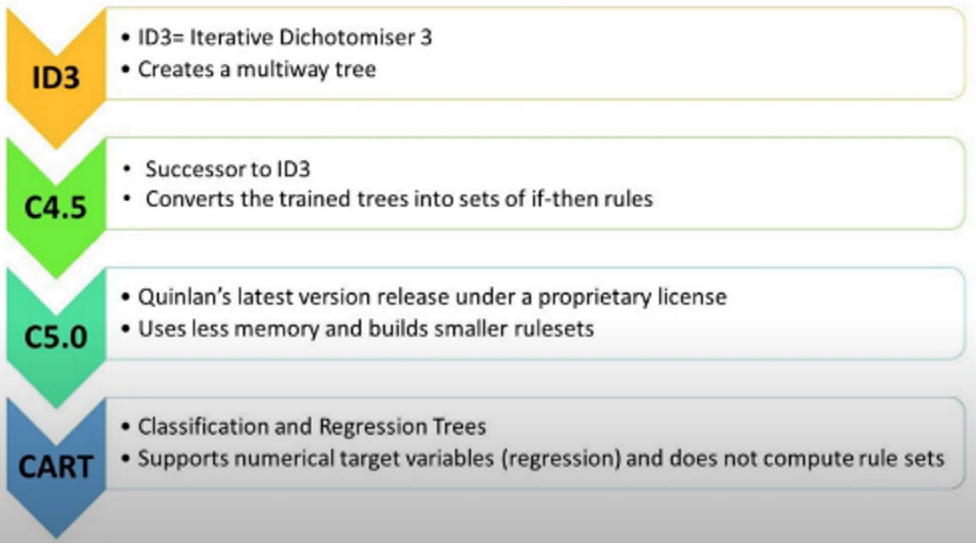
Decision Tree

23 March 2023 06:18 PM

Quick recap

- Non-parametric supervised learning methods.
- Can learn classification and regression models.
- Predicts label based on rules inferred from the features in the training set.

Tree algorithms



sklearn implementation of trees

scikit-learn uses an optimized version of the **CART algorithm**; however, it does not support categorical variables for now

Classification `sklearn.tree.DecisionTreeClassifier`

Regression `sklearn.tree.DecisionTreeRegressor`

Both these estimators have the same set of parameters except for `criterion` used for tree splitting.

`splitter` `max_depth` `min_samples_split`

sklearn tree parameters

`splitter` Strategy for splitting at each node. `best` `random`

`max_depth` Maximum depth of the tree.

`int`

When `None`, the tree expanded until all leaves are pure or they contain less than `min_samples_split` samples.

`min_samples_split` The minimum number of samples required to split an internal node. `2`

`int` `float`

`min_samples_leaf` The minimum number of samples required to be at a leaf node. `1`

sklearn tree parameters

criterion — Specifies function to measure the quality of a split.

Classification	Regression
gini	squarred_error
entropy	friedman_mse
	absolute_error
	poisson

Tree visualization

sklearn.tree.plot_tree

decision_tree	The decision tree to be plotted.	
max_depth	The maximum depth of the representation. If none, the tree is fully generated.	
feature_names	Names of each of the features.	none
class_names	Names of each of the target classes in ascending numerical order.	none
label	Whether to show informative labels for impurity.	none

Avoiding overfitting of trees

Pre-pruning

Uses 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`.

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 number of features.
- Perform dimensionality reduction (PCA, or Feature Selection) on a data before using it for training the trees. It gives a better chance of finding discriminative features.
- Visualize the trained tree by using max_depth=3 as an initial tree depth to get a feel for the fitment and then increase the depth.
- Balance the dataset before training to prevent the tree from being biased toward the classes that are dominant.

- 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 number will usually mean the tree will overfit.
  - A large number will prevent the tree from learning the data.