Decision Trees

Decision trees are versatile machine learning algorithms that can perform both classification and regression, and fundamental for random forests.

Making Predictions

- The tree starts with the entire dataset and splits it based on feature values that best separate the target classes.
- Each internal node represents a decision based on a feature, while leaf nodes represent the final prediction or output class.
- Splitting criteria include Gini impurity, information gain, or mean squared error.
- The splitting process is recursive and continues until certain stopping conditions are met (e.g., maximum depth or minimum samples).
- To make a prediction, input features are passed through the tree, making decisions at each node based on feature values.
- The process continues until reaching a leaf node, which provides the final prediction.
- Decision trees can handle both continuous and categorical data through appropriate splitting techniques.
- Techniques like pruning and setting maximum depth help prevent overfitting.

A node sample attribute counts how many training instances it applies to.

A node's gini attribute measures how pure a node is (gini=0 == pure). (By default it uses gini index).

Sklearn uses CART algorithm that produces only binary-trees. (ID3 can produce more).

DT's are intuitive and their decisions are easy to predict such are called white box models. RF and NN are black box models. As they can make computations but is usually hard to explain why they such such a prediction.

A DT can also estimate the probability that an instance belongs to a particular class under k.

Computational Complexity

- **Training**: O(n *m* log(m))
- **Prediction**: O(d)
- **Traversal**: O(n)

Entropy

Entropy reaches 0 when all molecules are well ordered. It is 0 when all messages are identical . (Produces more balanced trees).

CART Algorithm

- The CART (Classification and Regression Trees) algorithm constructs decision trees for both classification and regression tasks. It aims to predict the target variable by creating a binary tree structure, where each internal node represents a decision based on feature values.
- It uses a threshold in the splitting criteria and that is decided by the pair that produces the purest subset weighted by their size.
- CART starts with the entire dataset and evaluates all possible splits for each feature. It uses different criteria for splitting: for classification tasks, it typically employs Gini impurity or entropy to assess the quality of splits, while for regression tasks, it uses mean squared error (MSE) to minimize prediction error. The algorithm chooses the split that provides the best improvement according to the chosen criterion.
- This process is recursive. At each step, CART selects the best split and creates two child nodes, repeating the evaluation for each subset of data until certain stopping conditions are met. These conditions might include reaching a specified maximum depth, having a minimum number of samples required to make a split, or when all samples in a node belong to the same class.
- Greedy algo and finding a optimal tree is NPC.

Regularization Hyperparameters

If left unconstrained the tree will adapt to itself , and overfit such is called non parametric model. (Unlimited freedom).

- **Max Depth**
- **Min Samples Split**
- **Min Samples Leaf**
- Max Features

For regression it tries to split in a way that minimizes the MSE. They are Prone to overfitting when dealing with regression.

Drawbacks of Decision Trees

- **High Variance**: Decision trees are prone to overfitting, especially when they are deep. They can capture noise and fluctuations in the training data, leading to poor generalization on unseen data.
- **Sensitivity to Axis** : Orthogonal boundaries which makes them sensitive to data's orientation.
- **Bias Toward Axis-Aligned Splits**: Decision trees only make axis-aligned splits (i.e., splits based on one feature at a time), which may not capture complex relationships or interactions between features effectively.