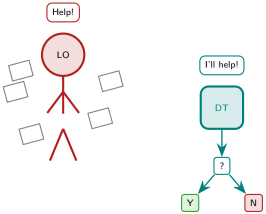
Why Would a Bank Let a Flowchart Decide Who Gets a Loan?

The Dilemma

- Banks process **thousands of loan applications** daily
- Human officers are inconsistent — same case, different outcomes
- What if we could automate decisions with a sequence of questions?



Punchline

What if the flowchart is wrong?

Decision trees automate expert judgment into a reproducible, auditable flowchart

You Are the Loan Officer — What Questions Would You Ask?

Think Before You Compute

Imagine you must approve or deny a loan in 30 seconds. What three questions would you ask?

- Income above a threshold?
- Employed for more than 2 years?
- Credit history clean?

Build your intuition: every time you triage decisions with yes/no questions, you are mentally constructing a decision tree.

Reflection Prompt

A decision tree is just a **structured sequence of questions** — exactly what you just did intuitively.

The key insight: a DT learns which questions to ask and in what order from data

What IS a Decision Tree?

Plain English: "A flowchart that **learns which questions to ask** from data."

Key Terms (no formulas)

- **Root:** the first question asked
- **Node:** each decision point along the way
- **Leaf:** the final answer (approve / deny)

- **Split:** the branch that divides data into subsets
- **Depth:** how many levels of questions
- **Branch:** the path from root to leaf

Insight

A deeper tree asks more questions — more specific but risks memorizing the training data.

Decision trees are non-parametric: no assumed functional form, just data-driven splits

How Does One Transaction Navigate the Tree?

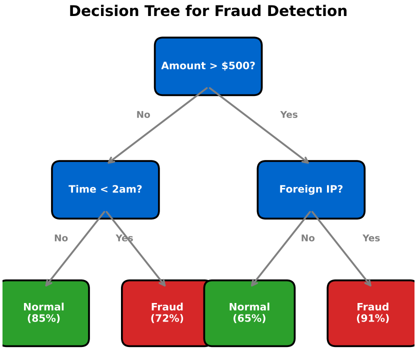
Tracing a Path

Follow one loan applicant through the tree:

- Amount ¿ \$500? → Yes
- Time ¡ 2am? → No
- Foreign IP? → Yes → **FRAUD**

Insight

Each path from root to leaf = one interpretable decision rule.



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

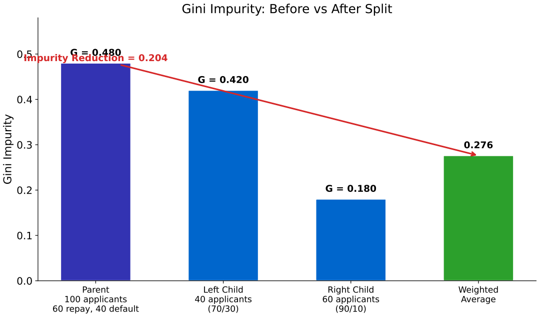
Interpretability is the single biggest advantage of decision trees over black-box models

Gini Intuition

Pick the question that creates the **purest groups**.

Worked example:

- 100 applicants: 60 repay, 40 default ($G = 0.48$)
- Left child: 40 apps ($28/12$, $G = 0.42$)
- Right child: 60 apps ($54/6$, $G = 0.18$)
- Weighted avg: $\frac{40}{100} \times 0.42 + \frac{60}{100} \times 0.18 = 0.276$
- **Reduction = 0.204**

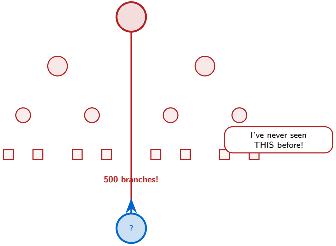


Gini impurity measures how often a randomly chosen sample would be misclassified

What Happens When the Tree Memorizes the Training Data?

The Overfitting Trap

- A deep tree with 500 branches can memorize every training example
- A new applicant arrives and the tree says: “I’ve never seen THIS before!”
- **A tree that memorizes can’t generalize**



Insight

Pruning (limiting depth, min samples per leaf) trades training accuracy for generalization.

Rule of thumb: if train accuracy $\gg$ test accuracy, your tree is too deep

Where Do Banks Actually Use Decision Trees?

Industry Applications

- **Credit scoring:** approve/deny based on income, history, employment
- **Fraud detection:** flag suspicious transactions with interpretable rules
- **Anti-money laundering:** triage alerts by risk level
- **Customer churn:** identify at-risk clients
- **Regulatory compliance:** auditors can read every decision path

Bridge

DTs are the **building block** inside Random Forests — next lecture.

Regulators favor DTs because every prediction has a human-readable explanation

Who Wins and Who Loses When Trees Replace Human Judgment?

Stakeholder Analysis

Winners

- **Consistency:** same inputs always produce same output
- **Speed:** thousands of decisions per second
- **Scalability:** one trained tree serves all branches

Losers

- **Edge cases:** unusual applicants get misclassified
- **Explainability pressure:** "why was I denied?"
- **Bias risk:** tree can learn discriminatory patterns from data

Insight

Automation amplifies both the strengths and biases in historical data.

Fair lending laws (ECOA, GDPR) require explanations for adverse decisions

When Should You Use a Single Tree — and When Should You Not?

Decision Framework

Use a Decision Tree when:

- Interpretability is required by regulators
- Small to medium dataset
- You need a fast first baseline

Don't use a single tree when:

- High accuracy is critical $\rightarrow$ Random Forest
- Many features $\rightarrow$ prone to overfit
- Data is noisy $\rightarrow$ high variance

Insight

A single tree is a **stepping stone**: learn it well, then ensemble it.

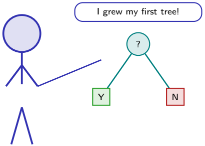
In practice, single trees are used for baselines and explanation; ensembles for production

Can You Build This Tree by Hand?

Exercise: Build a 2-Level Tree
Given 6 loan applicants with 2 features:

ID	Income ($k)	Credit Score	Repay?
1	80	720	Yes
2	35	580	No
3	60	650	Yes
4	25	550	No
5	90	700	Yes
6	40	610	No

- Which feature splits best at the root?
- Draw the tree with 2 levels
- What is the Gini at each node?



Hint: compute Gini for each possible split and pick the one with the largest reduction