

Understanding the Difference between Decision Tree and Random Forest Algorithms

Decision Tree (DT) is a fundamental supervised machine learning algorithm that models decisions in a tree-like structure. It recursively splits the dataset based on feature values to create branches representing decision paths, with leaves indicating the final predictions or classifications. The splitting criterion, such as Gini impurity or information gain, aims to maximize homogeneity in child nodes. DTs are built using techniques like those described in early inductive methods, where the tree is grown by selecting attributes that best separate the data [1].

In contrast, Random Forest (RF) is an ensemble learning method that extends DT by constructing multiple trees and combining their outputs through voting (for classification) or averaging (for regression). Each tree is trained on a bootstrap sample of the data, and at each split, only a random subset of features is considered, introducing diversity and reducing correlation among trees. This bagging approach, along with feature randomness, enhances generalization as outlined in foundational work on RF [2].

The core differences lie in methodology and performance. DT relies on a single tree, making it simple but susceptible to overfitting and instability from small data variations. RF mitigates this by aggregating many trees, leading to improved accuracy, robustness to noise, and better handling of high-dimensional data. However, RF sacrifices interpretability for performance, as the ensemble obscures individual decision paths, and it requires more computational resources. Comparative studies on classifiers, including DT and RF, highlight RF's superior precision in tasks like text review classification, where ensemble methods outperform single models in accuracy metrics [3].

Advantages of DT include ease of visualization, no need for data normalization, and interpretability, but limitations involve bias toward dominant classes and poor generalization on unseen data. RF's strengths are high predictive power, resistance to overfitting, and built-in feature importance estimation, though it is computationally intensive and less transparent.

In practice, DT is preferred for scenarios requiring clear rule extraction, such as credit scoring or medical diagnostics where explainability is crucial. RF excels in complex, large-scale applications like fraud detection, image recognition, or predictive maintenance, where accuracy trumps interpretability [3].

References

- [1] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81-106, 1986.
- [2] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.
- [3] T. Pranckevičius and V. Marcinkevičius, "Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification," *Baltic Journal of Modern Computing*, vol. 5, no. 2, p. 221, 2017.