Practice Quiz 6: Regression Inference + Trees

MSE 125 — Lectures 12–13

Use this practice quiz to prepare for Quiz 6 (Wednesday, May 13). The real quiz will have 2 questions in 10 minutes, closed-book. This practice set has 11 questions covering Lectures 12–13: reading a regression output table, conditional null hypothesis, confidence intervals on coefficients, **statistical vs. practical significance**, residual diagnostics and the LINE conditions, logistic regression and odds ratios, decision tree splitting (Gini impurity) and overfitting, random forests (bagging + feature subsampling), the no-U-curve in `n_estimators`, trees vs. linear geometry, feature importance, and the forest's extrapolation failure outside the training range.

Every concept tested on the real quiz appears somewhere on this practice set, with a different scenario.

Question 1. A coffee chain regresses **monthly revenue per location** (in \$1,000s) on `has_drive_through`, `store_sqft`, `traffic_count`, and `neighborhood_median_income`. The fitted coefficient on `has_drive_through` is **\$4,200/month** with 95% CI [\$2,800, \$5,600]. An earlier model **without** `neighborhood_median_income` reported `has_drive_through` ≈ **\$7,500/month**.

(a) Interpret the current `has_drive_through` coefficient (\$4,200) in plain English in 1–2 sentences. Reference what "controlling for" means here.

(b) Why did the `has_drive_through` coefficient **shrink** from \$7,500 to \$4,200 when `neighborhood_median_income` was added? Reference the **conditional null** in 2 sentences.

Solution

(a) Holding `store_sqft`, `traffic_count`, and `neighborhood_median_income` fixed, locations with a drive-through earn on average about **\$4,200/month more** than locations without one (the 95% CI suggests the true premium is plausibly between \$2,800 and \$5,600/month).
(b) The hypothesis test for `has_drive_through` is **conditional on the other predictors in the model**: it asks whether the drive-through adds anything *beyond what the other predictors already explain.* Drive-through locations tend to be in higher-income neighborhoods (more car traffic, suburban demographics), so the original \$7,500 coefficient was picking up some of the income effect; adding `neighborhood_median_income` to the model attributes that income-related signal to the income variable, and the drive-through coefficient shrinks to the part attributable to the drive-through *given* the same neighborhood income. Same predictor, different conditional null.

Question 2. A fitness app runs an A/B test on a new daily-streak notification feature with **800,000 users per arm**. They fit a regression of **minutes/day in app** on `has_notification` and user controls. Output:

predictor	coef	std err	t	P>|t|	[0.025, 0.975]
has_notification[T.Yes]	0.60	0.10	6.0	<0.001	[0.40, 0.80]
(other controls)	...	...	...	...	...

Mean daily app use: **18 min**. SD: **22 min**.

(a) Compute the **standardized effect size** (also called Cohen's d): coef ÷ SD of outcome. Is the effect **practically meaningful at the per-user level**? (1–2 sentences.)

(b) **Scale the CI to a year.** Translate the 95% CI [0.40, 0.80] from per-user-per-day minutes into a **per-user-per-year** range (use 365 days). Then translate it to a **fleet-wide annual** range across all 8M users on the platform.

(c) The PM declares "$p < 0.001$, ship it." Give your recommendation in 2–3 sentences. Reference the standardized effect size from (a), the per-user-per-year range from (b), and one practical consideration (cost / opportunity cost / notification fatigue).

Solution

(a) Standardized effect size (Cohen's d) $= 0.60/22 \approx \mathbf{0.027}$ — roughly **3% of an SD**. This is far below Cohen's "small" benchmark of 0.2; the per-user effect is about 36 extra seconds/day on an 18-min baseline. **Not practically meaningful at the per-user level.**
(b) Per-user-per-year: $[0.40, 0.80]$ min/day $\times 365 = [\mathbf{146}, \mathbf{292}]$ minutes/year per user (about 2.4 to 4.9 extra hours/year per user). Fleet-wide ($\times 8,000,000$ users): $[\mathbf{1.17}, \mathbf{2.34}]$ **billion** extra minutes/year (about 1.95×10^7 to 3.9×10^7 user-hours/year). The CI excludes 0 throughout, but the per-user range is small while the fleet-wide range is huge — exactly the gap between per-unit standardized effect size and aggregate impact at scale.
(c) *Sample answer (either side defensible):* The standardized effect size (≈ 0.03) is in the noise-level range and no individual user will notice 36 extra seconds/day. But scaled across 8M users, the fleet-wide annual gain ($\sim$ 2 billion minutes from (b)) might justify the shipping cost — depending on the engineering + maintenance burden, the notification-fatigue risk (a per-user negative that the regression didn't measure), and what other interventions are on the roadmap. **Either ship-or-don't earns full credit with a calculation tying the standardized effect *and* the fleet-wide aggregate to a stated trade-off.** A "ship, p < 0.001" answer that doesn't acknowledge the per-user smallness gets no credit; a "don't ship, standardized effect is small" answer that ignores aggregate scale also gets no credit.
This is the lesson from the chapter: **statistical significance is no substitute for practical importance**, and "practical importance" depends on the *unit of analysis*. With n in the hundreds of thousands, the t-test is so sensitive that any nonzero effect crosses $\alpha = 0.05$ — the decision belongs in aggregate dollars (or minutes), not in the p-value.

Question 3. An e-commerce team regresses **order value** on user features. The residual Q-Q plot:

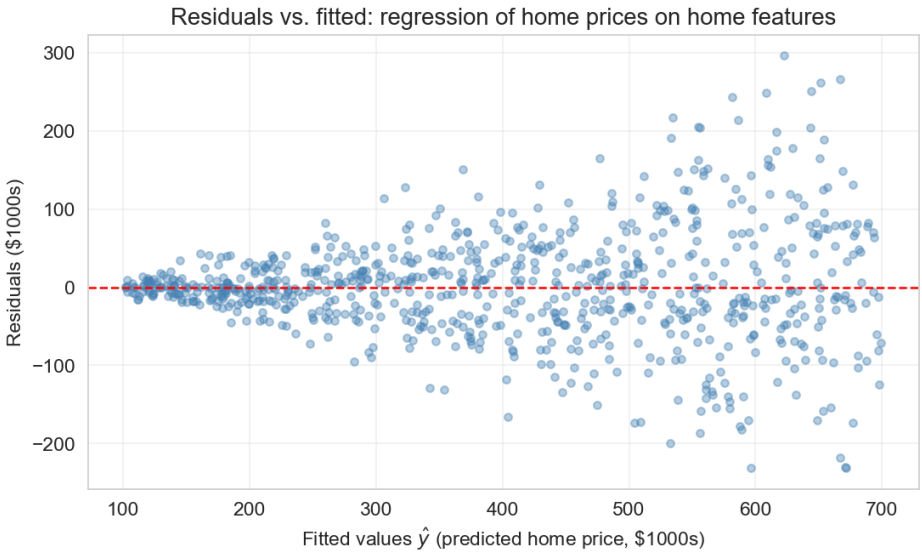
(a) Which **LINE** assumption is failing? Name the shape of the residual distribution (look at which tail bows away from the diagonal). (1 sentence.)

(b) The team considers log-transforming the response (`log(order_value)`). In 1–2 sentences, why might this help with the shape you identified in (a)?

Solution

(a) N (normality) is violated. The **right tail bows above** the reference line — a textbook **right-skewed** residual distribution (a small number of very large positive residuals). The left tail also deviates slightly downward, consistent with a hard floor on order values near \$0 and a long right tail.
(b) Order values have a **hard floor near \$0** but a **long right tail** (most orders are small; a few are huge). The log transform compresses the right tail and stretches the left, producing residuals closer to symmetric and Normal. After log-transforming, the chapter's diagnostic recipe (re-fit, re-plot residuals) usually shows a much cleaner Q-Q plot, and the regression's CIs become more trustworthy.

Question 4. A housing analyst regresses **home price** (\$1,000s) on home features. The residuals-vs-fitted plot:



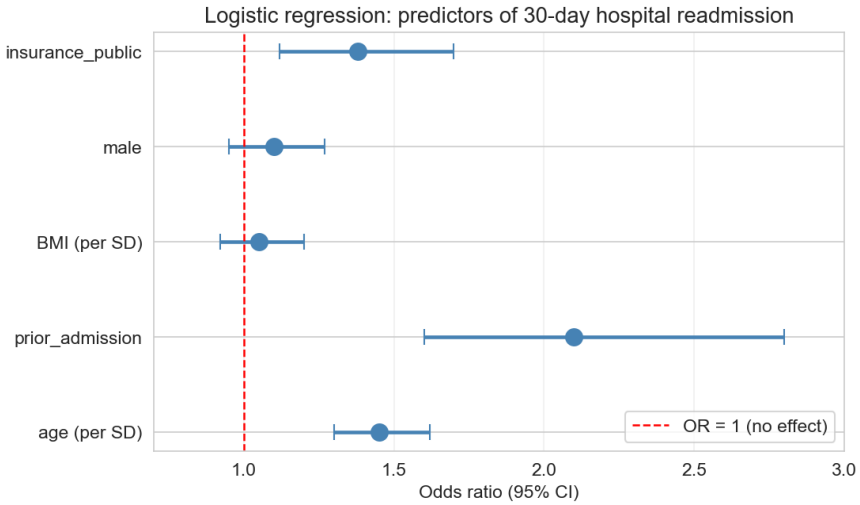
(a) Which **LINE** assumption is failing? What does the fan shape mean substantively about how the model fits at low vs. high price levels? (1–2 sentences.)

(b) Name **one fix** from the lecture for this issue.

Solution

(a) E (equal variance) is violated — the plot shows **heteroscedasticity** (the spread of residuals fans out as fitted price grows). Substantively, **cheap homes vary in price by a few thousand dollars; expensive homes vary by tens or hundreds of thousands**. The constant-variance assumption underlying the formula-based CIs is wrong: prediction intervals will be too wide for cheap homes and too narrow for expensive ones.
(b) Log-transform the response (fit `log(price)` instead of `price`). Log compresses high values more than low ones, so the residual variance is stabilized across fitted values. (Other accepted answers: *bootstrap CIs* — not assumption-bound; *robust SEs* — adjust the SE formula to remain valid under heteroscedasticity.)

Question 5. A hospital fits a logistic regression to predict 30-day readmission. The forest plot of odds ratios with 95% CIs:



(a) Which predictors are **statistically distinguishable from "no effect"** at the 95% level? What is the criterion on the CI for an odds ratio? (1–2 sentences.)

(b) Interpret the **age** odds ratio in plain English. (`age` is on a per-SD scale; 1 SD of age is about 12 years in this cohort.)

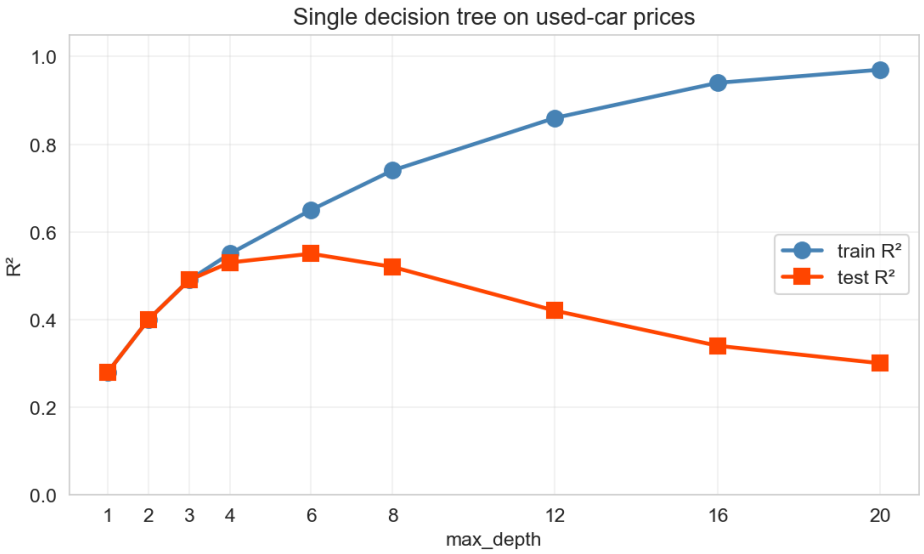
Solution

(a) A predictor is statistically distinguishable from "no effect" if its 95% CI for the odds ratio **excludes 1** (the null value for OR; equivalent to $\beta = 0$ on the log-odds scale). From the plot:

- **age** (CI roughly [1.30, 1.62]) — significant
- **prior_admission** (CI [1.60, 2.80]) — significant
- **insurance_public** (CI [1.12, 1.70]) — significant
- **BMI** (CI [0.92, 1.20]) — NOT significant (crosses 1)
- **male** (CI [0.95, 1.27]) — NOT significant (crosses 1)

(b) Holding the other predictors fixed, **a one-SD increase in age (about 12 years) multiplies the odds of readmission by about 1.45** — a 45% increase in odds per 12 years of age. (Equivalently: going from age 60 to age 72 raises the odds of 30-day readmission by about 45%.)

Question 6. A used-car dealer fits decision trees of varying depth to predict car prices. Train and test R^2 as `max_depth` grows:



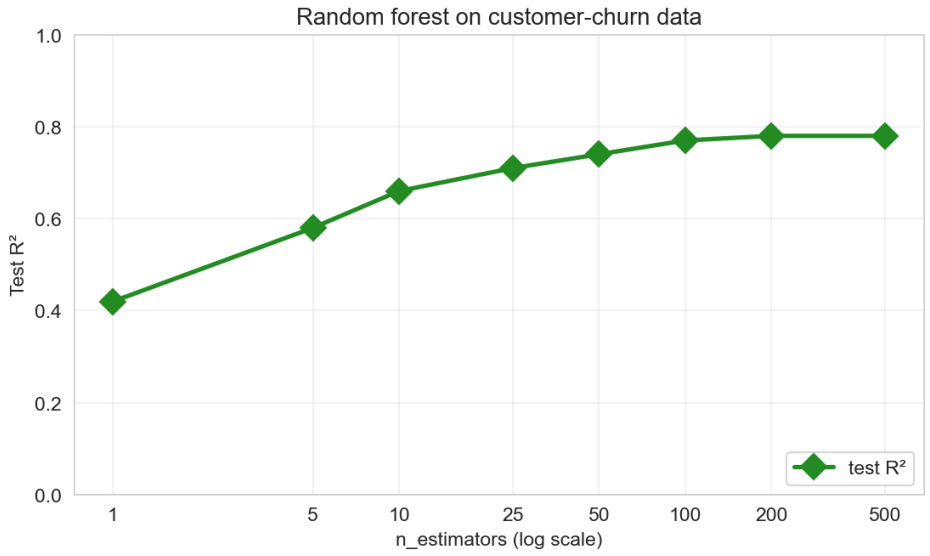
(a) At `max_depth = 20`, the train R^2 is ≈ 0.97 but the test R^2 is ≈ 0.30. Name the phenomenon. In 1 sentence, what has the tree learned at depth 20?

(b) A teammate proposes picking the `max_depth` that **maximizes train R^2**. Why is this a bad idea? What should they do instead?

Solution

(a) Overfitting. At `max_depth = 20`, every leaf holds a tiny subset of training rows; the tree has **memorized the specific training observations and their idiosyncratic noise**, fitting the training data almost perfectly but capturing nothing that generalizes.
(b) Maximizing **train** R^2 always picks the deepest tree (more depth $\Rightarrow$ more flexibility $\Rightarrow$ more training fit), but as the test curve shows, that's exactly the worst place to be on the test set. Instead, pick depth by **cross-validation**: sweep candidate depths, score each by 5-fold CV on the training data, pick the depth with the highest CV score. The test set stays clean for a final honest evaluation.

Question 7. A SaaS company fits random forests of varying size to predict customer churn:



(a) From the curve, what is the test R^2 **plateau** approximately? Name **two distinct sources of randomness** that distinguish the individual trees in a random forest from each other (i.e., what makes their predictions disagree). (1–2 sentences total.)

(b) In 1–2 sentences, explain why adding more trees doesn't push the test R^2 to 1.

Solution

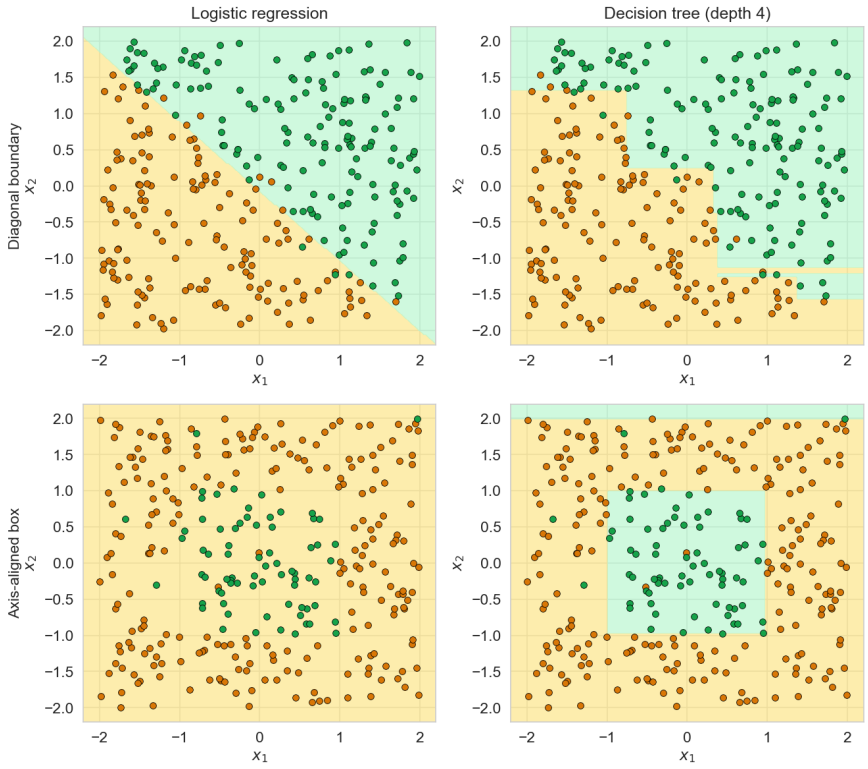
(a) Plateau at test $R^2 \approx$ **0.78**. The two sources of randomness:

1. **Bootstrap samples of rows.** Each tree trains on a different random sample of rows (drawn with replacement from the original training data) — so each tree sees a different ~63% of distinct rows.
2. **Feature subsampling at each split.** At each internal node, the tree considers only a random subset of features when picking the best split — forcing trees to find different splits even when given the same data.

Both decorrelate the trees so that averaging buys variance reduction.

(b) Adding more trees only **reduces variance, not bias**. The plateau reflects (i) the **irreducible noise** in the data (features don't perfectly determine the outcome), and (ii) the **average correlation between trees** — when trees make similar mistakes (which they will, because they share training data), averaging can only cancel out the *uncorrelated* part of the noise. No amount of additional trees can erase these two floors.

Question 8. Two synthetic binary classification problems, each fit with logistic regression and a depth-4 decision tree. Yellow points are class 0; green are class 1.

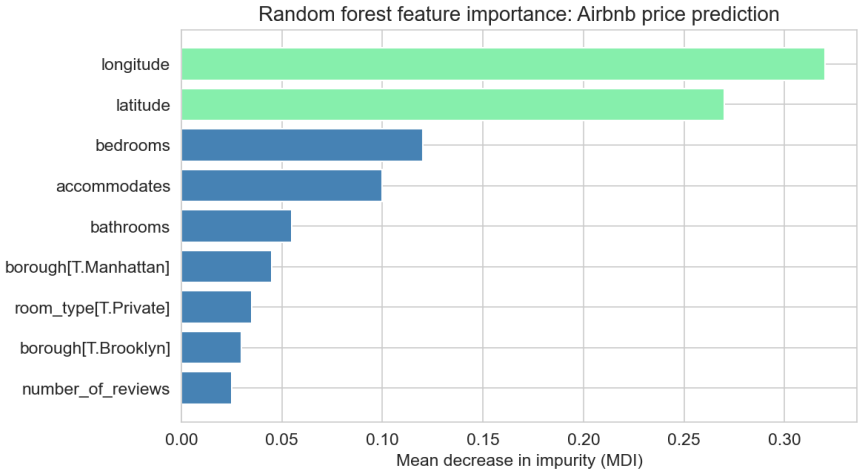


For each row (**diagonal boundary**, top; **axis-aligned box**, bottom), which model fits **better**, and why in 1 sentence each?

Solution

Top row (diagonal): Logistic regression wins. Its single linear boundary cuts cleanly along the diagonal; the depth-4 tree has to approximate the diagonal with a staircase of axis-aligned cuts and misclassifies points near the boundary.
Bottom row (axis-aligned box): Decision tree wins. A box is defined by four axis-aligned cuts, which trees handle natively; logistic regression's single hyperplane cannot enclose a region (no single line separates "inside box" from "outside box") — it picks an arbitrary diagonal and is wrong on most of the corners.
The lesson: the **right model is a property of the data**, not a universal ranking. Linear models excel on smooth, monotone effects; trees excel on corners, thresholds, and interactions. On real data you don't know the geometry — fit both and compare on held-out data.

Question 9. A random forest predicts **Airbnb nightly price** in NYC. The feature importance (MDI) bar chart:



A real-estate blog reads this chart and writes: *"To raise an apartment's price, the most effective interventions are (1) move it to a different latitude/longitude, and (2) add bedrooms."*

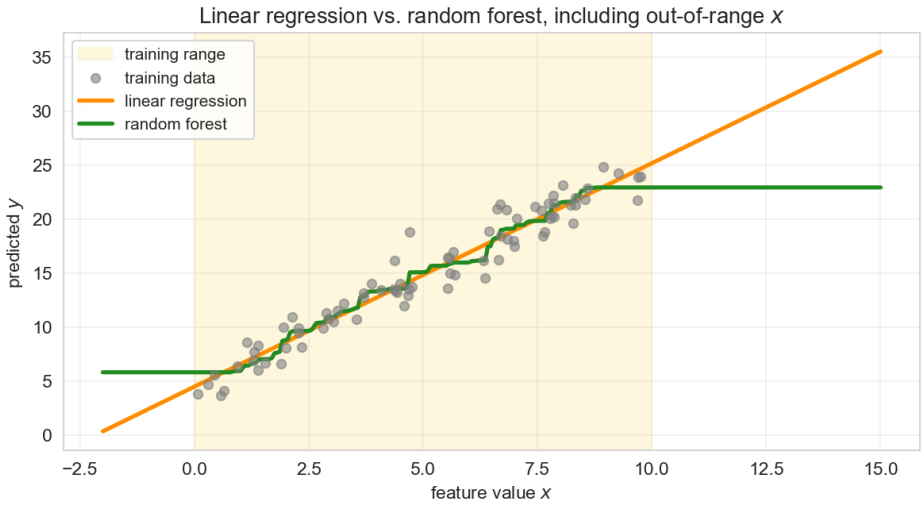
(a) What's wrong with the blog's interpretation? In 2 sentences, what does high MDI actually mean?

(b) Suppose a host genuinely can add a bedroom (it's intervenable). Does the MDI chart tell them how much price will go up? What tool from this course actually answers that question?

Solution

(a) MDI is descriptive, not causal. High MDI on a feature means the forest **uses that feature for splitting frequently and effectively** — the splits drop impurity a lot, *given the other features*. It does **not** mean intervening on the feature would change the outcome. ("Move your apartment" is also nonsense in the controllable sense — but even setting that aside, MDI just reflects model usage, not real-world causal effects.)
(b) No, MDI doesn't tell the host the *magnitude* of price change from adding a bedroom. The tool that does is a **regression coefficient** (with appropriate controls): the coefficient on `bedrooms` from a multiple regression of price tells you, *controlling for the other features in the model*, how much the predicted price changes per added bedroom. Even this is an *association*, not a guaranteed causal effect — a properly randomized intervention (the gold standard of Ch 18) would be required to settle causality, since adding a bedroom usually changes other things (square footage, building stock, layout) simultaneously.

Question 10. A small bank trains a linear regression and a random forest to predict default rates as a function of one feature, `loan_size`. Training data lies in $x \in [0, 10]$ (in \$100K units, so loans of \$0–\$1M). Both models still produce predictions for inputs outside that range. The plot:



(a) For $x = 15$ (a \$1.5M loan, outside training range), the **random forest** outputs a flat prediction near 23. In 1–2 sentences, explain what the forest's prediction *is* (in terms of training data) and why it is **flat** outside the training range.

(b) Suppose the true underlying relationship really is linear and **continues at the same slope** past $x = 10$. Which model's extrapolation at $x = 15$ is approximately right? Now suppose the true relationship **bends downward** past $x = 10$ (e.g., very large loans default at a lower rate because they go through extra underwriting). Which model is closer to the truth then? (1–2 sentences.)

Solution

(a) The forest's prediction at $x = 15$ is the **average training-y inside whichever leaf the input routes into** — here, the leaf containing the **largest training values** (x near 10). Since no training row has $x > 10$, every input with $x > 10$ routes to the same boundary leaf, so the forest's prediction is **constant** out there: it has no information about what happens at $x = 15$, and its prediction is bounded by the maximum training y.
(b) If the true relationship is **linear past** $x = 10$, the **linear model's extrapolation is approximately right** (it extends the fitted slope); the forest's flat prediction misses by a growing amount as x grows.
If the true relationship **bends downward past** $x = 10$, the linear model **overshoots** (extends a slope that no longer applies) and the forest **undershoots** (flatlines at a value calibrated on the training range). **Neither is right**, but the forest happens to be closer if the true value at $x = 15$ is near the training-range maximum.
The deeper lesson: **forests cannot extrapolate** (their predictions are bounded by training outcomes), while **linear models extrapolate as if their fitted relationship continues forever**. Both are risky out-of-distribution; both call for distribution-shift monitoring and additional data before trusting the prediction.

Question 11. A classification tree is deciding whether to split a node containing 100 training examples: **60 customers who churned** (class 1) and **40 who stayed** (class 0). A candidate split on `tenure < 6 months` would partition the node into:

- **Left child** (`tenure < 6 months`): 50 churned, 10 stayed (60 customers)
- **Right child** (`tenure  6 months`): 10 churned, 30 stayed (40 customers)

Recall the **Gini impurity** for a binary node with class proportions $\hat{p}$ and $1 - \hat{p}$:

$$\text{Gini} = 2\hat{p}(1 - \hat{p}).$$

(a) Compute the Gini impurity of the **parent** node and of each **child** node. Show the arithmetic.

(b) What Gini value indicates a **perfectly pure** node? Compute the **sample-weighted average** of the two children's Gini values. Does this split reduce impurity below the parent?

Solution

(a) - **Parent** (60/100 = 0.6 churned, 0.4 stayed): Gini $= 2 \times 0.6 \times 0.4 =$ **0.48**. - **Left child** (50/60 0.833 churned, 10/60 0.167 stayed): Gini $= 2 \times \frac{50}{60} \times \frac{10}{60} = \frac{1000}{3600} \approx$ **0.278**. - **Right child** (10/40 = 0.25 churned, 30/40 = 0.75 stayed): Gini $= 2 \times 0.25 \times 0.75 =$ **0.375**.
(b) A **perfectly pure node has Gini = 0** (one class only, $\hat{p} \in \{0, 1\}$). The maximum for a binary node is 0.5 (a 50/50 split).
Sample-weighted average of the children's Gini values, weighting each child by its fraction of the parent's samples:

$$\tfrac{60}{100} \times 0.278 + \tfrac{40}{100} \times 0.375 \approx 0.167 + 0.150 = \mathbf{0.317}.$$

The split **reduces impurity** from 0.48 (parent) to 0.317 (weighted child average), an impurity drop of about 0.16. This is the splitting criterion the tree maximizes at each node: pick the (feature, threshold) pair that drops weighted child impurity the most.