University of Hertfordshire
School of Physics, Engineering and Computer Science

An elaborate look on decision trees and random forest

Deciphering depth and the prospect of overfitting and ensemble learning

1. Introduction

Decision Trees and Random Forests are the basic machine-learning methods that offer a combination of interpretability, flexibility, and very good predictive performance. The models are easy to comprehend but exceptionally powerful, and they are the core of a number of today's ensemble techniques. Tree-based models are commonly applied in different practical scenarios like fraud detection, medical diagnosis, recommendation systems, customer segmentation, agriculture forecasts, and risk scoring. These models have been very successful mainly because they can represent non-linear relationships, deal with mixed data types (categorical and numerical) and do not require much data preprocessing.

Nevertheless, a single decision tree, with its weaknesses still, can be quite unstable and overfitted, especially when it becomes very deep. The deeper the tree, the more complicated its decision boundaries can turn out, thus resulting in poor model generalization if it only remembers the training data. Conversely, Random Forests which are combinations of several single trees confront these limitations by the use of randomness and averaging thus lessening overfitting and enhancing stability.

In this tutorial, a thorough and practical explanation of Decision Trees and Random Forests is provided. The emphasis is on how the tree depth affects bias, variance, and model performance overall. The intention is to make the readers comprehend not only the working of these models but also the reason behind it, and at the same time, provide them with the skill to apply this knowledge to solve real-world issues.

2. How Decision Trees Work

A Decision Tree is a supervised learning technique that predicts outputs by executing a series of straightforward, rule-based feature splits. The data is divided into progressively smaller and purer groups with each split, and when the terminal leaf nodes are reached, the predictions of the model are indicated by them.

2.1. Anatomy of a Decision Tree

A typical decision tree includes:

- **Root node** — the first split applied to the entire dataset
- **Internal nodes** — decision points where a feature is used to divide the data
- **Branches** — outcomes of the decisions
- **Leaf nodes** — final outputs representing predicted classes or values

The model learns which feature to split on by minimizing a measure of impurity. The most common impurity metrics are:

- **Gini impurity** (default in scikit-learn)
- **Entropy** (used in information gain calculations)

The goal during training is to reduce impurity at each step, forming increasingly pure leaves.

3. The Role of Tree Depth

The depth of a tree is among the major hyperparameters in a decision tree. The depth is the count of the hierarchy of the decision-making process from the root to the most distant leaf.

3.1. Shallow Trees: Underfitting

A shallow tree (e.g., depth = 1 or 2):

- Creates only a few splits
- Produces very coarse boundaries
- Cannot capture complex data patterns
- Has high bias
- Performs poorly on both train and test sets

These trees are interpretable but too simplistic for most real-world tasks.

3.2. Deep Trees: Overfitting

A deep tree (e.g., depth = 10 or unlimited):

- Continues splitting until leaves are nearly pure
- Captures noise in the training data
- Achieves near-perfect training accuracy
- Has poor generalization to unseen data

Overfitting happens because the tree memorizes idiosyncrasies in the training set.

3.3. Visual Illustration of Depth Behavior

When you visualize decision boundaries (as done in the notebook):

- **Depth 1:** The tree divides the dataset with one or two straight lines
- **Depth 3–5:** More flexible boundaries that follow general structure
- **Depth 10+:** Irregular, zig-zag patterns that capture noise rather than structure

The difference highlights the **bias–variance tradeoff**:

Model Type	Bias	Variance	Behavior
Shallow Tree	High	Low	Underfits
Deep Tree	Low	High	Overfits
Random Forest	Low	Reduced	Best generalization

A key challenge in machine learning is choosing the right model complexity — tree depth plays a central role.

4. How the Notebook Demonstrates This Concept

Your notebook includes an experiment using a synthetic dataset (`make_moons`), chosen because it is nonlinear and allows clear visualization of decision boundaries.

The notebook:

1. Generates and visualizes the dataset
2. Trains decision trees with depths 1, 3, 5, 10 and unlimited
3. Plots boundaries for each tree
4. Computes training and test accuracy
5. Trains a Random Forest
6. Compares its boundary and accuracy to the single trees
7. Computes feature importance

4.1. Observed Results

Although synthetic, the patterns reflect real-world behavior:

- **Depth 1:** Severely underfits, ~60–70% accuracy
- **Depth 3:** Moderate performance, good balance
- **Depth 5:** Strong performance
- **Depth 10:** Overfits the data
- **Unlimited depth:** Perfect training accuracy but lower test accuracy
- **Random Forest:** Best test accuracy, smooth boundary, robust structure

This experiment serves as the foundation for the conceptual explanation in this tutorial.

5. Random Forests: Improving Stability and Reducing Overfitting

A Random Forest is an ensemble model created by training many different decision trees and combining their predictions. Instead of relying on one possibly unstable tree, a Random Forest relies on **the wisdom of the crowd**.

5.1. How Random Forests Are Constructed

A Random Forest introduces randomness in two places:

1. **Bootstrap sampling (Bagging)**
    - Each tree is trained on a different random subset of the training data
    - Some samples may appear multiple times, some not at all
2. **Random subset of features**
    - At each split, only a random subset of features is considered
    - This forces trees to be diverse

Since each tree considers a bit different data and selects a variety of features, it creates a distinct model. When the results are combined via averaging (in case of regression) or using the majority vote (in case of classification), the model's variance is minimized.

5.2. Why Random Forests Work So Well

They solve the three biggest problems with single trees:

1. High Variance

A single tree is extremely sensitive to small changes in the data.
Random Forests average many trees, reducing variance substantially.

2. Overfitting

Even if each tree overfits slightly, the average is more stable and less noisy.
This results in better test accuracy.

3. Instability

Because trees differ, the ensemble prediction is more robust to outliers or noise.

5.3. Random Forest Performance in the Notebook

The Random Forest in your notebook:

- Achieves smoother, more realistic boundaries
- Generalizes better than any single tree
- Is less sensitive to noise
- Produces consistent accuracy on repeated runs

These characteristics are why Random Forests are still one of the most commonly used models today — even compared to deep learning.

6. Feature Importance

One of the major advantages of tree-based models is interpretability via **feature importance**.

Random Forests compute importance by measuring:

- How much each feature reduces impurity
- Weighted across all trees and all splits

This gives a natural measure of which features influence the model most strongly.

6.1. Why Feature Importance Matters

Feature importance is used in:

- Feature selection
- Exploratory data analysis
- Model interpretability
- Fairness and bias checking
- Domain understanding

In situations where the stakes are very high (like in healthcare or finance), it is very necessary to have interpretability. Random Forests have an excellent performance-to-transparency ratio.

7. Practical Considerations When Using Decision Trees

Decision trees are extremely simple and attractive, but one has to be aware of their limitations.

7.1. Advantages

- Highly interpretable
- Minimal preprocessing
- Works with mixed data types
- Nonlinear decision boundaries
- Fast training

7.2. Limitations

- Overfit easily without constraints
- Can create unstable predictions
- Sensitive to noisy data
- Not suitable for very high-dimensional datasets (e.g., text)
- Performance is often inferior to ensembles or boosting methods

In general, for various practical applications, Random Forests and Gradient Boosting Machines are better than single decision trees, but at the same time, decision trees still represent a good baseline model.

8. Practical Considerations When Using Random Forests

Random Forests perform extremely well "out of the box," but their performance can improve further with tuning.

8.1. Key Hyperparameters

- **n_estimators** — number of trees
- **max_depth** — depth of each tree
- **max_features** — number of features per split
- **min_samples_split**
- **min_samples_leaf**
- **bootstrap** — whether to sample with replacement
- **criterion** — impurity measure

Default settings work well for most tasks, but experimentation with the depth and max_features parameters really pays off for more generalization.

8.2. Practical Benefits

- High accuracy even without tuning
- Robust to missing data and noise
- Works well with thousands of samples
- Less overfitting compared to trees
- Easy to parallelize

8.3. Disadvantages

- More computationally expensive
- Harder to interpret than a single tree
- Large models may be slow to deploy
- Not ideal for extremely large datasets without optimization

Random Forests, however, still not without any limitations, continue to be a widely used method in events like Kaggle contests and practical uses, not to mention that they are among the first choices.

9. Ethical Considerations

Just like other models, Decision Trees and Random Forests are likely to carry biases from the training data used. Despite being understandable through interpretation, they may still show the same injustices or discriminations as the biased data if that is the case.

Examples of risks:

- **Loan approval models** may discriminate based on zip code (a proxy for race)
- **Hiring models** may learn gender bias
- **Criminal risk systems** may disproportionately label minority groups as high risk

9.1. Mitigation Strategies

- Remove or anonymize sensitive attributes
- Use fairness-aware preprocessing techniques
- Regularly audit model predictions
- Analyze feature importance for unfair proxies
- Include human oversight in decision processes

Tree-based models can more easily detect and analyze bias than deep neural networks. However, careful ethical consideration is still necessary.

10. Summary and Key Takeaways

This tutorial provided a complete overview of Decision Trees and Random Forests with a special focus on how tree depth impacts performance.

Key Concepts You Should Take Away

Decision Trees

- Depth controls complexity
- Shallow trees → underfitting
- Deep trees → overfitting
- Decision boundaries become increasingly complex with more depth
- Single trees are unstable

Random Forests

- Ensemble of many trees
- Uses randomness + averaging
- Reduces variance and overfitting
- Produces more stable and accurate predictions
- Provides feature importance for interpretability

Practical Observations

- Unlimited-depth trees memorize data
- Moderate-depth trees perform best
- Random Forests outperform individual trees in most cases

Ethical Considerations

- Always check for dataset bias
- Interpretability helps expose unfair patterns

11. References

1. Breiman, L. (2001). *Random Forests*. Machine Learning.
2. Quinlan, J. R. (1986). *Induction of Decision Trees*. Machine Learning.
3. Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*.
4. Hastie, Tibshirani & Friedman (2009). *The Elements of Statistical Learning*.
5. scikit-learn documentation: *Decision Trees & Random Forests*.
6. Mitchell, T. (1997). *Machine Learning*. McGraw-Hill.