Hyperparameters

Monday 20 May 2024 11:28 AM

Splitter
- Best → for each feature select the best split-value → select the best out of the best
- Random → for each feature select random split_value → select the best out of the random splits
  - ↳ Randomness + speed

max_depth

min_samples_split : minimum # of samples required to split an internal node

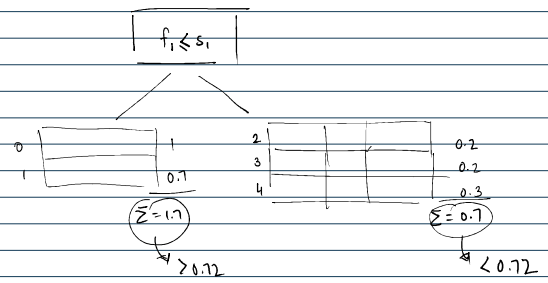
min_samples_leaf : The minimum # of samples required to be at a leaf node.
A split will occur only if it leaves at least min_samples_leaf training samples in each right & left branches.

min_weight_fraction_leaf :

	f_1	f_2	target	weight
0				1
1				0.7
2				0.2
3				0.2
4				0.3

$\Sigma = 2.4$

Suppose min_weight_fraction_leaf = 0.3 → $0.3 \times 2.5 = 0.72$



Split NOT allowed

max_features: chosen at each split [int, float, sqrt, log2, None]

Note: the search for a split does not stop until at least one valid partition of the node samples is found, even if it requires to effectively inspect more than `max_features` features.

random_state : → deterministic behaviour

max_leaf_nodes : Grow a tree with `max_leaf_nodes` in best-first fashion. Best nodes are defined as relative reduction in impurity. If None then unlimited number of leaf nodes.

Current state:

Assume max_leaf_nodes = 4

Choose the one that is giving more decrease in impurity

min_impurity_decrease :

A node will be split if this split induces a decrease of the impurity greater than or equal to this value.

The weighted impurity decrease equation is the following:

```
N_t / N * (impurity - N_t_R / N_t * right_impurity
                    - N_t_L / N_t * left_impurity)
```

where `N` is the total number of samples, `N_t` is the number of samples at the current node, `N_t_L` is the number of samples in the left child, and `N_t_R` is the number of samples in the right child.

`N`, `N_t`, `N_t_R` and `N_t_L` all refer to the weighted sum, if `sample_weight` is passed.

class_weight:

"balanced"

0
0
1
0
0
0
0
1
1
1

class_weight: $\frac{n\_samples}{n\_classes * np.bincount(y)} = \frac{10}{2 * [6,4]}$

$= \frac{10}{[12,8]}$

$= \left\{0, \frac{10}{12}, 1, \frac{10}{8}\right\}$

$\frac{10}{12} \times 6 = 5 \quad \frac{10}{8} \times 4 = 5$

$= \{0, \frac{10}{12}, 1, \frac{10}{8}\}$

$\frac{10}{12} \times 6 = 5 \quad \frac{10}{8} \times 4 = 5$

For multi-output, the weights of each column of y will be multiplied.

Note that these weights will be multiplied with sample_weight (passed through the fit method) if sample_weight is specified.