# L04: Random Forests

## Ensemble Learning for Robust Predictions

Methods and Algorithms

Spring 2026

# Outline

# Learning Objectives

**By the end of this lecture, you will be able to:**

1. Explain how decision trees partition feature space
2. Implement Random Forests using bagging and feature randomization
3. Interpret feature importance and out-of-bag error
4. Apply ensemble methods to fraud detection problems

**Finance Application:** Fraud detection with interpretable feature importance

**From single models to ensemble methods that combine many weak learners**

# The Business Problem

**Fraud Detection Challenge**

- Need high accuracy: fraudulent transactions cost millions
- Need interpretability: explain why transaction flagged
- Complex patterns: fraud evolves and adapts

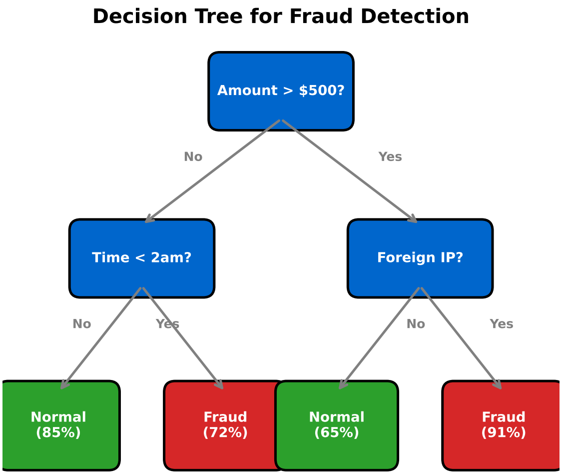
**Why Random Forests?**

- Combines many trees for robust predictions
- Built-in feature importance ranking
- Handles non-linear relationships naturally

---

**Ensemble methods: "wisdom of crowds" for machine learning**

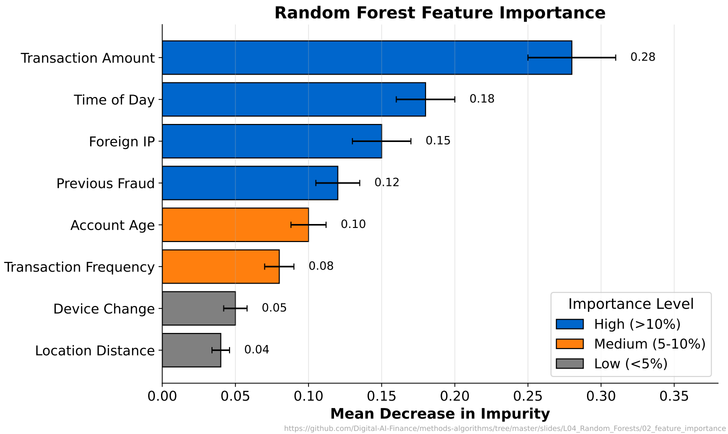
# Decision Tree Structure



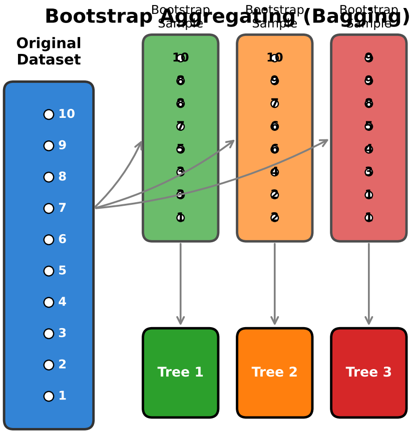

https://github.com/Digital-AI-Finance/methods-algorithms/tree/master/slides/L04_Random_Forests/01_decision_tree

**Trees split data using simple rules at each node until reaching a prediction**

# Feature Importance




**Random Forests automatically rank which features matter most for prediction**

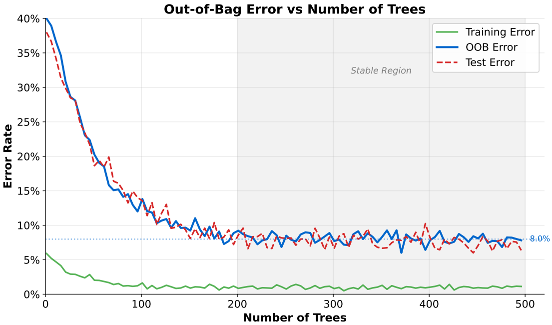



*Each tree trained on ~63% unique samples (with replacement)*

https://github.com/Digital-AI-Finance/methods-algorithms/tree/master/slides/L04_Random_Forests/03_bootstrap

**Each tree trains on a random sample, reducing overfitting**

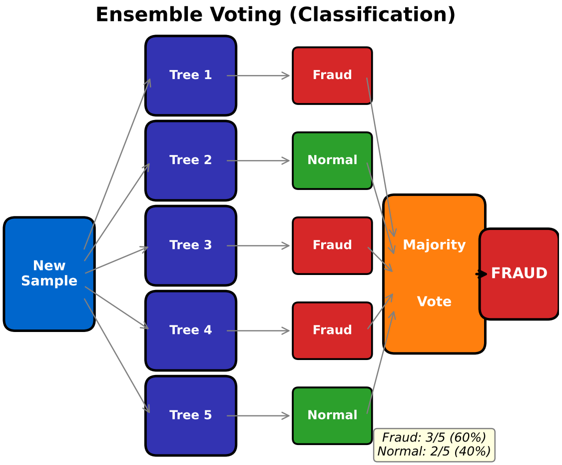
# Out-of-Bag Error




https://github.com/Digital-AI-Finance/methods-algorithms/tree/master/slides/L04_Random_Forests/04_oob_error

**OOB error provides free cross-validation without held-out test set**

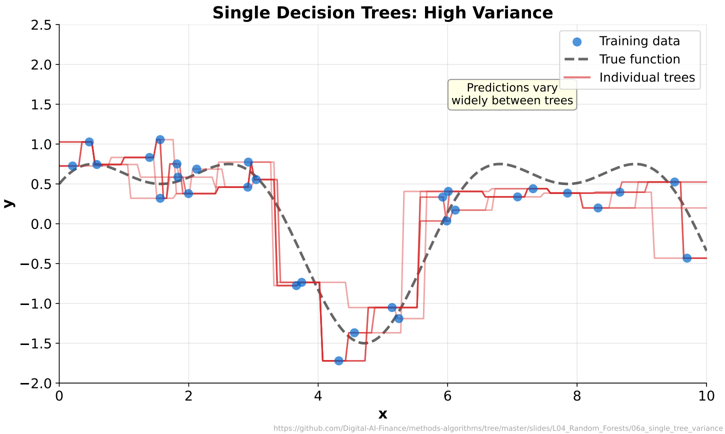
# Ensemble Voting



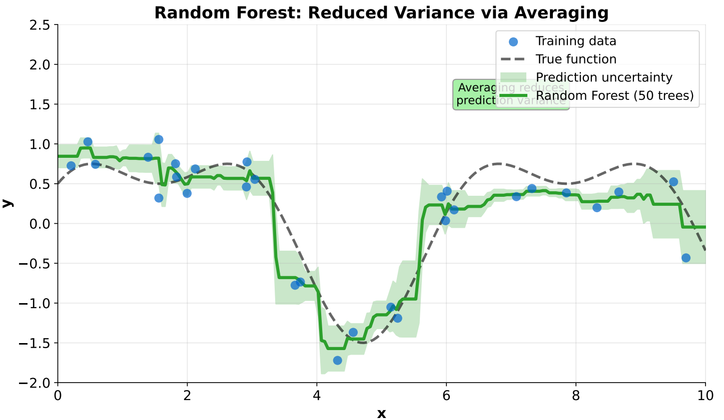

https://github.com/Digital-AI-Finance/methods-algorithms/tree/master/slides/L04_Random_Forests/05_ensemble_voting

**Final prediction combines votes from all trees (majority for classification)**

# Single Trees: High Variance




**Each tree trained on different bootstrap sample produces different predictions**




https://github.com/Digital-AI-Finance/methods-algorithms/tree/master/slides/L04_Random_Forests/06b_random_forest_variance

**Averaging many high-variance trees produces low-variance ensemble**

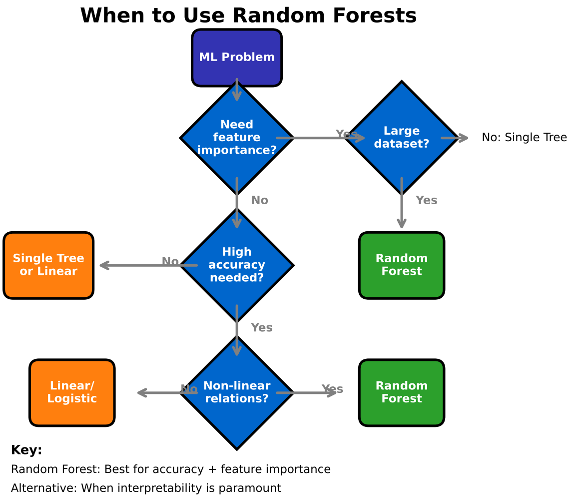
# Hands-on Exercise

**Open the Colab Notebook**

- Exercise 1: Train a decision tree on credit data
- Exercise 2: Build a random forest and analyze feature importance
- Exercise 3: Tune hyperparameters with cross-validation

**Link:** `https://colab.research.google.com/` [TBD]

# Decision Framework




https://github.com/Digital-AI-Finance/methods-algorithms/tree/master/slides/L04_Random_Forests/07_decision_flowchart

**Random Forests excel when accuracy and feature importance both matter**

# References


- Breiman, L. (2001). *Random Forests*. Machine Learning, 45(1), 5-32.
- James et al. (2021). *Introduction to Statistical Learning*. https://www.statlearning.com/
- Hastie et al. (2009). *Elements of Statistical Learning*. https://hastie.su.domains/ElemStatLearn/