INDIAN INSTITUTE OF TECHNOLOGY MADRAS

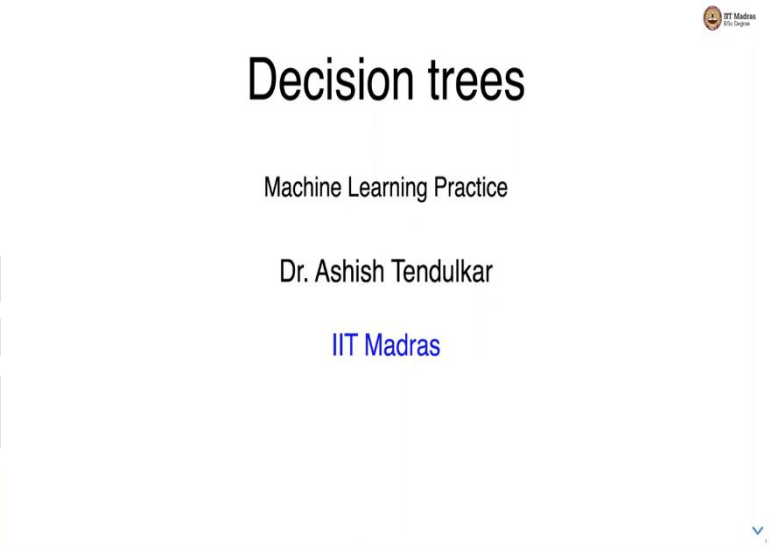
सिद्धिर्भवति कर्मजा

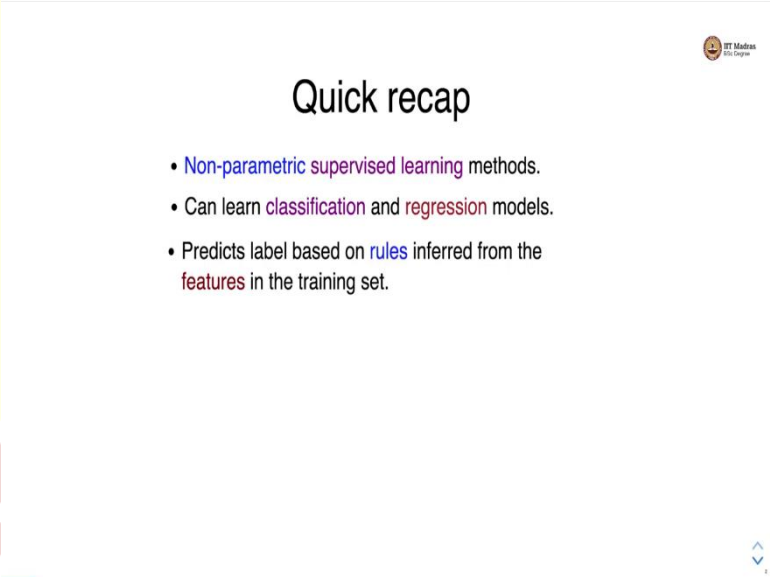
# IIT Madras

## ONLINE DEGREE

**Machine Learning Practice**
**Professor Dr. Ashish Tendulkar**
**Indian Institute of Technology, Madras**
**Decision Trees**

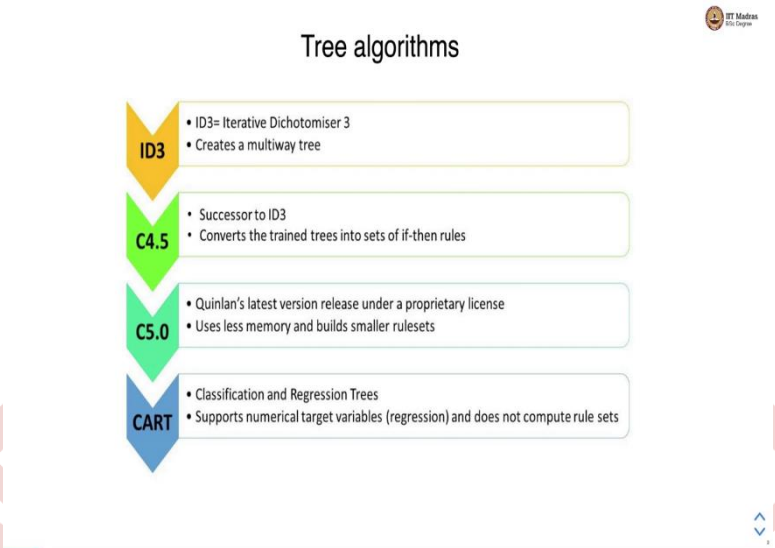
**(Refer Slide Time: 0:11)**



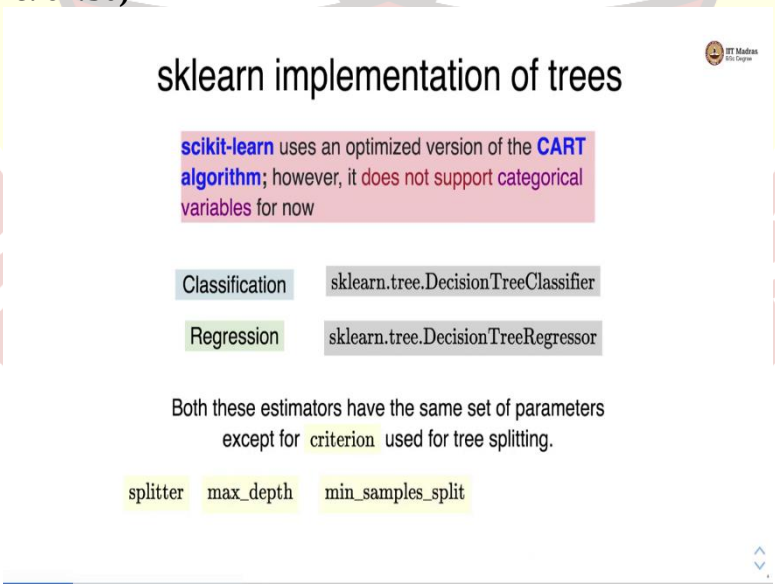

Namaste! welcome to the next video of Machine Learning Practice Course. In this video, we look at decision trees as implemented in sklearn. Let us take a quick recap of what decision tree does? Decision tree is a non-parametric supervised learning method, it can learn both classification as well as regression models. Decision Tree predicts label based on rules inferred from the features in the training set.

**(Refer Slide Time: 0:46)**




There are several tree algorithms, ID3 is the basic tree algorithm. It stands for Iterative Dichotomiser and it can create multiway trees, it is succeeded by C4.5 that converts the trained trees into set of if-then rules. That once version of C4.5 is called as C5.0 that uses less memory and builds smaller rule sets. And then there is a CART algorithm which is Classification Regression Trees, and it supports regression besides classification, as done in the traditional tree algorithm. CART however, does not compute rule sets.

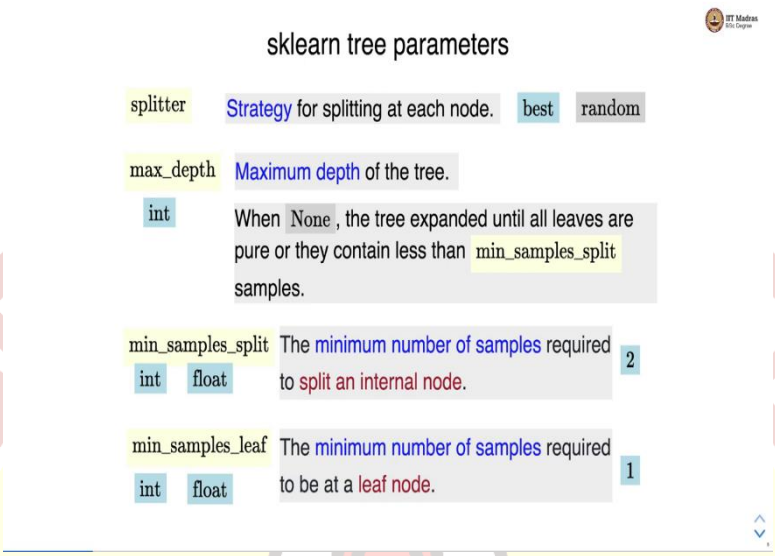
**(Refer Slide Time: 01:30)**




Let us look at sklearn implementation of trees. Sklearn uses an optimised version of the CART algorithm. However, it does not support categorical variables for now. For classification, sklearn.tree.DecisionTreeClassifier API is used. While for regression, we use sklearn.tree.DecisionTreeRegressor. Both these estimators have the same set of parameters

except for the criterion used for tree splitting. There are parameters like splitter, max _depth, min _samples _split and min _samples _leaf.

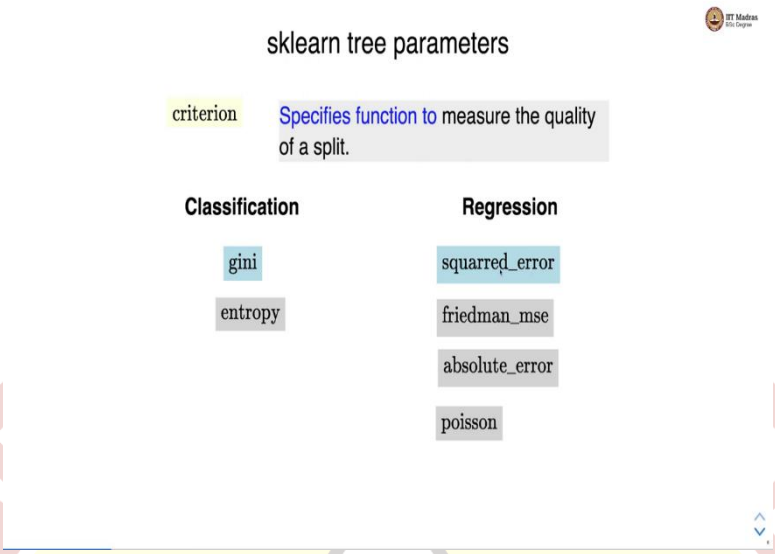
**(Refer Slide Time: 02:09)**




Let us look at what these parameters actually mean. The splitter specifies strategy for splitting at each node. It has two values best and random, best is the default value used for splitter strategy. Then there is max _depth, which specifies the maximum depth of the tree, when None is specified by default, and None is the default setting for max _depth, the tree is expanded until all leaves are pure or they contain less than min _samples _split samples. And max _depth, we can also specify an integer value in max _depth.

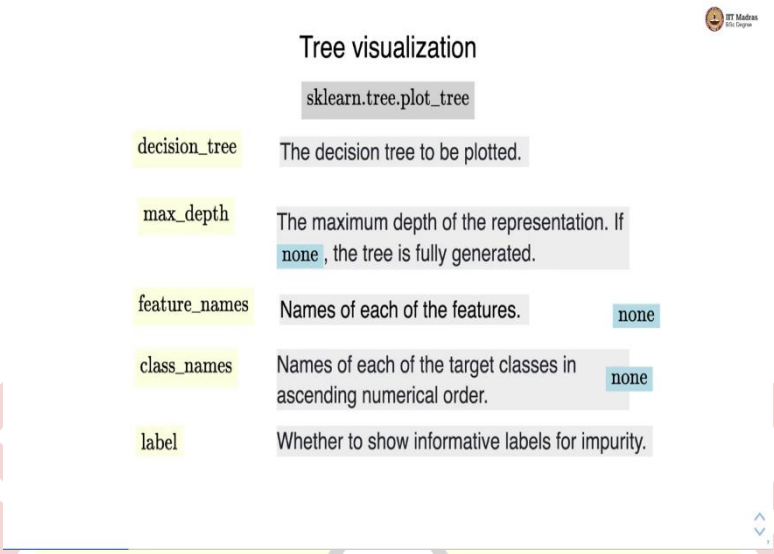
Then there is min sample split that specifies the minimum number of samples required to split an internal node, this is again an integer quantity. We can also alternatively specify a float number over here. By default, the min _samples _split is set to 2. Then there is min _samples _leaf that specifies the minimum number of samples required to be at a leaf node. And by default, the value of min _samples _leaf is 1, it is again just like min sample split, it can take both integer and float values.

**(Refer Slide Time: 03:32)**




Let us look at criterion which is different for classification and regression. So criterion specifies function to measure the quality of a split. Let us look at criterions for classification and regression task. In classification, we use gini and entropy, while for regression we have a squared _error, friedman _mse, absolute _error and poisson as criterions. For classification, gini is the default choice, whereas for regression squared error is the default criterion.

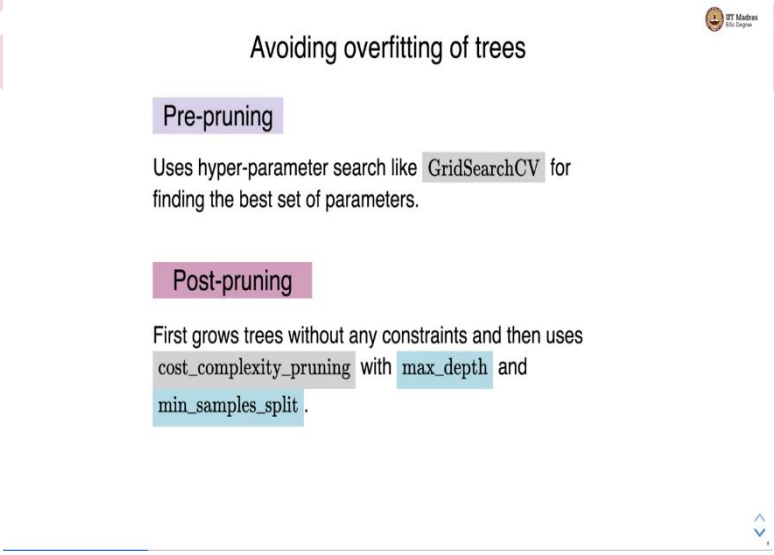
**(Refer Slide Time: 04:08)**




Up to the trees trained, we can visualize it with sklearn.tree.plot _tree utility. It takes the following parameters, the decision tree estimator or the decision tree to be plotted which is derived from the estimator, then the max _depth of the representation, if the max _depth is none, which is the default value, the full tree is shown in the plot tree utility.

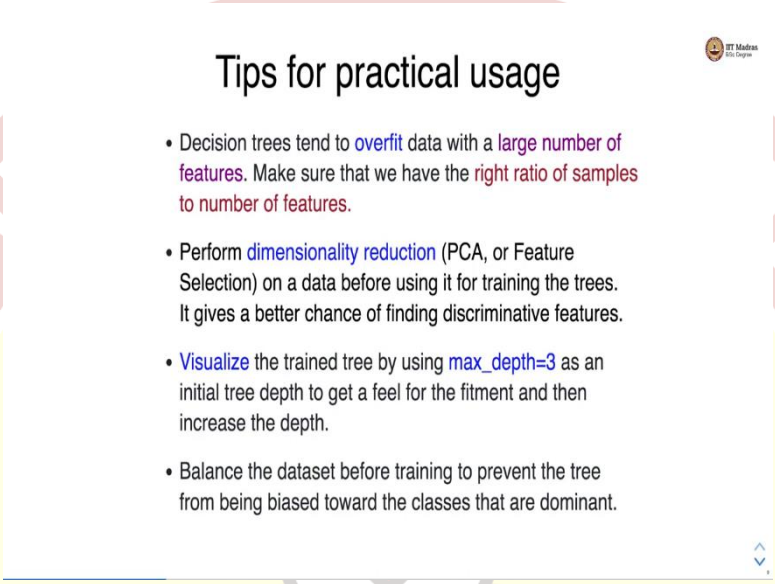
Then you have to specify names of each features, in features _names, argument, and by default, it takes none as the value. Then there are class _names, where we specify names of each of the target classes in ascending numerical order. And by default, this is also none and finally, the label where the label we specify whether to show informative labels for impurity, by default label is also none.

**(Refer Slide Time: 05:14)**



Let us see how to avoid overfitting of trees, overfitting is one of the issues that tree models can face. There are two strategies for avoiding overfitting of trees. One is pre-pruning and second is post-pruning. In pre-pruning, we use hyper-parameter search like GridSearchCV for finding the best set of parameters, post-pruning first grows trees without any constraints, and then uses cost _complexity _pruning with max _depth and min _samples _split parameters.

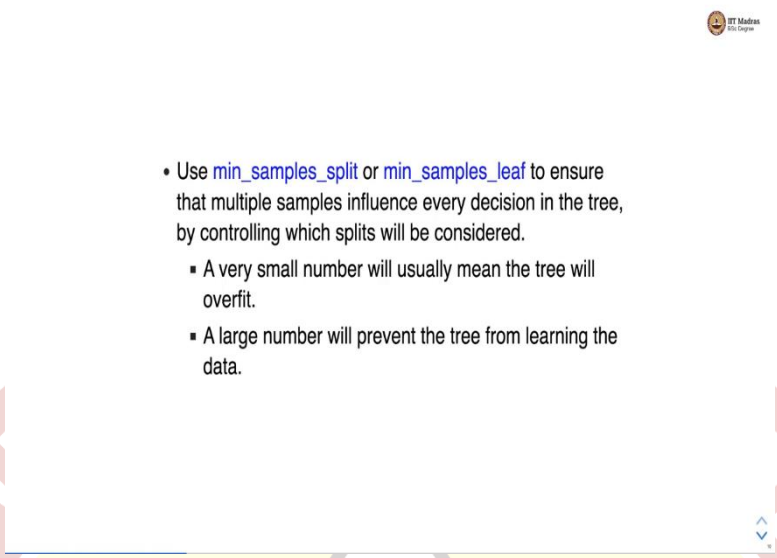
**(Refer Slide Time: 05:40)**




Let us look at some of the tips for practical usage. Decision trees tend to overfit data with a large number of features, make sure that we have the right ratio of samples to the number of features. It is a good idea to perform dimensionality reduction using techniques like PCA, or feature selection. The dimensionality reduction on the data before using it for training the trees gives a better chance of finding discriminative features.

Visualize a trained tree by using max _depth = 3 as an initial tree depth to get a feel for the fitment and then increase the depth. By doing this you ensure that you have a tree that is, that makes sense for the data on which you are training. It is a good idea to balance the dataset before training to prevent a tree from being biased towards the dominant classes.

**(Refer Slide Time: 06:51)**




Finally, use min _samples _split or min _samples _leaf to ensure that multiple samples influence every decision in the tree by controlling which splits will be considered. A very small value over here will usually mean that the tree will overfit. On the other hand if you specify a large number that will prevent a tree from learning from the data.

So in this video, we talked about different scale and utilities for implementing decision trees. We also looked at how to visualize a decision tree with plot _tree utility. We also looked at how to avoid the overfitting of the tree. We will use all these utilities in the practical setup in the following collapse.