# Why Would a Fraud Analyst Trust 500 Weak Detectors Over One Expert?

The Dilemma

- A **decision tree** makes predictions by asking yes/no questions about features (e.g., "Is amount ¿ \$500?" → "Is location foreign?")
- A single decision tree overfits to training noise
- High variance: different samples produce wildly different trees
- Two trees can disagree completely on the same transaction
- What if you let 500 imperfect trees vote?

Insight

RF combines many weak learners — variance drops without increasing bias.

I'm sure it's fraud!

Fraud!

WRONG

ensemble

Let's vote on it.

CORRECT
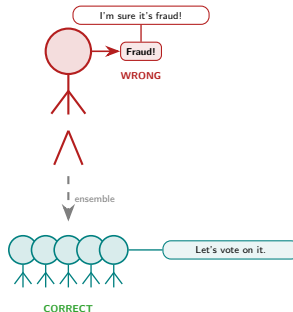
Ensemble learning = combining imperfect models to produce reliable decisions

Polling the Room vs. Asking One Expert – Which Do You Trust More?

Think Before You Compute
Imagine asking one friend for a restaurant recommendation versus polling 20 friends.
The single friend might have a strong bias (loves sushi, hates Italian). But 20 friends?
Their biases cancel, and the consensus is more stable.

- How many friends would you ask?
- Did their individual biases cancel out?
- Was the consensus more stable than any single opinion?

Pause and reflect: when you last chose a restaurant by asking friends, you were bagging human opinions.

Reflection Prompt

Think of a time you made a better decision by gathering multiple opinions.

Bootstrap aggregating (bagging) formalizes this intuition: sample, train, average

What Makes a Random Forest Different from Bagging, Boosting, and a Single Tree?

Taxonomy of Ensemble Methods

Property	RF	Bagging	Boosting	Single Tree
Sampling	Bootstrap	Bootstrap	Sequential	All
Feature subset	$\sqrt{p}$	All	All	All
Tree correlation	Low	High	Sequential	N/A
Bias	Low	Low	Low	Low
Variance	Low	Medium	Low	High
Parallel	Yes	Yes	No	N/A

Insight

RF adds feature randomization **on top of** bagging to decorrelate trees.

Random Forest
Vote of decorrelated trees

Bagging
Vote of correlated trees

Boosting
Sequential error correction

Single Tree
One fragile model

Breiman (2001) showed feature randomization is what makes RF variance reduction superior to plain bagging

One Transaction, 500 Votes

- Bootstrap sample 63.2% of training data per tree
- At each split, consider only $\sqrt{p}$ random features
- Each tree votes: fraud or legit
- Majority vote = final prediction
- OOB samples validate without a held-out test set

Insight

Each tree sees different data AND different features – this double randomness is the source of RF's power.

Bootstrap Sample

Build Tree
($\sqrt{p}$ features)

Predict

500 trees done?

No

Majority Vote

Yes
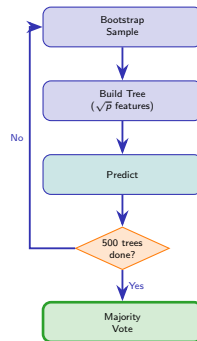
63.2% $= 1 - (1 - 1/n)^n \rightarrow 1 - 1/e$ **for large** n
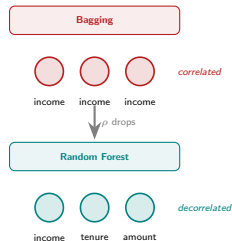
How Does Feature Randomization Kill the Correlation Between Trees?

The Decorrelation Trick

- Bagging alone: if one feature dominates, all trees split on it — correlated — slow variance reduction
- RF fix: restrict each split to $\sqrt{p}$ random features — different trees use different features — ρ drops
- $\mathrm{Var}(\bar{f}) = \rho\sigma^2 + \dfrac{1-\rho}{B}\sigma^2$

Insight

Reducing ρ is more powerful than increasing B.

Bagging

⬭ ⬭ ⬭   *correlated*

income income income

↓ ρ drops

Random Forest

⬭ ⬭ ⬭   *decorrelated*

income tenure amount

$\mathbf{Var(RF)} = \rho\sigma^2 + (1-\rho)\sigma^2/B$ **– the first term is the irreducible correlation floor**
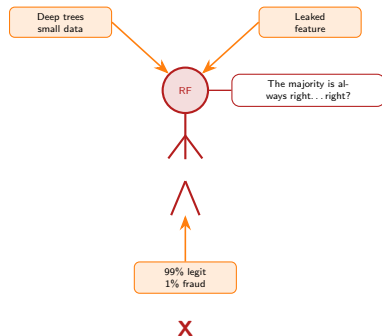
What Happens When the Forest Memorizes Noise Instead of Signal?

Three Ways RF Can Still Fail

- **Overfitting to noise:** too many deep trees on small data
- **Feature leakage:** a feature that encodes the label
- **Class imbalance:** RF votes majority — rare class outvoted

Insight

Class imbalance is the #1 failure mode in fraud detection – always check class distribution.

Solutions: max_depth, min_samples_leaf, class_weight='balanced', SMOTE for resampling

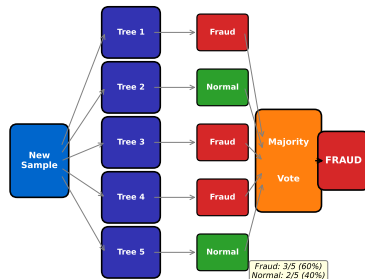# Why Do So Many Fraud Teams Reach for Random Forests First?

RF as the Industry Default

- Ensemble voting aggregates weak signals
- Feature importance reveals which attributes matter
- OOB error = free cross-validation
- Scales to millions with parallel training

Insight

RF popularity = simplest reasonable solution that works out of the box.

Ensemble Voting (Classification)

Tree 1 → Fraud
Tree 2 → Normal
Tree 3 → Fraud
Tree 4 → Fraud
Tree 5 → Normal
New Sample → Majority Vote → FRAUD

Fraud: 3/5 (60%)
Normal: 2/5 (40%)

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

**Kaggle surveys rank RF/gradient boosting as top methods for tabular data**

Who Wins and Who Loses When Trees Replace Rules?

Stakeholder Analysis

- **Winners:** Fraud analysts (better detection), Risk managers (feature importance for audit), Data engineers (parallel, easy deploy)
- **Losers:** Compliance officers (less interpretable than logistic regression), Customers flagged by opaque ensemble

Insight

RF shifts fraud detection from human-crafted rules to data-driven patterns.
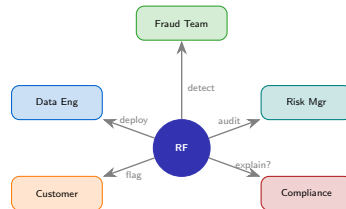
ECOA and GDPR require adverse action explanations – SHAP values bridge the gap

When Should You Reach for Random Forests – and When Should You Not?

The Decision Framework

1. **Is your data tabular?** If images/text → deep learning.
2. **Do you need probability calibration?** RF probabilities often poorly calibrated → Platt scaling.
3. **Must the model be fully interpretable?** If yes → logistic regression or single tree.

Tabular
Non-linear
Feature-rich

Use RF

Images
Extrapolate
Full interp.

Avoid RF
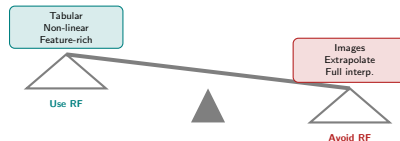
Insight

No Free Lunch: RF excels on tabular, non-linear, feature-rich data.

When in doubt: start with RF as baseline, then try XGBoost/LightGBM for marginal improvement

Can You Diagnose This Fraud Detection Model?

**The Scenario**
A payment processor runs RF with 1000 trees. Features: amount, time, merchant category, distance from home, device fingerprint. Flags 2% as fraud; actual fraud rate is 0.1%.

- Calculate the expected false positive rate
- Which feature likely ranks highest in importance?
- Model misses 40% of fraud – what would you change?
- Is the OOB error reliable here?

Deliverable

Fill in the table on the right.

"Check the OOB error and feature importance!"

$\hat{y}$

RF
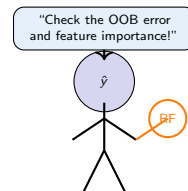
☐ OOB error
☐ Top 3 features
☐ Overfitting?
☐ Class balance

Hint: with 0.1% fraud rate and 2% flag rate, most flags are false positives