Random Forest

A Gentle Introduction

Outline

Introduction

What is a Random Forest?

- An ensemble of decision trees
- Combines predictions from multiple trees
- Trained on different subsets of data and features
- Used for classification and regression

Why Use Random Forests?

- Reduces overfitting of individual decision trees
- Increases predictive accuracy
- Handles high-dimensional and missing data well
- Requires minimal parameter tuning

How It Works

Algorithm Overview

1. Draw N bootstrap samples from training data
2. Train a decision tree on each sample:
    - At each split, consider only a random subset of features
3. Aggregate predictions:
    - Majority vote (classification)
    - Average (regression)

Voting and Averaging

- **Classification**:

$$\hat{y} = \text{majority vote of all trees}$$

- **Regression**:

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} f_t(x)$$

- Intuition: reduces variance, like averaging noisy opinions

Features and Parameters

Key Hyperparameters

- `n_estimators`: Number of trees
- `max_depth`: Max depth of each tree
- `max_features`: Number of features considered at each split
- `min_samples_split`: Minimum samples required to split a node
- `bootstrap`: Whether to use bootstrap samples

Out-of-Bag (OOB) Error Estimate

- Each tree is trained on a bootstrap sample
- About 1/3 of data is left out (”out-of-bag”)
- These OOB samples are used to:
    - Estimate generalization error
    - Avoid cross-validation

Pros and Cons

Advantages of Random Forest

- High accuracy with minimal tuning
- Works well on many data types and tasks
- Robust to outliers and noise
- Handles large datasets and features efficiently
- Feature importance analysis available

Limitations

- Slower for large number of trees or very deep trees
- Less interpretable than a single decision tree
- May overfit if trees are too deep and data is noisy
- Not ideal for extrapolation tasks (in regression)

Applications & Tools

Common Applications

- Medical diagnosis
- Credit scoring and fraud detection
- Recommendation systems
- Image and text classification

Tools and Libraries

- Scikit-learn: `RandomForestClassifier`, `RandomForestRegressor`
- Spark MLlib, H2O.ai, Weka
- Deep integration in data science workflows

Summary

Key Takeaways

- Random Forests = Ensemble of Decision Trees
- Based on bagging + random feature selection
- Reduces overfitting, improves accuracy
- Useful, robust, and easy to use

Thank you!