Binary & Categorical Predictors/Targets
Regression, Logistic Regression, and Trees

Learning Goals

By the end, you should be able to:

- Encode and interpret binary and categorical predictors in linear and logistic models.
- Model binary and multi-class targets (logistic and multinomial logistic regression).
- Build and interpret regression trees (continuous Y) and classification trees (categorical Y).
- Translate between coefficients/odds ratios and decision-tree rules.
- Choose metrics and thresholds appropriate to the task.

The Four Building Blocks

1. **Binary predictor** ($X \in \{0, 1\}$)
2. **Categorical predictor** ($K > 2$ levels)
3. **Binary target** ($Y \in \{0, 1\}$)
4. **Categorical (multi-class) target** ($Y \in \{1, \ldots, K\}$)

We will pair these with: linear regression, logistic regression, multinomial logistic regression, regression trees, and classification trees.

Linear Regression with a Binary Predictor (Expanded)

Model:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i, \quad X_i \in \{0, 1\}.$$

Interpretation:

$$\beta_1 = \mathbb{E}[Y|X=1] - \mathbb{E}[Y|X=0].$$

- β_0 — mean of Y for the baseline group ($X = 0$).
- β_1 — difference in mean Y between groups ($X = 1$ vs. $X = 0$).
- Equivalent to a two-sample mean difference (under equal variance).
- When adding other predictors, $Y = \beta_0 + \beta_1 X + \beta_2 Z + \cdots$:
  - β_1 is the adjusted group difference, holding Z constant.
  - β_2 measures the change in Y per unit of Z, holding X constant.

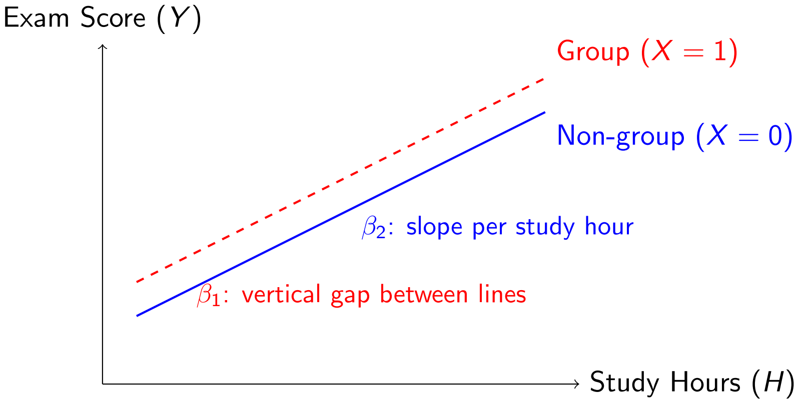
Worked Example: Interpreting Coefficients

Scenario: Exam score (Y) depends on study group membership (X) and hours studied (H):

$$Y = \beta_0 + \beta_1 X + \beta_2 H + \varepsilon.$$

- β_0: average score for non-group students ($X = 0$) who study 0 hours.
- β_1: expected difference between group and non-group students *at the same number of hours*.
- β_2: expected increase in score for each additional hour studied, holding group membership fixed.
- Example fit: $\hat{Y} = 65 + 5X + 3H$.
  - A student not in a group who studies 10 hours: $\hat{Y} = 65 + 3(10) = 95$.
  - A group member who studies 10 hours: $\hat{Y} = 65 + 5 + 3(10) = 100$.

Visualization: Binary + Continuous Predictor



- Two roughly parallel lines — same slope (β_2), different intercepts (β_1 shift).
- β_1 shows the average gap between groups at equal study time.
- β_2 shows the effect of more study hours within each group.

Tree View (Regression Tree)

- A single split on X minimizes MSE and yields two leaves:
  - Leaf for $X = 0$: predicts $\hat{Y}_0 = 70$.
  - Leaf for $X = 1$: predicts $\hat{Y}_1 = 80.7$.
- With multiple predictors, the tree may split first on H (study hours) if it explains more variance, and then on X.
- **Equivalence:** With one binary split and leaf means, the regression tree reproduces group-wise averages.

Quick Check A

1. In $Y = \beta_0 + \beta_1 X$, what are $\hat{Y}$ at $X = 0$ and $X = 1$? What is their difference?
2. In $Y = \beta_0 + \beta_1 X + \beta_2 H$, how do you interpret β_1 and β_2?
3. How many times can you split on a binary predictor variable in a regression tree?

Business Scenario: Regional Sales and AdSpend

A company operates in three regions — **West**, **North**, and **East**. They want to understand how both **region** and **advertising spending** affect monthly sales.

- Y = Monthly sales revenue (\$ thousands)
- X_1 = Advertising spend (\$ thousands)
- C = Region (*West*, *North*, *East*)

Goal: Estimate how much sales differ by region after controlling for ad spending.

Regression Model with Categorical Predictors

We model:

$$Y = \beta_0 + \beta_1 \text{AdSpend} + \beta_2 \, \mathbb{1}\{C = \text{North}\} + \beta_3 \, \mathbb{1}\{C = \text{East}\} + \varepsilon$$

Interpretation of coefficients:

- β_0: average sales in the **West** region when AdSpend $= 0$ (base level).
- β_1: expected change in sales for each \$1k ad spend increase, holding region fixed.
- β_2: expected North vs West difference, adjusting for ad spend.
- β_3: expected East vs West difference, adjusting for ad spend.

How One-Hot Encoding Works

A regression model requires numeric inputs. For a categorical variable with K levels, we include $K-1$ binary indicators.

Region	D_{North}	D_{East}
West (base)	0	0
North	1	0
East	0	1

- West is the base category (absorbed in β_0).
- β_2 and β_3 compare other regions to that base.

Worked Example: Data

Monthly sales (Y, $k) by region ($C$) and ad spend ($X_1$, $k):

Obs	Region	AdSpend	Sales
1	West	10	40
2	West	8	37
3	North	10	46
4	East	9	44
5	North	7	39

We fit the model:

$$\hat{Y} = 30 + 2.5\,\text{AdSpend} + 5.0\,D_{\text{North}} + 3.0\,D_{\text{East}}$$

Interpreting the Coefficients

- $\hat{\beta}_0 = 30$: baseline sales in West with AdSpend $= 0$.
- $\hat{\beta}_1 = 2.5$: every \$1k ad spend increases sales by about \$2.5k, holding region fixed
- $\hat{\beta}_2 = 5.0$: North averages \$5k more than West, at the same AdSpend.
- $\hat{\beta}_3 = 3.0$: East averages \$3k more than West, at the same AdSpend.

Interpretation: Regional differences shift the intercepts; ad spending raises sales equally across all regions.

Model by Region

Substitute dummy values to get three region-specific lines:

$$\text{West:}\;\; Y = 30 + 2.5(\text{AdSpend})$$

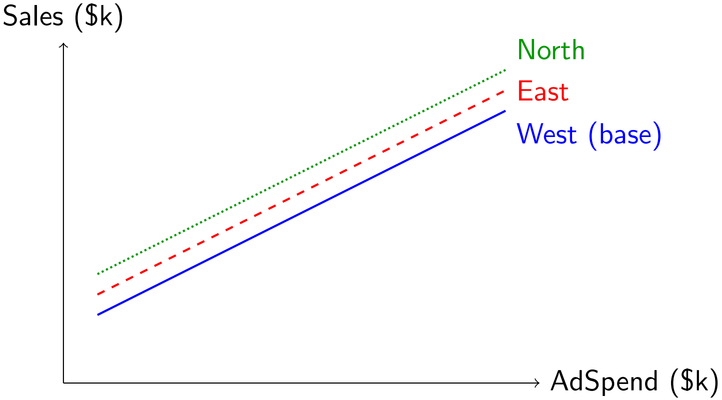
$$\text{North:}\;\; Y = 35 + 2.5(\text{AdSpend})$$

$$\text{East:}\;\; Y = 33 + 2.5(\text{AdSpend})$$

Interpretation:

- All regions have the same slope (common ad response).
- North has highest intercept, East moderate, West lowest.
- Parallel lines = same marginal effect of AdSpend.

Visualizing Regional Differences



Interpretation:

- Equal slopes (same β_1) $\rightarrow$ same return on ad spend.
- Different intercepts $\rightarrow$ persistent regional advantages.

Tree View (Regression Tree)

- A regression tree splits on variables that most reduce error.
- It may first split on Region if regional mean differences are large.
- If ad spend drives more variation, it may split on AdSpend first.
- **Interpretation:** Trees reveal natural segmentations (e.g., "North behaves differently").

Quick Check B

1. If West is the base, how do you interpret a *negative* β_{East}?
2. Why might a regression tree split on Region=North first, even if OLS shows East's coefficient is larger?
3. Can a regression tree ever split on the same category twice?

When the Target is Binary

When our target Y only takes two values (e.g., $0 =$ No, $1 =$ Yes), linear regression no longer fits well.

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

- Can predict values outside 0–1 $\rightarrow$ not valid probabilities.
- The effect of X is usually not constant — it may change as probability approaches 0 or 1.

We need a model that:

- Predicts probabilities between 0 and 1.
- Captures how small changes in X have different effects at different probability levels.

The Logistic Regression Model

The logistic regression model gives probabilities directly:

$$p(x) = \Pr(Y = 1 \mid X = x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots)}}$$

- Output $p(x)$ is always between 0 and 1.
- If $p > 0.5$, we often predict "Yes"; if $p < 0.5$, we predict "No."
- The model fits an S-shaped curve rather than a straight line.

Interpretation:

- A positive coefficient ($\beta_j > 0$) makes $Y = 1$ more likely — increases the predicted probability.
- A negative coefficient ($\beta_j < 0$) makes $Y = 1$ less likely — decreases the predicted probability.

From Probability to Logit: Making Logistic Look Linear

We can write logistic regression as:

$$p(x) = \Pr(Y = 1|X = x) = \frac{1}{1 + e^{-(\beta_0+\beta_1 X)}}$$

This can be rearranged into a linear form using the **logit** function:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X$$

Interpretation:

- The logit is the **log of the odds** of $Y = 1$.
- The model is linear in the logits, not in the probabilities.
- Positive β_1 increases the odds (and probability) of $Y = 1$.
- Negative β_1 decreases the odds (and probability) of $Y = 1$.

Reminder: Because of the curve's shape, a one-unit change in X changes the *log-odds* by β_1, but the probability change depends on where we start.

Example 1: Customer Renewal Model

$$\text{logit}(p) = \beta_0 + 0.8(\text{Received Reminder})$$

Interpretation:

- A positive coefficient (+0.8) means receiving a reminder increases the likelihood of renewal.
- However, this does **not** mean the probability rises by 0.8.

Illustration:

$$\text{If no reminder: } \beta_0 = -0.4 \Rightarrow p_0 = \frac{1}{1+e^{0.4}} = 0.40,$$

$$\text{With reminder: } \beta_0 + 0.8 = 0.4 \Rightarrow p_1 = \frac{1}{1+e^{-0.4}} = 0.60.$$

Result: A +0.8 change in the coefficient increases the probability by only 0.20 (from $0.40 \rightarrow 0.60$).

Key takeaway: Logistic regression effects are *nonlinear in probability*.

Example 2: Credit Approval

$$\text{logit}(p) = \beta_0 + 0.03(\text{Credit Score}) - 0.02(\text{Debt Ratio})$$

Interpretation:

- Higher **credit score** (positive coefficient) increases approval probability.
- Higher **debt ratio** (negative coefficient) decreases approval probability.
- The effects are **additive on the logit scale** — they shift the curve left or right.

Example values:

$$\text{Low credit (600)} : \eta = -1.0 \Rightarrow p = 0.27,$$

$$\text{High credit (700)} : \eta = -1.0 + 0.03(100) = 2.0 \Rightarrow p = 0.88.$$

Insight: Even small slopes like 0.03 per 1-point increase can have large cumulative effects over realistic ranges.

Example 3: Employee Attrition

$$\text{logit}(p) = \beta_0 - 1.2(\text{Job Satisfaction}) + 0.5(\text{Overtime})$$

Interpretation:

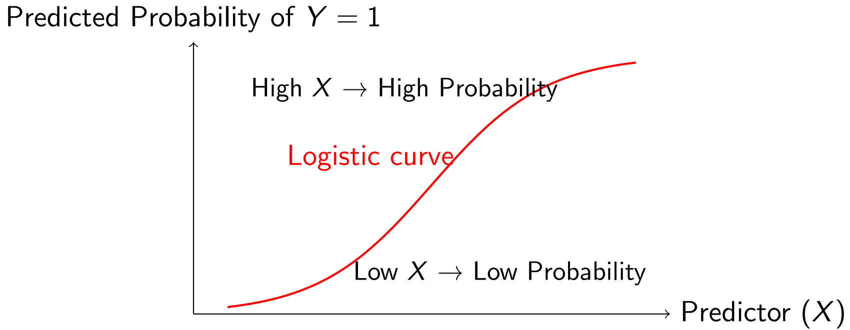
- **Job Satisfaction (–1.2):** Higher satisfaction reduces the log-odds of leaving — employees with higher scores are less likely to leave.
- **Overtime (+0.5):** Working overtime increases the likelihood of leaving.
- **Intercept (β_0):** Baseline log-odds of leaving for a non-overtime worker with average satisfaction.

Example Calculation:

$$\text{Baseline: } \beta_0 = 0 \Rightarrow p_0 = 0.5,$$

$$\text{Satisfied (JobSat=2): } \eta = 0 - 1.2(2) = -2.4 \Rightarrow p = 0.08,$$

Visualizing Logistic Regression



Interpretation:

- As X increases, $p(Y = 1)$ increases — but never exceeds 1.
- The middle region (around 0.5) shows the strongest effect of X.
- We can apply a cutoff (often 0.5) to make a Yes/No prediction.

How It Differs from Linear Regression

Linear regression:

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

- Continuous outcome (any numeric value).
- β_1 = constant change in Y per 1-unit change in X.

Logistic regression:

$$\Pr(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

- Binary outcome (0 or 1).
- β_1 changes probability nonlinearly.
- Effect size depends on current probability.
- Produces probabilities, not continuous values.

Big idea: Linear regression predicts *how much*; logistic regression predicts *how likely*.

Logistic Regression vs Classification Trees

Logistic Regression

- Produces a smooth probability curve.
- Every variable affects the probability continuously.
- Positive coefficients $\Rightarrow$ more likely; negative coefficients $\Rightarrow$ less likely.

Classification Trees

- Divide the data using simple rules (e.g., Age $< 30 \rightarrow$ "No").
- Each split aims to create purer subgroups — groups that are mostly Yes or mostly No.
- Final groups ("leaves") each get a constant predicted probability.

Summary:

- Logistic: smooth and continuous; assumes one global curve.
- Tree: segmented and piecewise; learns separate local predictions.

Interpreting Terminal Nodes (Leaves)

Each terminal node has two key outputs:

1. **Predicted class:** whichever outcome (0 or 1) is most common in that leaf.
2. **Predicted probability:** proportion of 1's in that leaf.

Example:

If a leaf has 90 "Yes" and 10 "No," $p = 0.90 \Rightarrow 90\%$ probability Yes.

Interpretation:

- Logistic regression gives one smooth curve of probabilities across X.
- A tree gives several flat "steps" — one probability per group.
- Each leaf acts like a mini average for similar cases.

Target Functions: Regression vs Classification Trees

You've already seen **regression trees** that minimize the sum of squared errors (SSE):

$$\text{Split chosen to minimize } \sum (Y_i - \hat{Y}_{\text{node}})^2$$

For **classification trees**, the goal is different:

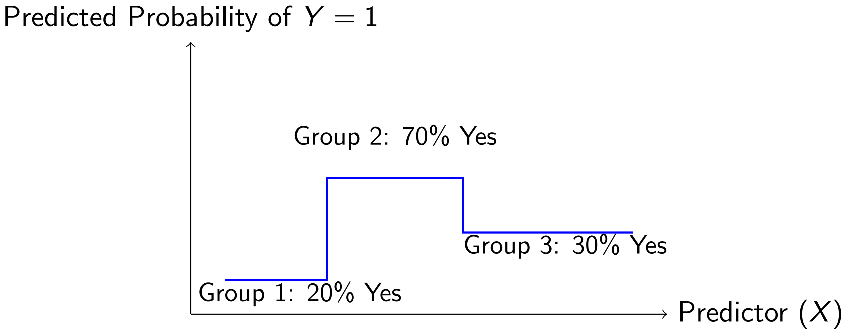
Split chosen to minimize impurity:

- Using Gini: $2p(1-p)$ — smaller = purer.
- Using Entropy: $-p\log(p) - (1-p)\log(1-p)$ — smaller = purer.

Key distinction:

- Regression tree $\rightarrow$ predicts *average numeric value*.
- Classification tree $\rightarrow$ predicts *probability or majority class*.

Visual Example: Stepwise Predictions



Interpretation:

- Each "step" represents a terminal node's predicted probability.
- Here, probabilities rise from Group 1 $\rightarrow$ Group 2, then fall again in Group 3.
- Trees naturally allow nonlinear, non-monotonic patterns through successive splits.

Quick Check: Trees and Logistic Models

1. What does the predicted value at a terminal node represent?
2. How does a classification tree's objective (Gini/Entropy) differ from regression tree's SSE?
3. Why might a business prefer a tree model to a logistic regression?

Quick Check C

1. What does a positive coefficient tell us about the predictor's effect on $Y = 1$?
2. What about a negative coefficient?
3. In logistic regression, why might we use a cutoff like 0.5 to make a Yes/No decision?
4. When might a tree model be easier to explain to a client or manager?

When the Target Has More Than Two Categories

So far, we have looked at binary outcomes (0/1 or Yes/No). What if the target Y can take on several categories?

- Examples:
    - Loan decision: *Deny, Hold, Approve*
    - Customer status: *New, Active, Churned*
    - Product choice: *Economy, Premium, Luxury*
- We now want to model $\Pr(Y = k \mid X)$ for each possible class k.

Two major approaches:

1. Multinomial logistic regression — an extension of logistic regression.
2. Classification trees — a rule-based model that handles multiple classes naturally.

Multinomial Logistic Regression: The Idea

For K outcome categories, we compare each class k to a chosen **base class** K:

$$\log \frac{\Pr(Y = k)}{\Pr(Y = K)} = \alpha_k + \beta_k^\top X, \quad k = 1, \ldots, K - 1.$$

I want you to know it can be done, but we won't be doing it for this class.

Classification Trees for Multiple Classes

Idea: Trees extend naturally to more than two classes.

- At each split, the algorithm looks for the variable and cutoff that best separate the classes.
- The same impurity measures (Gini, entropy) are used, but now computed across all K categories:

$$\text{Gini} = 1 - \sum_{k=1}^{K} p_k^2, \quad \text{Entropy} = -\sum_{k=1}^{K} p_k \log(p_k).$$

- The chosen split is the one that most reduces impurity (i.e., makes child nodes more homogeneous).

Advantage: Trees handle multi-class outcomes automatically — you don't have to specify a base class or multiple equations.

Terminal Nodes in Multi-Class Trees

Each **terminal node (leaf)** represents a subgroup of observations.

For each node, the tree reports:

1. The most common class (the predicted label).
2. The estimated probability for each class, based on relative frequencies.

Example:

$$\text{Node: } \begin{cases} \text{Approve: } 60\% \\ \text{Hold: } 25\% \\ \text{Deny: } 15\% \end{cases} \quad \Rightarrow \text{Predicted class: Approve, } p(\text{Approve}) = 0.6$$

Interpretation:

- Every terminal node behaves like a localized conditional probability model.
- Logistic regression provides one global equation; trees provide many small “local” models.