

scaling is not needed for or

② Decision Tree

for both Regression & classification
(supervised learning)

→ As of now we have Regression models,

which makes a best fit line.

or a best fit polynomial.

→ But In case of classification model,

we have this logistic regression,

which can only classify by using lines.

* what if our data is classified in polynomial shape?



Logistic Regression

can't be used here

drawback

So, this is where "decision tree" came to picture

→ As the name suggests, this algorithm works by dividing the whole dataset into a

"tree-like structure" based on some rules & conditions & then gives predictions based on those conditions. Let's see how it works.

1 → First it selects a root node based on given cond.



2 → then the root node will split into child nodes based on cond.

3 → If any node doesn't fulfill any cond., then it again splits into new cond.

4 → continue till all the cond. are met or if you have pre-defined the depth of your tree

Q) But, how do we select a root node?

Ans: This is where mathematical induction comes in.

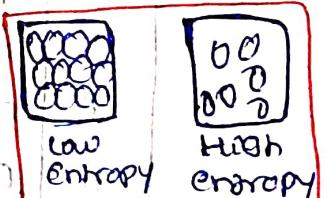
- usually, Regression tree are used for quantitative data.
- For scale or qualitative/categorical data we use classification trees.
- In regression tree, we split the nodes based on RSS (Residual Sum of Squares) criteria.

In classification, it is done using classification error rate, Gini impurity & Entropy.

We need a pure element (root node) to build tree.
So to find that we can use Entropy.

* Entropy is the measure of randomness of data.
It gives the impurity present in the dataset.

- When we split our node into two regions and put different observations in both the regions, the main goal is to reduce the randomness in the region & divide our data cleanly than it was in the previous node. If splitting the node doesn't lead to entropy reduction, we try to split based on different cond, or we stop.



- A region is **clean (low entropy)** when it contains data with the same labels & random if there is a mixture of labels present (**high entropy**).

We need to calculate Entropy for each column u.

$$E = -P \cdot \log(P)$$

P = Probability,
 $P = \frac{n}{N}$

$$E = P \cdot \log P = (P \cdot \log_2 P)$$

④ Information gain (G_n) calculates the decrease in entropy after splitting a node. It is the difference b/w entropies before & after the split. The more the information gain, the more entropy is removed, & more clean the node.

$$G(n) = 1 - \sum \left(\frac{s_i}{S} \cdot E_i \right)$$

$$G(n) = \text{Entropy}(T) - \text{Entropy}(T'|n)$$

Q) Let's take three data,

n	x	y	class
1	1	1	I
1	1	0	I
0	0	1	II
1	0	0	II

At we introduced new data,

new row $\begin{pmatrix} 1 & 0 & 1 \end{pmatrix}$, what is class?

First we need individual column entropy.

(E_x, E_y, E_z)

In each column we have (1, 0) & total 2 class (I, II).

$$E_x(I) = \left(\frac{-2}{3} \cdot \log \frac{2}{3} \right) + \left(\frac{-1}{3} \cdot \log \frac{1}{3} \right) = -0.276$$

we have '3' 1's in x column,
out of which '2' are I class

'1' out of 3 is II class.

I

$$E_0(0) = \left(\frac{1}{2} \cdot \log \left(\frac{1}{2} \right) \right) + \left(\frac{1}{2} \cdot \log \frac{1}{2} \right) = 0$$

$$E_1(1) = \left(\frac{1}{2} \cdot \log \frac{1}{2} \right) + \left(\frac{1}{2} \cdot \log \frac{1}{2} \right) = 0$$

$$E_2(0) = \left(\frac{1}{2} \cdot \log \frac{1}{2} \right) + \left(-\frac{1}{2} \cdot \log \frac{1}{2} \right) = 0$$

$$E_2(1) = \left(-\frac{1}{2} \cdot \log \frac{1}{2} \right) + \left(-\frac{1}{2} \cdot \log \frac{1}{2} \right) = 1$$

$$E_2(0) = \left(-\frac{1}{2} \cdot \log \frac{1}{2} \right) + \left(-\frac{1}{2} \cdot \log \frac{1}{2} \right) = 1$$

Information Gain

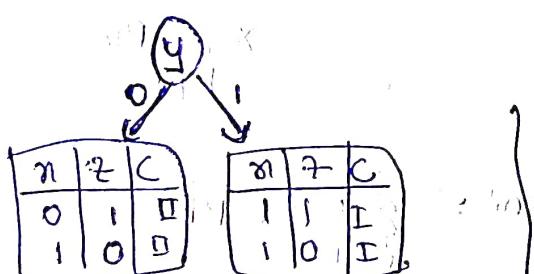
$$G_S = 1 - \left[\frac{S(0)}{S} \cdot E(1) + \frac{S(1)}{S} \cdot E(0) \right] \quad \text{Total no. of evaluation}$$

$$G_S = 1 - \left[\frac{3}{4} \times 0.276 + \frac{1}{4} \times 0 \right] \approx 0.276$$

$$G_Y = 1 - \left[\frac{2}{4} \times 0 + \frac{2}{4} \times 0 \right] = 1$$

$$G_Z = 1 - \left[\frac{2}{4} \times 1 + \frac{2}{4} \times 1 \right] = 0$$

$G_Y = 1$, is the highest info gain, so low entropy.
Hence it is
Root Node: (Y)



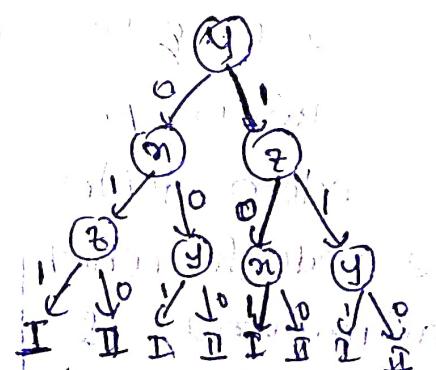
Again we have to find E_0, E_1, G_0, G_1 .

Find largest if parent node.

until leaf node gives a class.

Find E_0, E_1, G_0, G_1 .

Finally our can be like this.



So, $140-1 \rightarrow \text{I class}$

Gini Impurity

Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labelled if it was randomly labelled according to the distribution of labels in the subset.

→ It is calculated by multiplying the probability that a given observation is classified into the correct class & sum of all the probabilities when that particular observation is classified into the wrong class.

→ Let's suppose there are k number of classes and an observation belongs to the class i^{th} , & its probability is given by P_i .

Then probability that observation belongs to any other class other than i^{th} is,

$$\sum_{k \neq i} P_k = 1 - P_i$$

then gini impurity = $\sum_{i=1}^k P_i \times \sum_{k \neq i} P_k$

gini impurity value

will be 0 for P_i ,

0 - no impurity

1 - Random distn.
the node for which

gini impurity is least is selected
as Root node
to split

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

$$Gini = \sum_{i=1}^k P_i - \sum_{i=1}^k P_i^2$$

$$Gini = \sum_{i=1}^k P_i - \sum_{i=1}^k P_i^2$$

$$Gini = 1 - \sum_{i=1}^k P_i^2$$

(b) cover data

<u>class</u>	<u>gender</u>	<u>Stay in hotel</u>
9	M	Yes
10	F	No
8	F	Yes
8	F	No
9	M	Yes
10	M	No
11	F	Yes
11	M	Yes
8	F	Yes
9	M	No
11	M	No
10	M	Yes
10	M	No

Let's understand how root node is selected by Gini impurity.

We have 2 features which can use for hotel class & gender. We will calculate Gini for each of the features & then select that feature which has least Gini Impurity.

Q1

<u>class</u>	<u>stay</u>	<u>Total</u>	<u>P(class)</u>	<u>P(stay)</u>
8	Yes = 2, No = 1	3	8/14	2/3
9	Yes = 2, No = 1	3	2/14	2/3
10	Yes = 1, No = 3	4	4/14	1/4
11	Yes = 3, No = 1	4	4/14	3/4

Gini for class Feature

$$G(class=8) = 1 - (P(class))^2 - (P(stay))^2 = 1 - (4/14)^2 - (2/3)^2 = 4/9$$

$$G(class=9) = 1 - (2/14)^2 - (2/3)^2 = 4/9$$

$$G(class=10) = 1 - (4/14)^2 - (1/4)^2 = 6/16$$

$$G(class=11) = 1 - (4/14)^2 - (3/4)^2 = 6/16$$

Weighted sum of Gini for class

$$G(class) = \frac{\text{no. of class 8}}{\text{Total}} \times G(class=8) + \frac{\text{no. of class 9}}{\text{Total}} \times G(class=9) + \dots$$

$$G(class) = \frac{8}{14} \times \frac{4}{9} + \frac{3}{14} \times \frac{4}{9} + \frac{4}{14} \times \frac{6}{16} + \frac{4}{14} \times \frac{6}{16}$$

$$G(class) = 0.404$$

ii) ARI for gender

			<u>D(4)</u>	<u>P(N)</u>
<u>Gender</u>	<u>start</u>	<u>n</u>	<u>5/8</u>	<u>3/8</u>
m	year 25, no 23	8	1/2	1/2
F	year 3, no 3	6		

$$a(m) = 1 - (5/8)^2 - (3/8)^2 = 0.167$$

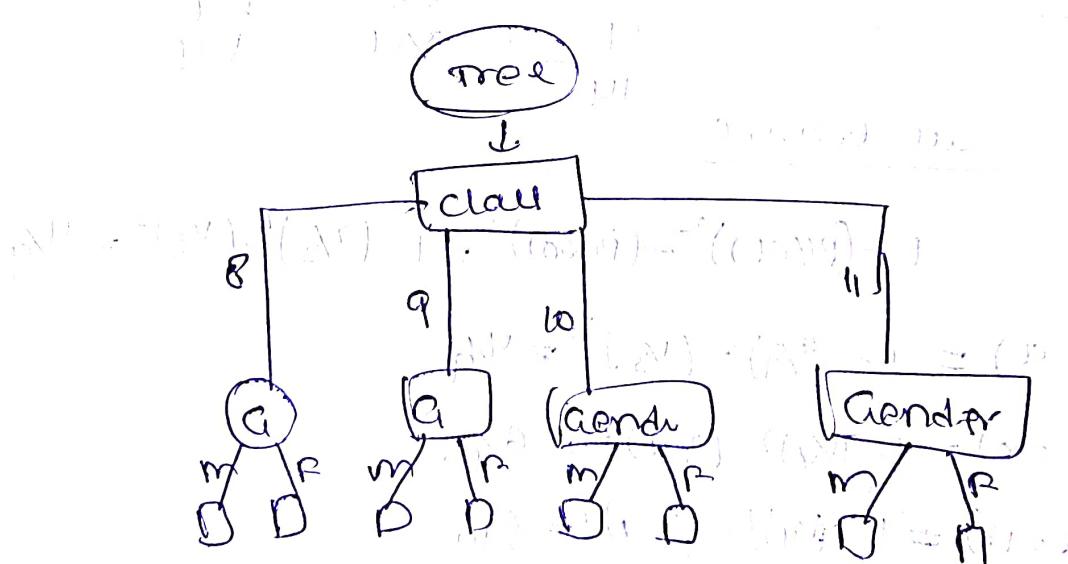
$$a(F) = 1 - (1/2)^2 - (1/2)^2 = 0.5$$

$$a(\text{Gender}) = \frac{8}{14} \times 0.167 + \frac{6}{14} \times 0.5$$

$$a(\text{Gender}) = 0.16722$$

we can say

$$a(\text{Class}) < a(\text{Gender})$$



this is how decision tree node is selected by calculating gini impurity for each node individually if other features, then we need to repeat same process after selecting each nodes

Note F

- ① we should control growth of tree by "min-sample-split" for which we will not, because a infinitely grown tree with depth (~100) also therefore a high chance at time of prediction. it may go "overfitting"

Decision Tree for Regression

→ when performing regression with a decision-tree, we try to divide them given values or x into distinct & non-overlapping regions.

Eg1 For a set of possible values, x_1, x_2, \dots, x_p , we will try to divide them into J distinct & non-overlapping regions R_1, R_2, \dots, R_J . For a given observation falling into the region " R_j ", the prediction is equal to the mean of the response value for each training observation in the Region R_j .

- ② Regions are selected in a way to reduce residual sum,

$$\sum_{j=1}^J E[(y_i - \hat{y}_j)^2]$$

↳ mean draw the response var in the regions!

Disadv of DT

- | | | |
|---|--|---------------------------------------|
| ① Small change in data can cause instability in the model because of Greedy approach. | ② Proba of overfitting is very high for DT | ③ take more time to train a DT model, |
|---|--|---------------------------------------|

Recursive Binary splitting (Greedy Approach)

- As mentioned, we try to divide the X values into S regions, but is very expensive in terms of computational time to try to fit every set of X values into S regions.
- Thus decision tree opt for a greedy approach in which nodes are divided into regions based on the given condition is called greedy, bcoz it does the best split at a given step at that point of time rather than looking for splitting a step for a better tree in upcoming steps.
- It decides threshold to divide others into different regions.

$$\sum_{i \in R_1(j, s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i \in R_2(j, s)} (y_i - \hat{y}_{R_2})^2$$

These process have a high chance of overfitting the training data as it is very complex.

Tree Pruning

→ It is the method of trimming down a full tree to reduce the complexity & variance in the data. Just as we regularized linear regression, we can also regularize the decision tree model by adding a new term.

$$\sum_{m=1}^{|T|} \sum_{i \in T_m} (y_i - \hat{y}_{i,m})^2 + \alpha |T|$$

where,

'T' is the subtree which is a subset of full tree T_0 . ' α ' is non-negative tuning parameter, which penalizes the MSE with an increase in tree length.

Post-Pruning (Backward pruning)

→ It is the process where DT is generated first & then the non-significant branches are removed. Cross-validation set of data is used to check the effect pruning & test whether expanding a node will improve the improvement or not. If node is having improvement, we convert it to leaf node.

Pre-Pruning (Forward Pruning)

→ It stops the non-significant branch from generating, or use a condition to decide when should it terminate splitting of some of the branch prematurely at tree is generated.

* Different algorithms for Decision tree

① ID3 (Iterative Dichotomiser) :-
It is one of the algorithms used to construct decision trees for classification. It uses information gain as the criteria for finding the root node and splitting them.

It only accepts categorical attributes

② C4.5 :-

It is an extension of ID3 algorithm.
It is better than ID3 as it deals with both continuous & discrete values.

Also used for classification purpose.

③ Classification & Regression Algo. (CART) :-

It is the most popular algorithm for constructing decision trees. It uses "gini" impurity as the default calculation for selecting root node, however one can use "entropy" for criteria as well. Both for regression & classification problem.

Advantage of Gini is easy to calculate, Entropy complicated.

which is why CART uses Gini.