Date :— 12th November, 2022

(97)

Machine Learning (Day-9) :-

Agenda :-

1) Pruning
2) Pre Pruning and Post Pruning
3) Decision Tree Algorithm.
4) Hyperparameter for Classification And Regression
5) OverFiting and Under Fitting.

✱) Pruning :-

⇒ Pruning Means Cutting

⇒ Pruning is Done to reduce Overfitting in Decision Tree.

Now,

⇒ Pruning is a Technique that removes the part of the Decision Tree which prevent it from growing to its Full depth. The parts that it removes from the tree are the parts that do not provide the power to Classify instances. A Decision Tree that is trained to its Full depth will highly likely lead to Overfitting the Training data - therefore Pruning is Important.

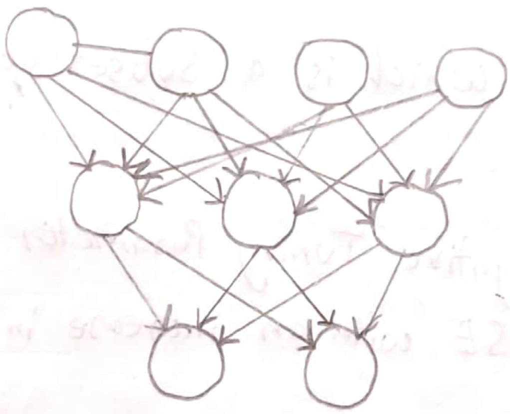
P.T.O.

Note:-

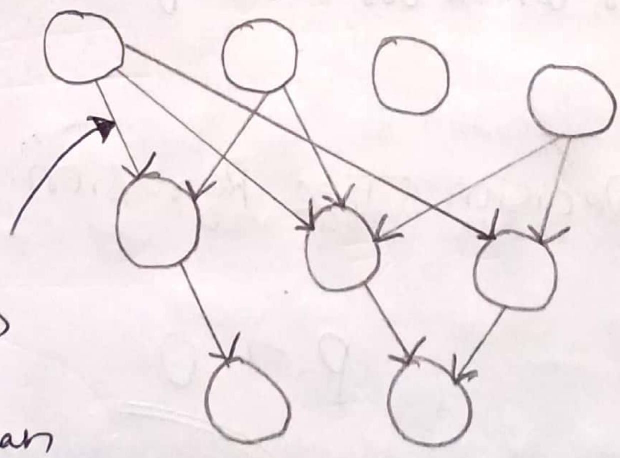
In Simpler Terms,

The aim of Decision Tree Pruning is to contruct an Algorithm that will perform worse on Training data but will generalize better on test data. Tuning the Hyperparameters of your Decision Tree model can do your model a lot of Justice and Save you a lot of Time and Money.

Visualisation Before Pruning And After Pruning



Before Pruning



Remove Synapses

* Also we can remove neuron

After Pruning.

Note :-

Formula :-

Non-overlapping Regions.

$$\boxed{\sum_{m=1}^{|T|} \sum_{i: x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|}$$

Sub Tree

Non-Negative Tuning Parameter

Where;

$\hat{y}_{R_m}$ → Mean of all the response Variable in the region 'm'.

T → SubTree which is a Subset of the Full Tree T_0.

α → Non-Negative Tuning Parameter which Penalises the MSE with an increase in Tree Length.

By using Cross-Validation such values of α and T are selected for which our Model gives the lowest test Error rate.

This is How the Decision Tree Regression Model Works.

P.T.O

*) Post Pruning :— Also known as Backward Pruning.

⇒ IT's the Process where the Decision Tree is Generated First and then the Non - Significant branches are removed.

⇒ Cross Validation set of data is used to check the effect of Pruning and test whether expanding a node will make an improvement or Not.

⇒ If any Improvement is there then we continue by expanding that node else if there is Reduction in Accuracy then the node not be expanded and Should be converted into leaf node.

⇒ This Technique is used when Decision Tree will have very large depth and will Show overfitting of Model.

✩ Pre Pruning :— Also Known as Forward Pruning.

⇒ It Stops the non Significant branches from Generating.

⇒ It uses a condition to decide when should it Terminate Splitting of some of the branches ~~Permat~~ Prematurely as the Tree is Generated.

⇒ Can be done using Hyper Parameter Tunning.

⇒ overcome the overfitting Issue.

*) Different Algorithm For Decision Tree :-

① ID3 : (Iterative Dishotomiser) :-

- ID3 used to construct Decision Tree for Classification
- It used Information Gain as the Criteria for finding the Root Nodes and Splitting them.
- It only accepts categorical Attributes.

② C4-5 :-

- It is an extension of ID3 Algorithm, and better than ID3 as it deals ~~both~~ Both Continuous and Discrete Value.
- It is also used for Classification Purpose.

③ CART : (Classification and Regression Algorithm) :-

⇒ It uses Gini Impurity as the Default Calculation for Selecting Root Nodes However one can use "Entropy" for Criteria as well.

- It works on both Regression as well as Classification Problems.

⇒ Entropy and Gini - Impurity can be used Reversibly
It doesn't affects the Result much.

Although, Gini is easier to compute than Entropy.
Since, Entropy has long term calculation.

That's why CART Algorithm uses Gini as the Default Algorithm.

(4) CAID: (Chi - Square Automatic Interaction Detection).

⇒ It finds out the Statistical Significance between the different between Sub-nodes and Parent Nodes.

⇒ We measure it by the Sum of Squares of Standardized Different Between Observed and Expected Frequencies of the Target Variable.

⇒ It works with Categorical Target Variable.

P.T.O

#) Question:) How to Decide a Threshold?

⇒ We look After the Custom Mechanism that is being Designed, internally in all these Algorithms.

ID3, C4.5, CART, CAID

⇒ Internally, we find out that Sometimes we take average and based on that we try to Divide or make Different bins / blocks.

These are Some ways to create threshold.

$$\boxed{\text{Information Gain} \propto \frac{1}{\text{Gini Impurity}}}$$

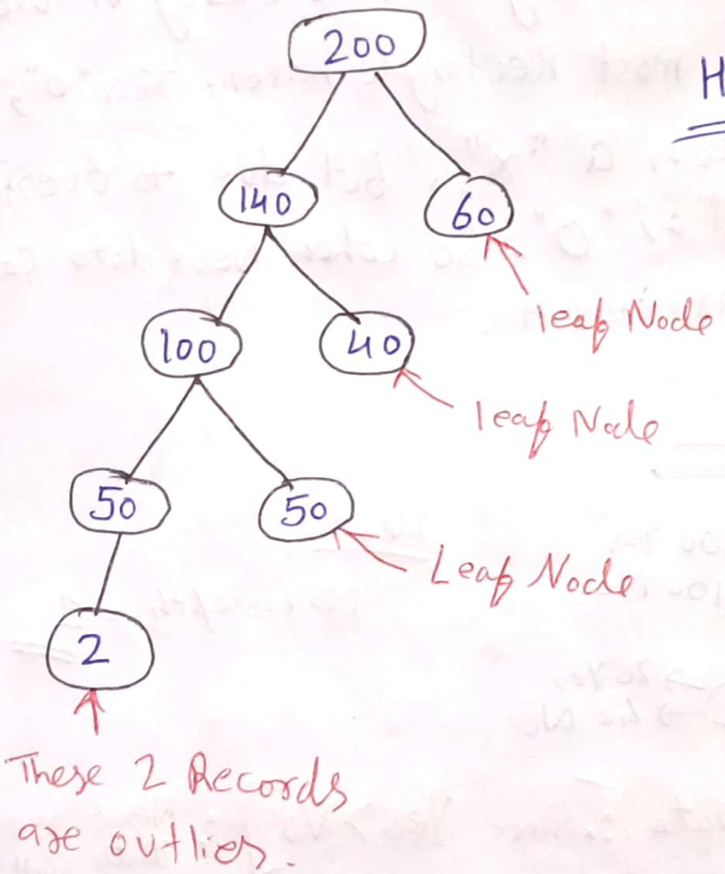
P.T.O

*) Overfitting And Underfitting :-

Note :- Max_depth or depth of tree is 'Responsible for over and Underfitting'.

*) Overfitting :
→ Performs well for Training Data
→ Performs Badly for Test Data.

Consider :- 200 Datapoint :-



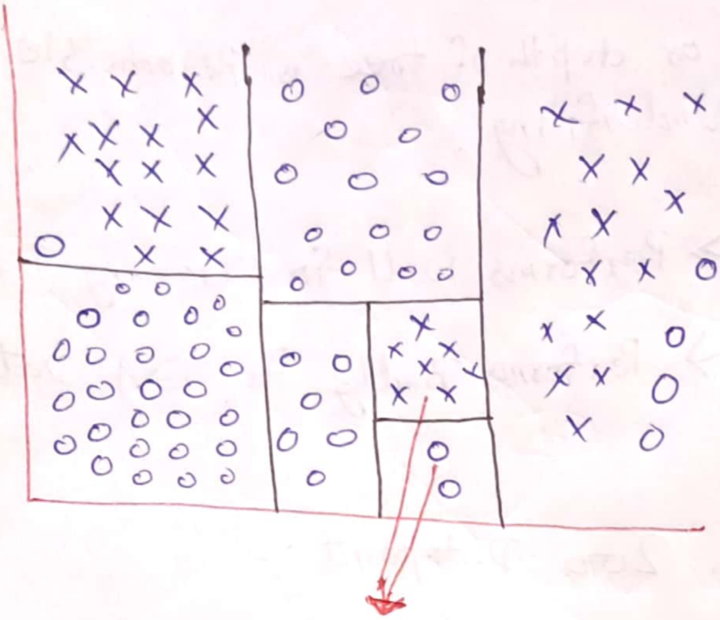
Here :-

Max_depth = None

Here,

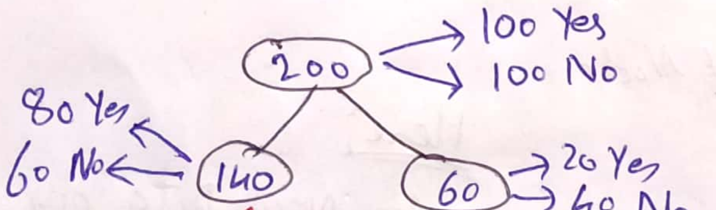
For New Data our model will not Perform good as Whole Decision making is Dependent on outlier Records.

*) Geometric Intuition of OverFittings :—



This Represents overfitting as it is Clearly Visible that the bottom most Rectangle having Two "O", Should have ~~been~~ a "x", but due to overfitting it is Classified as "O". So when New data Comes it wrongly Classifies it.

*) UnderFitting :—

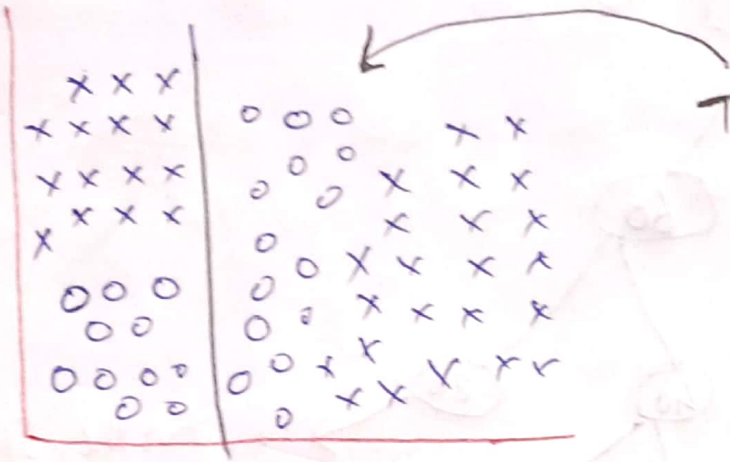


Here,

Max-depth = 1

New Query Data [∴ Since Yes > No] ∴ New Query Data will be labeled Yes.

P.T.O



This is a Case of Underfitting

A) Hyper-Parameters (From SK-Learn Implementation of DT)

1) Generally Gini gives better Results.

2) Splitter :-

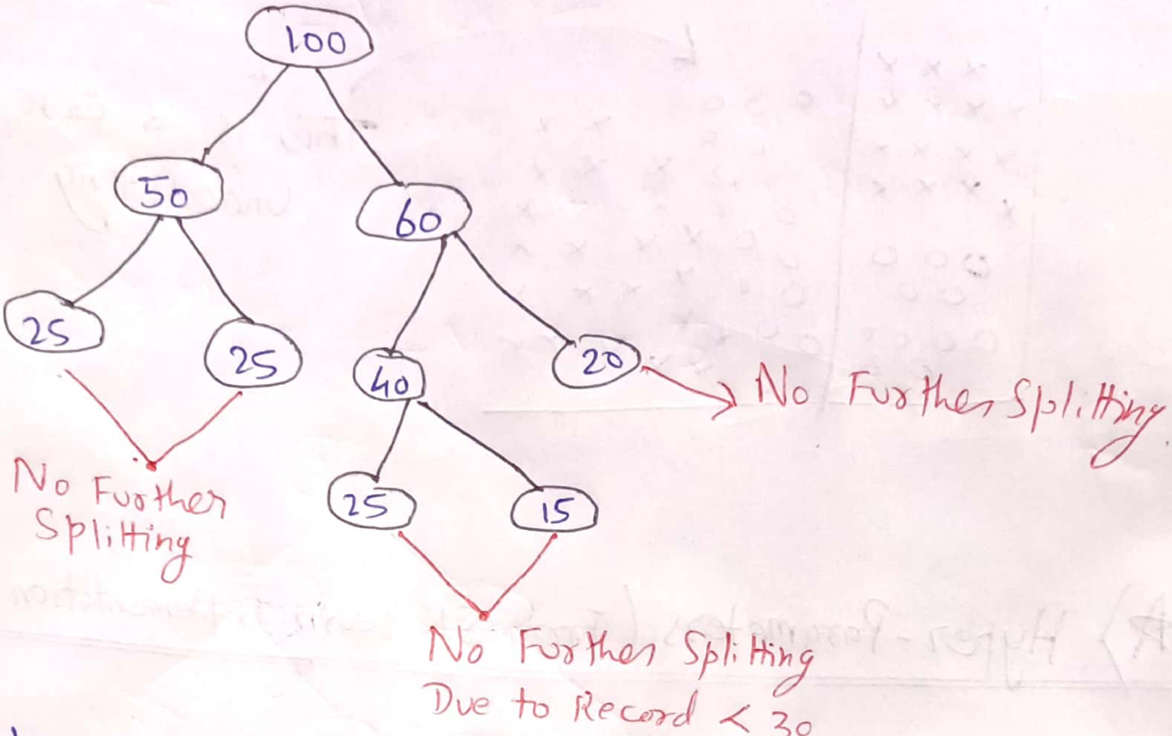
For Numerical Feature — Random (To Reduce over-fittings)
— Best (Considers all Splits which leads to overfit)

3) Max-depth → None (Cause over Fitting)
→ Very Less (Cause Under Fitting)

4) Min_Sample_Split :-

This is restriction on Splitting.

Suppot, we take its value as 30, So at those nodes Where the value of Records will be less than 30, No Further Split will take place for those Nodes and those nodes will Beome Leaf Nodes.



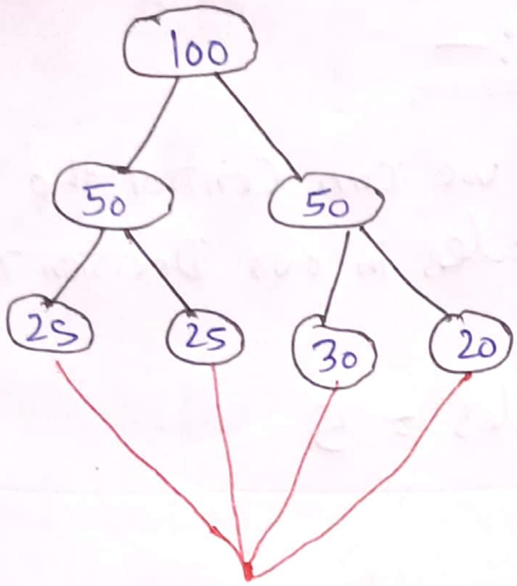
- Higher min_sample_split value may cause Underfitting and with Lower min_sample_split value there is more chance of overfitting.

5) Min_Sample_leaf :-

This also put Restriction on Splitting.

Support we have 100 Data points with Min_Sample_Leaf Value as 20. This means a node will not split, if it will create a Left node with less than 20 records.

P.T.O



These Nodes cannot split Further because as They will Split Further they will atleast. Create a Left Node having less than 20 Records.

⇒ Higher Min_Sample_leaf Value will Cause Underfitting Whereas with Lower value there is chance of over fitting.

6) Max_Features :—

⇒ Support we are Working with High Features problem which is leading to over Fitting, So in that case we can restrict No. of Features available at Node For deciding the best Feature for Splitting using Max_Features.

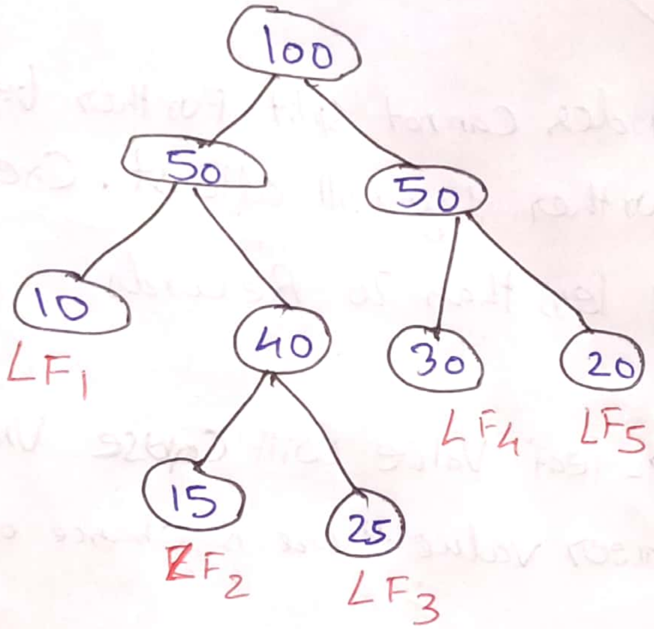
The Feature Selection is random and this is done to reduce overfitting.

P.T.O

(7) Max_leaf_Nodes :-

⇒ With this Feature we can Control the Number of Leaf nodes in our Decision Tree.

Say Max_leaf_Nodes = 5



Note :-

Higher value ⇒ Over Fitting
Lower Value ⇒ Under Fitting.

(8) Min_Impurity_Decrease :-

⇒ our Aim is to increase Information Gain and reduce Entropy / Gini Impurity.

⇒ This Parameter insured that the minimum Value decrease in impurity will only allow Splitting.

P.T.O

⇒ Say Min_Impurity_Decrease = 0.1

So, if Splitting will Decrease impurity by 0.1 or more then only this Splitting will be allowed.

∴ Higher Value ⇒ Underfitting
lower Value ⇒ over Fitting.

⭐) Hyper-Parameter Tuning For Regression Tree :—

1) Criterion :— MSE, MAE, Friedmam MSE

2) Splitter :— Best, Random (To reduce overfitting).

3) Max_depth :— How much we want our Tree to grow.

Higher Value :— Under fitting
Lower Value :— Over fitting.

4) Min_Sample_Split :— Min Sample to do Split

higer value :— Under Fitting
lower value :— Over Fitting.

5) Min_Sample_~~Leaf~~ Leaf ⇒ Min Samples in leaf for Splitting.
Lesser value ⇒ over fitting, [Higher Value = Underfitting]

6) Max_leaf_Node ⇒ Max leaf Node to be there in DT.
Lesser value = Under fitting and Higher value = overfitting

7) Min_Impurity_Decrease ⇒ Min Decrease in impurity to do Splitting
Less value = Over fitting and Higher ⇒ Underfitting.