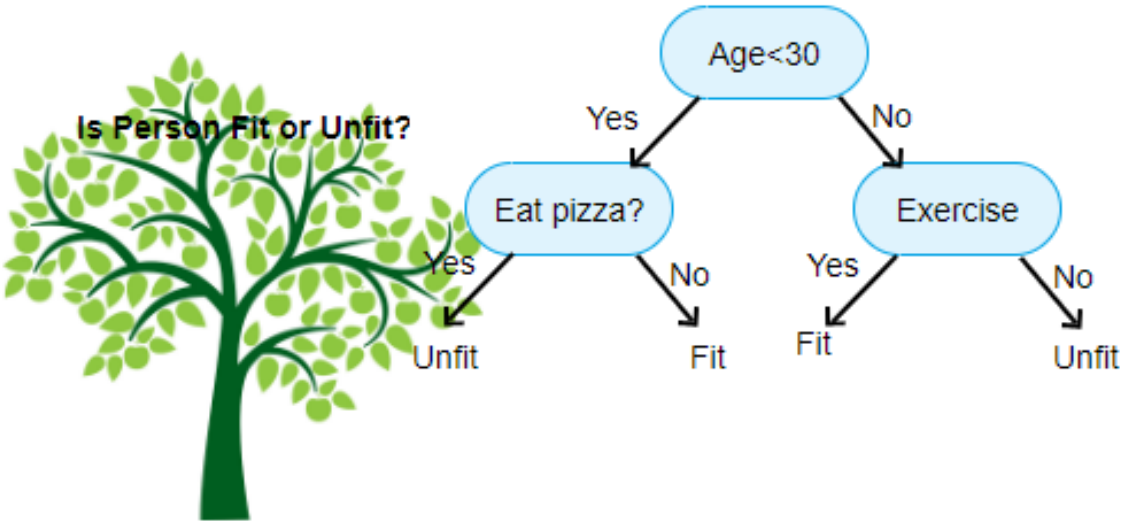
Decision Tree :



Sai Subhasish Rout

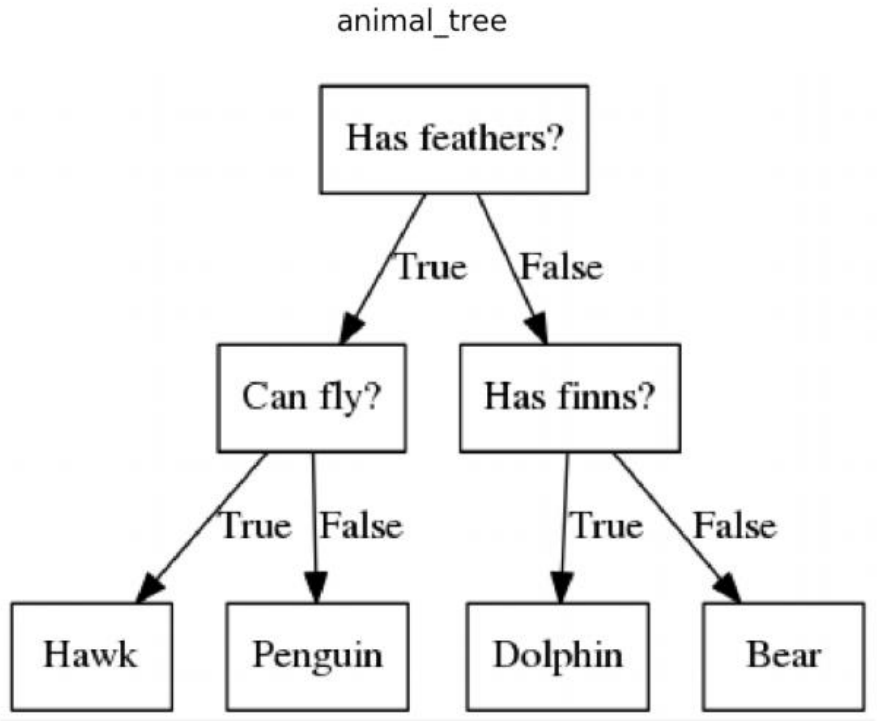
Connect with me: https://www.linkedin.com/in/sai-subhasish-rout-655707151/

Decision Tree is a supervised algorithm. This is one of the most powerful and easy to interpret algorithm. It can be used to visualize and it represents decisions.

In decision tree when we try to give data it will try to separate the dataset in if else type/ binary tree manner, where each internal node denotes a condition, each branch gives the outcome of the condition, and the end of the branch which doesn't split anymore is where we make decision is called leaf node.

We can use decision tree for any kind of problem but It works well with complex dataset for both regression and classification problem. We build decision tree based on output class.

By selecting the right root node we will get the leaf node quickly.



In Decision Tree the model selects the feature as root node where we get leaf node in less step, which is calculated using Information Gain(for classification → Entropy/Gini Impurity) or Reduction in Variance(for regression → MSE)

To get the leaf node quickly we need to select right internal node, for that we need to check purity in each step. For inner node to separate data points we will create conditions, based on which the tree splits into branches. Splitting goes on till we get a pure leaf node or clear separation.

If we will use the algorithm in the background it is performing operations like which features to choose and what conditions to use for splitting, along with knowing when to stop splitting.

For Decision Tree training time complexity will be more and testing time complexity will be less.

Decision Tree Classifier :

Decision Tree Classifier is where the output or decision we have to make with classes/ categorical variable.

For multi-class classification also we use decision tree.

For classification problem to check the purity or randomness of the features we have two most popular ways

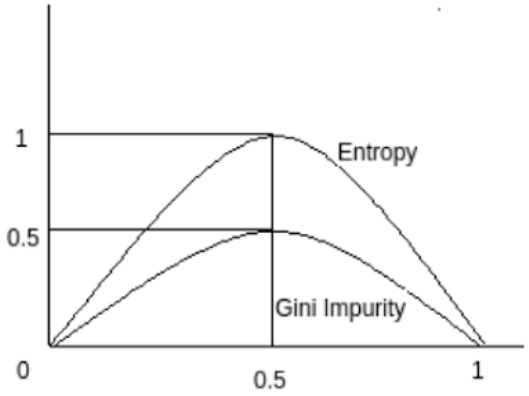
1. Entropy → ID3 algorithm
2. Gini impurity → CART

Entropy:

Entropy means randomness/Degree of freedom. It defines the purity of the feature. Value of Entropy always range between 0 to 1. Based on Entropy of root feature and internal nodes we will calculate Information Gain.

ID3 algorithm implements Entropy.

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i$$



In case of a high impure split (0.5Y/0.5N) the entropy value will be 1. The more the value of entropy towards 1, the more impure the feature is.

We calculate entropy before and after creation of tree. Information Gain is the difference between before and after dividing the data.

We will choose the column with less randomness or entropy and high Information Gain.

Low Entropy → More Stableness → High Information Gain → High Purity

Gini Impurity:

$$Gini\ Impurity = 1 - \sum_{i=1}^{K} p_i^2$$

$$= 1 - Gini\ Index$$

where K is the number of class labels,

p_i is the proportion of i^{th} class label

We find-out Gini-impurity with respect to every feature or classes. And select the feature with low Gini-impurity.

CART algorithm implements Gini-impurity.

Entropy and Gini impurity are the algorithms which works well with categorical data. But most of the time Gini-impurity is recommended to use than Entropy as it takes less time for computation because of simple maths and Entropy takes more time for logarithmic calculation.

For less data we can use Entropy.

Information Gain:

We find-out information gain with respect to every feature.

To find-out information gain we can use both Entropy and Gini-impurity.

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in D_A} \frac{|S_v|}{|S|} Entropy(S_v)$$

Feature with high Information Gain will be selected as Root Node.

Here Information Gain should be high and Impurity should be less.

Calculation for both Entropy and Gini-Impurity:

Lets see an example to decide the root node the calculation for Information Gain for Play Tennis Dataset:

Day	outlook	temperature	humidity	wind	Decision
1	sunny	hot	high	weak	No
2	sunny	hot	high	strong	No
3	overcast	hot	high	weak	Yes
4	rainfall	mild	high	weak	Yes
5	rainfall	cool	normal	weak	Yes
6	rainfall	cool	normal	strong	No
7	overcast	cool	normal	wtrong	Yes
8	sunny	mild	high	weak	No
9	sunny	cool	normal	weak	Yes
10	rainfall	mild	normal	weak	Yes
11	sunny	mild	normal	strong	Yes
12	overcast	mild	high	strong	Yes
13	overcast	hot	normal	weak	Yes
14	rainfall	mild	high	strong	No

Decession Tree

Play Tennes Dataset

- TO Fendout the root node we will follow 2 approaches.
  1. Information Gain → Max Value
  2. Gini Impurity. → Min Value

Information Gain

$$\text{Entropy} = \sum -P \log_2(P)$$

Label → Play Tennes

Output → Y=9 N=5

$$E(L) = -P(Y)\log P(Y) - P(N)\log P(N)$$

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

$$= -\frac{9}{14} \times (-0.64) - \frac{5}{14}(-1.49)$$

$$= 0.41 + 0.54$$

$$= 0.95$$

Entropy of Individual class in feature)

Outlook	Play Tennes	TV	P(Y)	P(N)
sunny	Y=2 N=3	5	2/5	3/5
overcast	Y=4 N=0	4	4/4	0/4
rainfall	Y=3 N=2	5	3/5	2/5

$$E(\text{sunny}) = -\left(\frac{2}{5}\right)\log\left(\frac{2}{5}\right) - \left(\frac{3}{5}\right)\log\left(\frac{3}{5}\right)$$

$$= -0.4 \times (-1.32) - 0.6 \times (-0.74)$$

$$= 0.53 + 0.44$$

$$= 0.97$$

$$E(\text{overcast}) = -\left(\frac{4}{4}\right)\log\left(\frac{4}{4}\right) - \frac{0}{4}\log\left(\frac{0}{4}\right)$$

$$= 0$$

$$E(\text{rainfall}) = -\left(\frac{3}{5}\right)\log\left(\frac{3}{5}\right) - \left(\frac{2}{5}\right)\log\left(\frac{2}{5}\right)$$
$$= -0.6 \times (-0.74) - 0.4 \times (-1.32)$$
$$= 0.42 + 0.53$$
$$= 0.97$$

$$E(\text{class}) = \sum \frac{\text{No of observations}}{\text{Total no. of observations}} \times E_i$$

$$E(\text{outlook}) = \frac{5}{14} \times 0.97 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.97$$
$$= 0.35 + 0.35$$
$$= 0.70$$

$$\text{Information Gain} = E_{\text{before}} - E_{\text{after}}$$

$$IG = 0.95 - 0.70$$
$$= 0.25$$

Temperature	Play Tennes	TV	P(Y)	P(N)
hot	Y= 2 N=2	4	2/4=1/2	2/4=1/2
mild	Y= 4 N= 2	6	4/6=2/3	2/6 = 1/3
cool	Y= 3 N= 1	4	3/4	1/4

$$E(\text{mild}) = -\frac{2}{3}\log\left(\frac{2}{3}\right) - \frac{1}{3}\log\left(\frac{1}{3}\right)$$
$$= -\frac{2}{3} \times (-0.59) - \frac{1}{3} \times (-1.59)$$
$$= 0.39 + 0.53 = 0.92.$$

$$E(\text{hot}) = -\frac{1}{2}\log\left(\frac{1}{2}\right) - \frac{1}{2}\log\left(\frac{1}{2}\right)$$
$$= -\frac{1}{2}(-1) - \frac{1}{2}(-1)$$
$$= 1.$$

$$E(\text{cool}) = -\frac{3}{4}\log\left(\frac{3}{4}\right) - \frac{1}{4}\log\left(\frac{1}{4}\right)$$
$$= -\frac{3}{4}(-0.42) - \frac{1}{4}(-2)$$
$$= 0.32 + 0.5$$
$$= 0.82$$

$E(\text{Temperature}) = \frac{4}{14} \times 1 + \frac{6}{14} \times 0.92 + \frac{4}{14} \times 0.82$

$= 0.28 + 0.39 + 0.23$

$= 0.90$

$I.G = 0.95 - 0.90$

$= \underline{0.05}$

Humedity	Play Tennes	IV	P(Y)	P(N)
high	Y= 3 N= 4	7	3/7	4/7
normal	Y= 6 N= 1	7	6/7	1/7

$E(\text{high}) = -\frac{3}{7}\log\left(\frac{3}{7}\right) - \frac{4}{7}\log\left(\frac{4}{7}\right)$

$= -\frac{3}{7}(-1.22) - \frac{4}{7}(-0.81)$

$= 0.52 + 0.46$

$= 0.98$

$E(\text{normal}) = -\frac{6}{7}\log\left(\frac{6}{7}\right) - \frac{1}{7}\log\left(\frac{1}{7}\right)$

$= -\frac{6}{7}(-0.22) - \frac{1}{7}(-2.80)$

$= 0.19 + 0.4$

$= 0.59$

$E(\text{Humedity}) = \frac{7}{14} \times 0.98 + \frac{7}{14} \times 0.59$

$= 0.49 + 0.295$

$= 0.79$

$IG = 0.95 - 0.79$

$= \underline{0.16}$

Wind	Tennes play	IV	P(Y)	P(N)
Weak	Y = 6 N = 2	8	$6/8 = \frac{3}{4}$	$2/8 = \frac{1}{4}$
Strong	Y = 3 N = 3	6	$3/6 = \frac{1}{2}$	$3/6 = \frac{1}{2}$

$$E(\text{Weak}) = -\frac{3}{4}\log\left(\frac{3}{4}\right) - \frac{1}{4}\log\left(\frac{1}{4}\right)$$
$$= -\frac{3}{4}\log(-0.42) - \frac{1}{4}\log(-2)$$
$$= 0.32 + 0.5$$
$$= 0.82$$

$$E(\text{Strong}) = -\frac{1}{2}\log\left(\frac{1}{2}\right) - \frac{1}{2}\log\left(\frac{1}{2}\right)$$
$$= \frac{1}{2} + \frac{1}{2} = 1$$

$$E(\text{Wind}) = \frac{8}{14} \times 0.82 + \frac{6}{14} \times 1$$
$$= 0.47 + 0.43$$
$$= 0.90$$

$$IG = 0.95 - 0.90$$
$$= 0.05$$

- As outlook feature has maximum Information Gain, we will consider it as Root Node.

Gini Impurity

$$G = 1 - \sum_{i=1}^{c} (P_i)^2$$

$$G(i) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2$$

$$= 1 - \frac{81}{196} - \frac{25}{196}$$

$$= \frac{90}{196} = 0.46$$

Outlook

$G(\text{sunny}) = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2 = 1 - \frac{9}{25} - \frac{4}{25} = \frac{12}{25} = 0.48$

$G(\text{overcast}) = 1 - \left(\frac{4}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0$

$G(\text{rainfall}) = 1 - \left(\frac{2}{5}\right)^2 - \left(\frac{3}{5}\right)^2 = 0.48$

$$G(\text{class}) = \sum \frac{\text{No. of observations}}{\text{Total no. of observations}} \times G_i$$

$$G(\text{outlook}) = \frac{5}{14} \times 0.48 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.48$$

$$= 0.17 + 0 + 0.17$$

$$= 0.34$$

$$IG = 0.48 - 0.34$$

$$= 0.121$$

Temperature

$$G(\text{hot}) = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2$$

$$= 1 - \frac{1}{4} - \frac{1}{4} = \frac{1}{2} = 0.5$$

$$G(\text{mild}) = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2$$

$$= 1 - \frac{4}{9} - \frac{1}{9} = \frac{4}{9} = 0.44$$

$$G(\text{cool}) = 1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2$$

$$= 1 - \frac{9}{16} - \frac{1}{16} = \frac{6}{16} = 0.375$$

$$G(\text{temperature}) = \frac{4}{14} \times 0.5 + \frac{6}{14} \times 0.44 + \frac{4}{14} \times 0.375$$

$$= 0.14 + 0.19 + 0.11$$

$$= 0.43$$

$$I.G = 0.48 - 0.43 = 0.05$$

Humedaty

$G(High) = 1-(\frac{3}{7})^2-(\frac{4}{7})^2$

$= 1-\frac{9}{49}-\frac{16}{49}$

$= \frac{24}{49} = 0.48$

$G(normal) = 1-(\frac{6}{7})^2-(\frac{1}{7})^2$

$= 1-\frac{36}{49}-\frac{1}{49}$

$= \frac{12}{49} = 0.24$

$G(Humedaty) = \frac{7}{14} \times 0.48 + \frac{7}{14} \times 0.24$

$= 0.24 + 0.12 = 0.36$

$IG = 0.48 - 0.36$

$= 0.12$

Wind

$G(weak) = 1-(\frac{6}{8})^2-(\frac{2}{8})^2$

$= 1-\frac{36}{64}-\frac{4}{64}$

$= \frac{64-36-4}{64} = \frac{24}{64} = 0.375$

$G(strong) = 1-(\frac{1}{2})^2-(\frac{1}{2})^2$

$= 1-\frac{1}{4}-\frac{1}{4} = \frac{2}{4} = 0.5$

$G(wind) = \frac{8}{14} \times 0.375 + \frac{6}{14} \times 0.5$

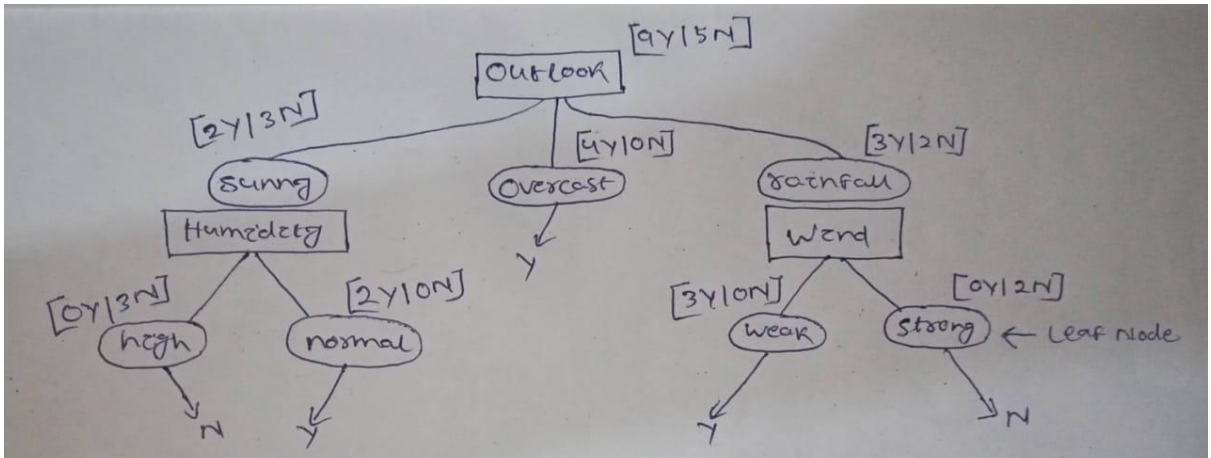
$= 0.21 + 0.21$

$= 0.42.$

$IG = 0.48 - 0.42$

$= 0.06$

- With respect to the calculation of Gini Impurity impurity and max information gain 'outlook' has least value. So, we will consider it as root node



- We won't split more, as we already got conclusion. Otherwise our model will be overfit.

Decision Tree Regressor :

Decision Tree Regressor is when the prediction is a continuous value. If dataset is non-linear or messy then also we use Decision Tree.

We use CART (Classification and Regression Trees.) Algorithm for Decision Tree Regressor.

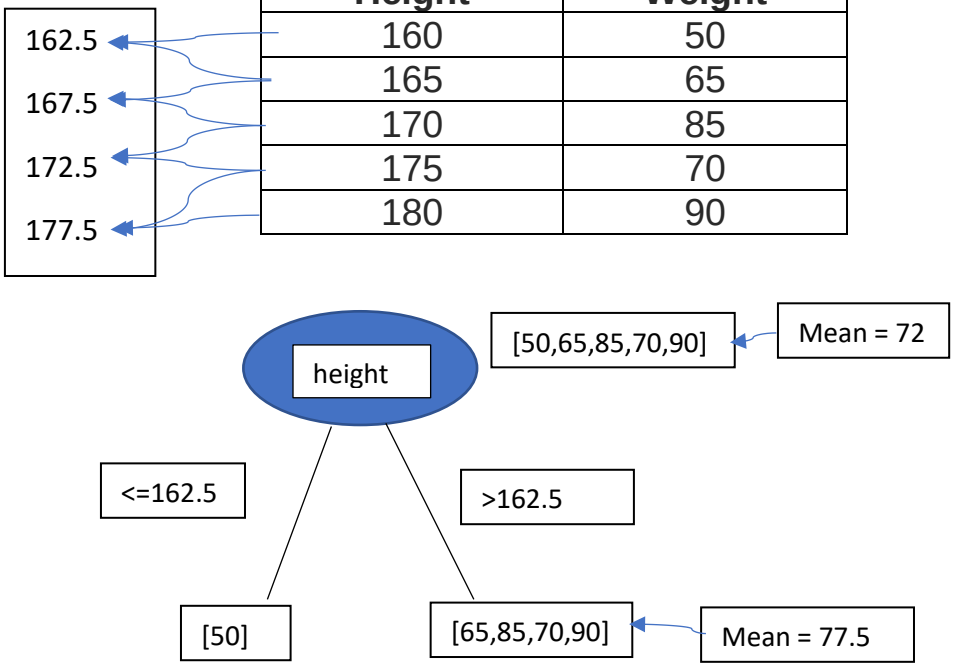
CART tries to take features and label and do calculation. It tries to find out ŷ value after dividing the data based on threshold and how it is different from y by calculating error. Ŷ will be calculated based on the average (mean) value of all the outcomes in a particular branch.

For regression problem we need to follow the following steps :

1. Sort the values
2. Average adj. value (Mean of the each pair values of root node)
3. MSE/MAE/RMSE
4. Reduction in Variation

Eg :

Height	Weight
160	50
165	65
170	85
175	70
180	90



$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Height[variance] = $\frac{(72-50)^2 + (72-65)^2 + (72-85)^2 + (72-70)^2 + (72-90)^2}{5}$ = 206

Var[Right] = $\frac{(77.5-65)^2 + (77.5-85)^2 + (77.5-70)^2 + (77.5-90)^2}{5}$ = 106.25

Var[left] = 50

Reduction in Variance (RV) = Var[Root] - $\sum_{i=1}^{n} w_i * var[child]$

= 206 – [$\frac{1}{5} * 0 + \frac{4}{5} * 106.25$]

= 121

In the same process we will calculate RV for each feature.

We calculate residual for every node, the feature with lowest MSE will be selected as Root Node. For splitting we need to select correct threshold with less impurity. As the MSE will be reduced, means we are more towards leaf node. And threshold will be selected based on MSE lower MSE value.

The disadvantage with Decision Tree Regressor is that it will take more time for training. (Time complexity increases)

Based on weightage of the class we can make conclusion otherwise Decision Tree over fits the data, if you'll plot Decision Tree for every feature/node.

We go for Decision Tree in case of

i. Complex Dataset
ii. If Dataset is not linear
iii. If we want to reduce over fitting
iv. For Multi-class classification

Feature Scaling is not required for Decision Tree as we are splitting the features based on certain conditions and

When to stop splitting?

We discussed above that if we will split the tree for each and every feature at every node that leads to over-fitting. There we need to split based on weightage of the output class. To do this we can use the hyper-

parameters : max_depth, min_samples_split, min_samples_leaf, max_feature, cpp_alpha.

We are going to do pre-pruning and post-pruning to reduce overfitting.

Pruning:

Pruning is a methodology by which performance of a tree can be increased. It is used to overcome the over-fitting and under-fitting condition of the Decision Tree. It removes the branches that use features having low importance. This way, we reduce the complexity of tree, and increases its predictive power by reducing overfitting.

Pre-pruning:

In pre-pruning we try to stop Decision Tree to create unnecessary branches before tree creation. Where we can control the depth of the tree by providing value to the hyper-parameters.

We use the hyper-parameters :

criterion: Function to measure the quality of split,

max_depth: Maximum depth of the tree, if none it'll expand till pure split.

min_samples_split: Min. number of samples req. to split an internal node, default value is 2.

min_samples_leaf: Min. number of samples required at leaf node, default vaue is 1.

max_features: Number of features to consider at the time of best split.

Post-pruning:

In post-pruning we go backward from leaf node and try to check the combinations. The combinations which are not related will get removed by hyper-parameter tuning with cross validation where we pass multiple hyper-parameters to get improvement in the model.

We use the "ccp_alpha" for post-pruning.

Thank you

Follow me in GitHub for more contents :
https://github.com/saisubhasish