**DECISION TREE CLASSIFIER API**

**Abhishek Mukundan Iyer**

**J021**

• class sklearn.tree.DecisionTreeClassifier(\*, criterion='gini', splitter='best',

max\_depth=None, min\_samples\_split=2, min\_samples\_leaf=1, min\_wei

ght\_fraction\_leaf=0.0, max\_features=None, random\_state=None, max\_le

af\_nodes=None, min\_impurity\_decrease=0.0, min\_impurity\_split=None,

class\_weight=None, ccp\_alpha=0.0)

• **Parameters –**

1. criterion: {“gini”, “entropy”}, default=”gini”

2. splitter: {“best”, “random”}, default=”best”

3. max\_depth: int, default=None

4. min\_samples\_split: int or float, default=2

5. min\_samples\_leaf: int or float, default=1

6. min\_weight\_fraction\_leaf: float, default=0.0

7. max\_features: int, float or {“auto”, “sqrt”, “log2”}, default=None

8. random\_state: int, RandomState instance or None, default=None

9. max\_leaf\_nodes: int, default=None

10.min\_impurity\_decrease: float, default=0.0

11.min\_impurity\_split: float, default=0

12.class\_weight: dict, list of dict or “balanced”, default=None

13.ccp\_alpha: non-negative float, default=0.0

• **Attributes -**

1. classes\_ndarray of shape (n\_classes,) or list of ndarray

2. feature\_importances\_ndarray of shape (n\_features,)

3. max\_features\_int

4. n\_classes\_int or list of int

5. n\_features\_int

6. n\_outputs\_int

7. tree\_Tree instance

• **Advantages: -**

1.Simple to understand and to interpret. Trees can be visualised.

2.Requires little data preparation.

3. The cost of using the tree (i.e., predicting data) is logarithmic in the

number of data points used to train the tree.

4. Able to handle both numerical and categorical data.

5. Able to handle multi-output problems.

6. Possible to validate a model using statistical tests. That makes it

possible to account for the reliability of the model.

7. Performs well even if its assumptions are somewhat violated by the

true model from which the data were generated.

• **Disadvantages:**

1. Decision-tree learners can create over-complex trees that do not

generalise the data well. This is called overfitting.

2. Decision trees can be unstable because small variations in the data

might result in a completely different tree being generated. This

problem is mitigated by using decision trees within an ensemble.

3. Predictions of decision trees are neither smooth nor continuous, but

piecewise constant approximations as seen in the above figure.

Therefore, they are not good at extrapolation.

4. Decision tree learners create biased trees if some classes dominate. It

is therefore recommended to balance the dataset prior to fitting with

the decision tree.