Soham Shah
J059

**API:**

*class* sklearn.tree.**DecisionTreeClassifier**(*, *criterion='gini'*, *splitter='best'*, *max_depth=None*, *min_samples_split=2*, *min_samples_leaf=1*, *min_weight_fraction_leaf=0.0*, *max_features=None*, *random_state=None*, *max_leaf_nodes=None*, *min_impurity_decrease=0.0*, *min_impurity_split=None*, *class_weight=None*, *ccp_alpha=0.0*)

**Important parameters:**

splitter – determines how the decision tree searches the features for a split. Default value is set to 'best' but could be changed to 'random'.

max_depth – will determine the maximum depth of a tree. The default value is none but this should be regularised to prevent overfitting.

min_samples_split – minimum number of samples a node must contain to consider splitting

min_sample_leaf- minimum number of samples needed to be considered a lead node. The default value is set to one.

Max_features- number of features to consider when looking for the best split. By default the decision tree will consider all the values available to make the best split.

**Decision tree working:**

Decision Trees are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

Advantages:

- Requires little data preparation and simple to understand
- Able to handle both- numerical and categorical data
- Able to handle multiple output problems
- Possible to validate it using statistical tests

Disadvantages:

- Decision tree learners could create over-complicated trees that do not generalize the data well and lead to overfitting.
- Decision trees could be quite unstable
- Predictions of decision trees are neither smooth or continuous but a piecewise constant approximation. Therefore, not providing a good extrapolation.
- It could lead to biased trees if some classes end up dominating.