Pratyush Patro
J047

DECISION TREE CLASSIFIER 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)

• 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.