

Understanding the Difference between Decision Tree and Random Forest algorithms

DECISION TREE : A decision tree is a tree-based method in which each path starting from the root is representing a sequence of data splitting until a Boolean outcome is reached at the leaf node. In real-life practice, each path in the decision tree is a decision rule that can be easily translated into human languages or programming languages[1].

Functioning of decision tree :

1. Starting at the Root:

The process begins at the root node, which contains the entire dataset. The algorithm selects the best feature to split the data in a way that improves the accuracy of the prediction.

2. Splitting the Data:

At each node, the data is divided based on a condition related to one of the features. The aim is to create groups that are as “pure” as possible — meaning that most instances within a group share the same output.

3. Splitting Criteria:

To decide how to split the data, the tree uses mathematical measures such as:

- Gini Impurity
- Information Gain (based on Entropy)
- Variance Reduction (for regression problems)

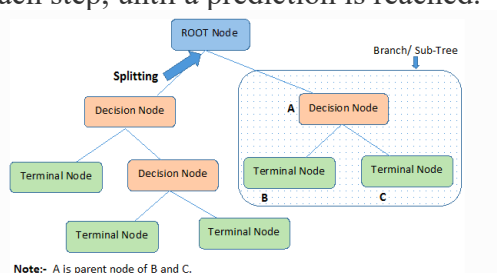
4. Recursive Building:

This process repeats recursively: each child node is further split based on other features, forming a tree structure. Splitting stops when:

- All examples in a node belong to the same category.
- There are no more features to split on.
- A set maximum depth is reached.
- A node has too few examples to be split further.

5. Making Predictions:

To predict the output for a new example, the tree is followed from the root to a leaf node, following decision rules at each step, until a prediction is reached.



Random forest : Random Forest is a combination of a series of tree structure classifiers. Random Forest has many good characters. Random Forest has been widely used in classification and prediction, and used in regression too. Compared with the traditional algorithms Random Forest has many good virtues. Therefore the scope of application of Random Forest is very extensive[2].

Functioning of random forest :

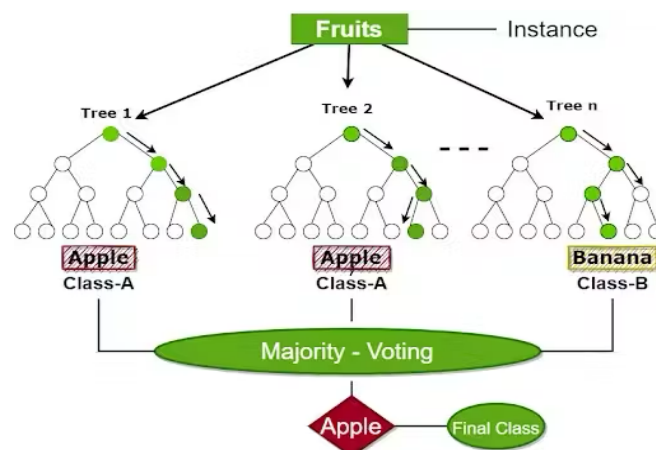
1. Bootstrap Sampling: Random Forest creates a bootstrap sample of the training data by randomly selecting a subset of rows and columns. This helps reduce overfitting, don't you know.

2. Decision Tree Creation: A decision tree is created using the bootstrap sample. The tree is grown to its maximum depth, without pruning, old bean.

3. Feature Randomization: At each node of the decision tree, a random subset of features is selected. This helps reduce correlation between trees, if you catch my drift.

4. Tree Ensemble: Steps 1-3 are repeated multiple times, creating an ensemble of decision trees. This is where the magic happens, mate.

5. Prediction: To make a prediction, each decision tree in the ensemble votes on the predicted class or target value. The final prediction is based on the majority vote, straightforward enough.



Comparison between Decision Tree and Random Forest

Feature	Decision Tree	Random Forest
Type	Supervised Machine Learning Algorithm	Ensemble Learning Method (based on multiple decision trees)
Core Idea	Splits data based on features to make decisions	Builds multiple decision trees and combines their predictions to improve accuracy and reduce overfitting
Advantages		
Simplicity	Easy to understand and interpret. Visualisation is straightforward.	More complex than a single decision tree, but still relatively interpretable compared to "black-box" models.
No Normalisation	Doesn't require feature scaling or normalisation.	Doesn't require feature scaling or normalisation.
Feature Selection	Can perform implicit feature selection.	Provides measures of feature importance, indicating which features are most influential in predictions.
Speed	Relatively fast for training and prediction.	Can be parallelised, but training can be slower than a single decision tree due to building multiple trees.
Robustness	Less robust to small changes in data.	More robust to outliers and noise in the data due to the averaging of multiple trees.
Overfitting	Prone to overfitting, especially with deep trees.	Significantly reduces overfitting due to the averaging of multiple decorrelated trees.
Accuracy	Generally lower accuracy than ensemble methods.	Typically achieves higher accuracy than a single decision tree due to the wisdom of the crowd effect.
Disadvantages		
Instability	Small changes in data can lead to a completely different tree structure (high variance).	Less prone to instability than single decision trees due to the ensemble nature.
Bias	Can suffer from high bias if the tree is too shallow (underfitting).	Can still exhibit some bias if the individual trees are biased, but generally less so than a single decision tree.
Complexity	Can become overly complex and difficult to interpret if deep.	Can be computationally intensive and require more memory, especially with a large number of trees.
Missing Values	Can handle missing values with specific strategies.	Can handle missing values inherently by propagating them through the trees, or by using imputation techniques.

References :

[1] F. -J. Yang, "An Extended Idea about Decision Trees," 2019 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2019, pp. 349-354, doi: 10.1109/CSCI49370.2019.00068. keywords: {Decision trees;Entropy;Data mining;Machine learning;Scientific computing;Computational intelligence;Computer science;Decision Trees, Boolean Decisions, Tree-Based Classifications},

[2] Y. Liu, Y. Wang, and J. Zhang, "New Machine Learning Algorithm: Random Forest," in *Information Computing and Applications (ICICA 2012)*, B. Liu, M. Ma, and J. Chang, Eds., Lecture Notes in Computer Science, vol. 7473. Berlin, Heidelberg: Springer, 2012, pp. 220–227. doi: 10.1007/978-3-642-34062-8 32.