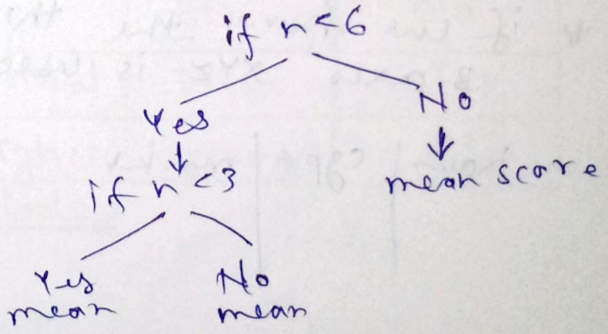


Linear Regression

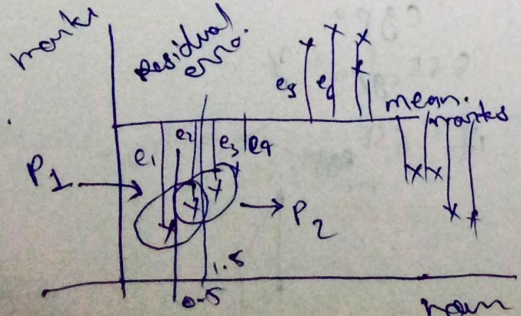
$y = mx + c$

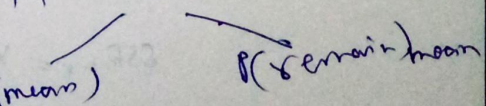


decision tree regression



if train data points are continuous then How we select the if condition eg. if n≥6, if n<3 it means how we select the spliting criteria





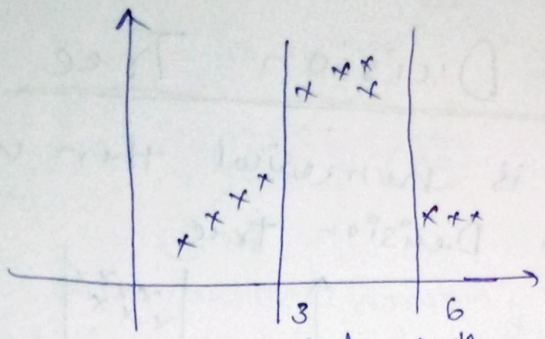
$SSE_1 = e_1^2 + e_2^2 + \dots e_n^2$

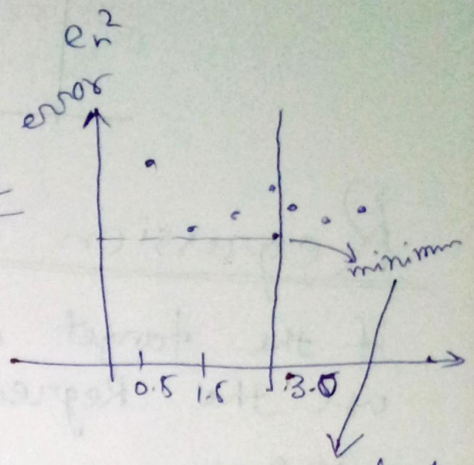
$SSE_2 =$

if $n < 1.5$

P_1 (mean) P_2 (mean)

$SSE_2 = e_1^2 + e_2^2 + \cdots e_n^2$





if number of point = 4 then we only split it (threshold value)

hence we select it as a split

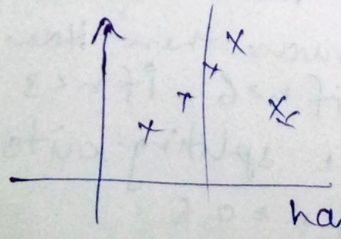
if $n < 3$ → marks.

if $n < 6$ → mean. → mean

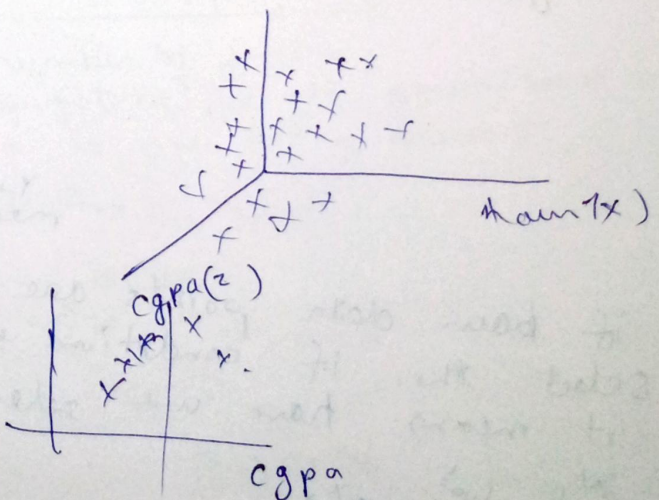
the above process is continued.

* if we have the three columns. it mean 3-axes XYZ is used.

han	cgpa	marks



$SSE_{hr} = X$



$SSE_{cgpa} = y$

if $SSE_m < SSE_{cgpa}$

if $SSE_{hr} > SSE_{cgpa}$

Ques- Based on given table make a decision tree.

Day	Weather	Temperature	Humidity	Wind	play
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	~~Rainy~~ Sunny	Mild	Normal	Strong	Yes
4	cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak.	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak.	Yes
10	Rainy	Mild	High.	Strong	No.

(1.) Finding the root Node:

entropy of whole data.

$$E(S) = -\sum_{i=1}^{n} P_i \log_2 P_i$$

$$= -P(Yes)\log_2 P(Yes) - P(No)\log_2(No)$$

$$= -\frac{5}{10}\log_2\frac{5}{10} - \frac{5}{10}\log_2\frac{5}{10}$$

$$= 1$$

Now calculate information gain.

(i) weather — IG(S, weather)

(ii) Temperature — IG(S, temp)

(iii) Humidity — IG(S, Humidity)

(iv) Wind — IG(S, Wind)

(I) calculat the IG for weather

Sunny [1+, 2-]
cloudy [3+, 0]
Rainy [1+, 3-]

entropy for sunny $= -\sum_{i=1}^{n} p_i \log_2 p_i$

$= -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3}$

$= 0.91$

entropy for cloudy $= -\frac{3}{3}\log_2\frac{3}{3} - \frac{0}{3}\log_2\frac{0}{3}$

$= 0$

entropy for Rainy $= -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4}$

$= 0.8112$

$IG(S, \text{weather}) = \text{entropy}(S) - \sum \frac{|S_v|}{|S|} \text{entropy}(S_v)$

$= 1 - \frac{3 \times 0.91}{10} - \frac{4 \times 0.81}{10}$

$IG(S, \text{weather}) = 0.403$

(II) calculat the IG for Temperature

Hot: [2+, 2-]
Mild: [3+, 2-]
Cool: [0, 1-]

entropy for Hot $= -\sum_{i=1}^{n} p_i \log_2 p_i$

$= -\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}$

$= 1$

entropy for Mild $= \frac{-3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}$

$= 0.9709$

entropy for cool = $-\frac{0}{1}\log_2\frac{0}{1} - \frac{1}{1}\log_2\frac{1}{1}$

$= 0$

$IG(S, Temp) = 1 - \frac{4}{10} \times 1 - \frac{5}{10}\ 0.97095$

$IG(S, Temp) = 0.114$

(III) Calculate the IG. for Humidity

High = [3+, 4-]

Normal = [2+, 1-]

entropy for High = $-\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7}$

$= 0.9852$

entropy for Normal = $-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}$

$= 0.9182$

$IG(S, Humidity) = 1 - \frac{7}{10} \times 0.9852 - \frac{3}{10} \times 0.9182$

$IG(S, Humidity) = 0.0349$

(IV) Calculate the information gain for wind.

weak: [3+, 1-]

Strong: [2+, 4-]

entropy for weak = $-\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}$

$= 0.8112$

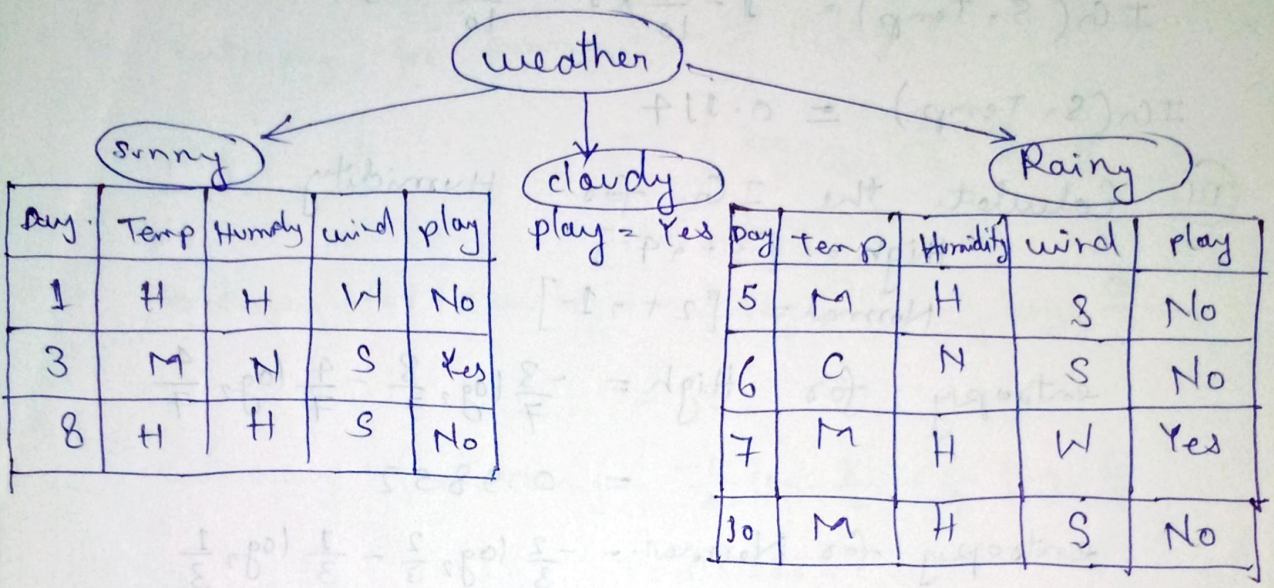
entropy for strong = $\frac{2}{6}\log_2\frac{2}{6} - \frac{4}{6}\log_2\frac{4}{6}$

$= 0.9182$

$IG(S, wind) = 1 - \frac{4}{10} \times 0.8112 - \frac{6}{10} \times 0.9182$

$IG(S, wind) = 0.1246$

~~IG(S, weath)~~ IG(S, weather) is greater than the all other Information gain of the colomnss. Hence root node is weather.



Day	Temp	Humdy	wind	play
1	H	H	W	No
3	M	N	S	Yes
8	H	H	S	No

Day	Temp	Humidity	wind	play
5	M	H	S	No
6	C	N	S	No
7	M	H	W	Yes
10	M	H	S	No

2. Table Sunny as a Node:

entropy of Sunny dataset

$$= -\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3} = 0.9183.$$

(I) calculate I G. for Temperatn.
entropy of its children = 0

IG(Sunny, Temp) = 0.9183

(II) calculate information gain for Humidity.
entropy of its children = 0

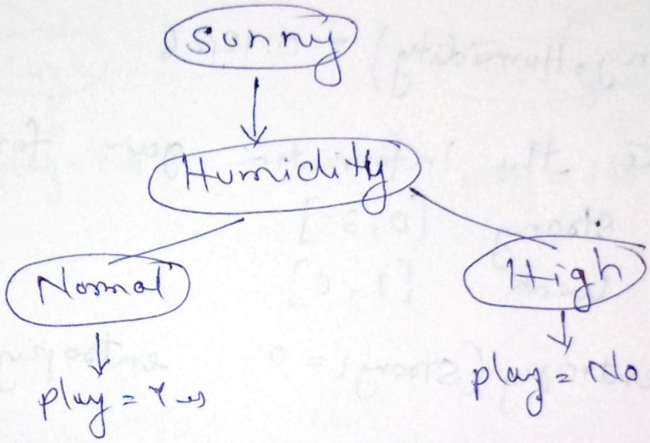
IG(Sunny, Humidity) = 0.9183

(III) calculate the information gain for wind.
entropy for its children
Strong: [1, 1]
Weak: [0, 1]

$$IG(Sunny, wind) = 0.9183 - \frac{2}{3}\left(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}\right)$$

$$IG(Sunny, wind) = 0.2516$$

for the sunny node information gain of Humidity is highest hence. Hence it is next node after sunny node.



(3.) Take Rainy as a Node:

entropy for the Rainy dataset

$$= -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} = 0.8113$$

(I) calculate the IG for temp.

Mild: [1, 2-]

Cool: [0, 1-]

$$entropy(mild) = \frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.6887$$

$$IG(Rainy, temp) = 0.8113 - \frac{3}{4} \times 0.6887$$

$$IG(Rainy, temp) = 0.1226$$

(II) calculat the information gain for Humidity.

High : [1+, 2-]

Normal : [0, 1-]

entropy(High) $= -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.6887$

entropy(Normal) $= 0$

$IG(Rainy, Humidity) = 0.8113 - \frac{3}{4} \times 0.6887$

$IG(Rainy, Humidity) = 0.1226$

(III) Calculate the information gain for wind.

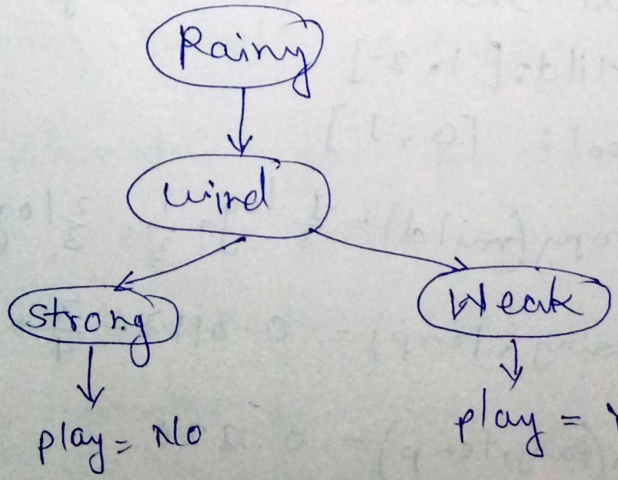
Strong [0, 3-]

Weak [1, 0]

entropy(Strong) $= 0$ entropy(weak) $= 0$

$IG(Rainy, wind) = 0.8113$

In the above we can see the the IG(Rainy, wind) is highest in respect to other Hence next node to Rainy is wind.



The final decision tree of above (question) table is: