Decision Tree & Random Forest

Jerry Peng

1 Decision Tree

A Decision Tree is a flowchart-like tree structure where an internal node represents a feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome. It's a type of supervised machine learning algorithm that is mostly used for classification problems but can be used for regression as well. The Figure 1 shows a simple decision tree classifier

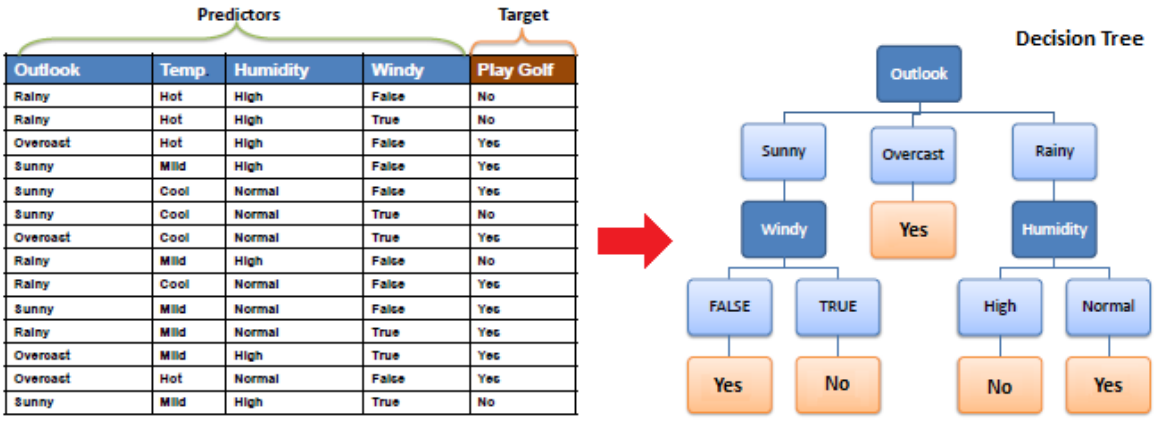

Figure 1: Decision Tree Classifier

1.1 Structure

In a Decision Tree, the topmost node is known as the root node. It works by partitioning the dataset into subsets based on the feature that provides the best separation between the target variable classes. The partitioning process is done recursively, producing a tree with decision nodes and leaf nodes. A decision node has two or more branches, and a leaf node represents a classification or decision. The decision tree makes decisions by asking multiple questions and following the path down the tree that corresponds to the answer.

Selection of Attribute: The attribute that provides the highest **Information Gain** or the lowest **Gini Index** is selected for the decision at the root node.

Splitting: The dataset is then split into subsets based on the chosen attribute's value. This process is done recursively for each child node.

Decision Making: Once the algorithm reaches a point where it cannot further split the data, it makes a decision. This decision is represented by the leaf node.

1.2 Choose Attribute at Each Node

While there are multiple ways to select the best attribute at each node, two methods, information gain and Gini impurity, act as popular splitting criterion for decision tree models. They help to evaluate the quality of each test condition and how well it will be able to classify samples into a class.

Entropy A measure of the impurity or disorder of a set. For a binary classification problem, the entropy H of a set S with p_+ positive samples and p_- negative samples is calculated as:

$$H(S) = -(p_+ \log_2(p_+) + p_- \log_2(p_-)) \tag{1}$$

If it is not binary classification problem and the set S contains C classes ($C > 2$), the equation 1 can be re-written as

$$H(S) = -\sum_{c \in C} p(c) \log_2 p(c) \tag{2}$$

Entropy values can fall between 0 and 1. If all samples in data set, S, belong to one class, then entropy H will equal zero. If half of the samples are classified as one class and the other half are in another class, entropy H will be at its highest at 1. In order to select the best feature to split on and find the optimal decision tree, the attribute with the smallest amount of entropy should be used.

Information Gain The reduction in entropy achieved by partitioning a dataset based on a feature. The attribute with the highest information gain will produce the best split as it's doing the best job at classifying the training data according to its target classification. It is calculated as:

$$\text{Information Gain}(S) = H(S) - \sum_{t \in T} \frac{|S_t|}{|S|} H(S_t) \tag{3}$$

where T is the set of subsets created by splitting S based on a feature, and S_t is one of those subsets.

1.3 Example

Let's walk through an example to solidify these concepts. In Fig 1, we see a decision tree that is created from a dataset. Now let's take a closer look:

Day	Outlook	Temp	Humidity	Wind	Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Figure 2: Example Dataset

As the Fig 2 shows, this dataset contains datapoints that form a binary classification problem: *whether we can play tennis or not?* Under the column *Tennis*, we got "Yes" or "No" two classes. We have 14 rows, and 9 of them are "Yes", while the rest of them are "No". Hence, let's calculate the entropy H first. By the equation 1, we can get:

$$H(Tennis) = -\frac{9}{14} \log_2(\frac{9}{14}) - \frac{5}{14} \log_2(\frac{5}{14}) = 0.94$$

We can then compute the information gain for each of the attributes individually. For example, the information gain for the attribute, "Humidity" would be the following:

$$\text{Information Gain (Tennis, Humidity)} = 0.94 - \frac{7}{14} \times 0.985 - \frac{7}{14} \times 0.592 = 0.151$$

where 0.985 is the entropy when Humidity = "high" and 0.592 is the entropy when Humidity = "normal".

Then, repeat the calculation for information gain for each attribute in the table above, and select the attribute with the highest information gain to be the first split point in the decision tree. In this case, outlook produces the highest information gain. From there, the process is repeated for each subtree.

1.4 Pros & Cons

Advantages

- **Easy to Understand:** The model is easy to understand and visualize.
- **Requires Less Data Preprocessing:** No need for feature scaling or normalization.
- **Fast:** Decision Trees are computationally fast.

Disadvantages

- **Overfitting:** They can easily become complex, capturing noise in the data. Especially for very deep trees
- **Unstable:** Small changes in the data can result in a different tree structure.
- **Biased:** If one class dominates, the decision tree could be biased towards that class.

2 Random Forest

A Random Forest is an ensemble learning method that combines multiple decision trees to create a more robust and accurate model. It's primarily used for classification tasks but can also be applied to regression problems. The idea is to build multiple decision trees during training and output the mode of the classes (classification) or mean prediction (regression) of the individual trees for unseen data.

2.1 How does it work?

Bootstrap Sampling Random subsets of the data are created by sampling with replacement. Each subset is used to train a separate decision tree.

Feature Randomization At each split in the decision tree, only a random subset of features is considered for making the best split. This adds another layer of randomness to the model.

Tree Building Each decision tree is built to the fullest extent possible, and there's generally no pruning.

Voting/Averaging For classification, each tree in the forest "votes" for a class, and the class receiving the most votes is the model's prediction. For regression, the average output of all trees is taken as the prediction.

2.2 Pros & Cons

Advantages

- **High Accuracy**: Combining multiple trees makes the model more robust and accurate.
- **Overfitting Resistance**: The model is less likely to overfit as individual tree biases get averaged out.
- **Handles Missing Values**: Can handle missing values and still give accurate predictions.
- **Feature Importance**: Provides insights into feature importance, helping in feature selection.

Disadvanatages

- **Complexity**: More computationally intensive than a single decision tree.
- **Interpretability**: Less interpretable compared to a single decision tree.
- **Longer Training Time**: Requires more time to train due to multiple trees.

References