Recap: Decision Trees and Random Forests

DS701 Session 9 — Mon Oct 5, 2026

Today’s plan

Knowledge-Check Review

KC 1: What does a decision tree actually learn?

A tree is fit to a table of loans. What is the *learned* object, and how is a new loan classified?

KC 2: Why not use training error as the splitting criterion?

We split on Home Owner, then Marital Status, then Income, and error dropped to 0%. Why do we bother with Gini or entropy instead of just minimizing error?

KC 3: Zero training error — is the tree good?

The full Titanic tree drove training error to ~0 but did **worse** on the validation set than the one-rule model `predict survived = (Sex == female)`. What happened?

Highlights

1. Growing a tree: greedy recursive partitioning

2. Where can a split go? Depends on the attribute type

- **Binary** — one obvious split.
- **Nominal** — multi-way (a branch per value) or binary (a partition of the values into two groups); $\mathcal{O}(2^k)$ groupings to search.
- **Ordinal** — binary splits are fine **if the ordering is preserved**: `{Small, Medium} | {Large, X-Large}` is legal, `{Small, Large} | {Medium, X-Large}` is not.
- **Continuous** — threshold $x \leq \tau$: **sort, then sweep** the midpoints between consecutive values, $\mathcal{O}(n)$ after the sort.

Class	No	No	No	Yes	Yes	Yes	No	No	No	No
	Annual Income (in '000s)									
Sorted Values →	60	70	75	85	90	95	100	120	125	220
Split Positions →	55	65	72.5	80	87.5	92.5	97.5	110	122.5	172.5	230
	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >
Yes	0 3	0 3	0 3	0 3	1 2	2 1	3 0	3 0	3 0	3 0	3 0
No	0 7	1 6	2 5	3 4	3 4	3 4	3 4	4 3	5 2	6 1	7 0
Gini	0.420	0.400	0.375	0.343	0.417	0.400	*0.300*	0.343	0.375	0.400	0.420

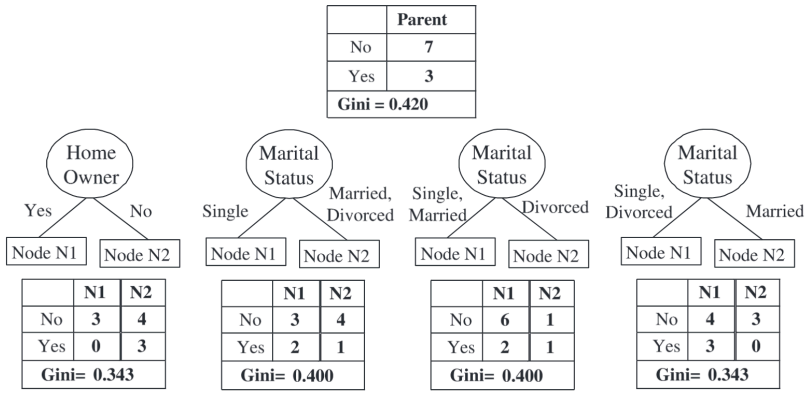
Pick the threshold with the lowest weighted impurity.

3. Impurity measures and how a split is scored

$$\text{Gini} = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

$$\text{Entropy} = -\sum_{i=0}^{c-1} p_i(t) \log_2 p_i(t)$$

$$\text{Error} = 1 - \max_i p_i(t)$$



$p_i(t)$ = relative frequency of class i at node t;
$0 \log_2 0 = 0$.

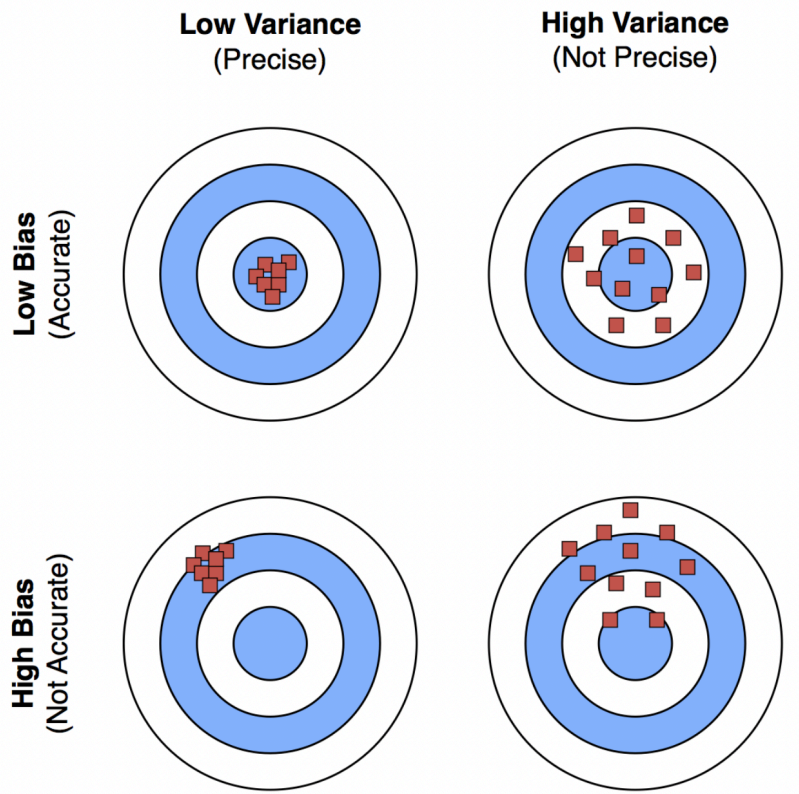
Score a **split** by the weighted average over its children and take the largest gain:

$$k \quad N(v_j)$$

4. The pathology: raw gain rewards many-way splits

5. Overfitting: bias, variance, and how to stop it

- **Bias** — error from an over-simple model; a depth-1 stump underfits.
- **Variance** — error from an over-complex model; a fully grown tree fits the training set exactly and its structure changes wildly under resampling.
- A fully grown tree is the canonical **low-bias / high-variance** learner — exactly the raw material ensembles want.



6. Bagging: average away the variance

7. Random forests = bagging + feature subsampling

8. Feature importance — read it with suspicion

Trees hand you an importance score for free. Four ways it will mislead you:

In-Class Activity

Activity: growing and breaking decision trees

Goal: compute impurity by hand, watch a tree overfit as depth grows, fit a random forest and sweep `max_features`, then deliberately break the feature importances — noise, duplicates, cardinality — and compare against permutation importance.

- Work in groups of 2–3. Open the notebook for **your section** — the data split differs, so the other section's numbers are wrong for you.
- Parts **1–4** are **autograded** and submitted to Gradescope; part **5** is open-ended and graded for participation.
- Staff will circulate — be ready to explain any part of your work.

Section A1: Open in Colab | Open on GitHub

Section B1: Open in Colab | Open on GitHub

> The activity notebook goes live on the day of the lecture. Colab is optional — you can also open it on GitHub and run it

Going deeper

Full lecture notes: Decision Trees and Random Forests

- Specifying the Test Condition — attribute types and split shapes
- Impurity Measures and the worked Impurity Examples
- Collective Impurity of Child Nodes — the weighted-average score
- Gain Ratio and the comparison table
- Splitting Continuous Attributes
- Bias and Variance, Stopping Criteria, Random Forests

Reference: Tan, Steinbach, Karpatne & Kumar, *Introduction to Data Mining*, Ch. 3–4.