LOGISTIC REGRESSION

Theory and a Fully Worked Numerical

1. Why Not Linear Regression?

We often need to predict a **categorical** outcome — Yes/No, 0/1 — rather than a number. Linear regression is unsuitable here for two structural reasons, not just a cosmetic one:

- **Unbounded output:** a straight line can output any real number, but a probability must lie in [0, 1]. Nothing in linear regression enforces this boundary.
- **Wrong error structure:** linear regression assumes errors are normally distributed with constant variance. For a binary outcome the true variance is P(X)(1−P(X)) — it changes with X, so this assumption fails by construction:

$$\mathrm{Var}(Y \mid X) = P(X)\big(1 - P(X)\big)$$

The fix is not to clip the line, but to model a **transformed**, unbounded version of P instead — which is exactly what logistic regression does.

2. The Sigmoid Function

Logistic regression first computes an ordinary linear score Z, then squashes it into [0, 1] using the **sigmoid function**:

$$P(Y=1) = \frac{1}{1+e^{-Z}}, \qquad Z = b_0 + b_1x_1 + b_2x_2 + \cdots + b_nx_n$$

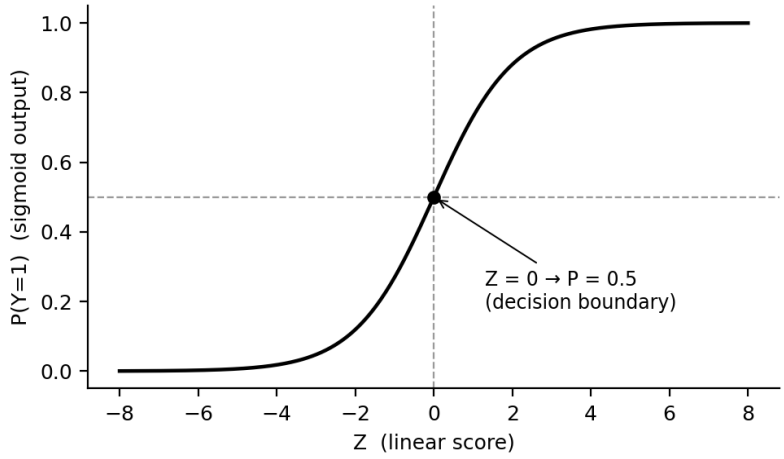

Figure 1. The sigmoid curve. Large positive Z → P → 1; large negative Z → P → 0; Z = 0 → P = 0.5 (the decision boundary).

3. Why "Logistic" — the Log-Odds Connection

Odds convert a probability into a different, more familiar quantity:

$$\mathrm{Odds} = \frac{P}{1-P}$$

Taking the natural log of the odds gives the **logit**, which ranges over all real numbers (−∞ to +∞) — exactly the range a linear equation naturally produces:

$$\ln\left(\frac{P}{1-P}\right) = Z = b_0 + b_1x_1 + \cdots + b_nx_n$$

This is the precise sense in which logistic regression is "linear": linearity holds in **log-odds space**, not in probability space. "Logistic regression" literally means "regression on the logit."

Interpreting a coefficient

b_i is the change in log-odds per unit increase in x_i. Exponentiating gives the more intuitive **odds ratio**:

$$\text{Odds ratio} = e^{b_i}$$

Example: if $b_i = 0.7$ for "has a prior purchase," then e^0.7 ≈ 2.01 — a prior purchase **doubles** the odds of buying again.

4. How It Is Fit — Maximum Likelihood Estimation

Logistic regression cannot use least squares (no closed-form solution exists here). Instead it uses **Maximum Likelihood Estimation (MLE)**: find the coefficients that make the *observed* 0/1 outcomes as probable as possible.

For one observation, this probability is written compactly using an exponent trick:

$$P_i^{y_i}(1-P_i)^{1-y_i}$$

This equals P_i when $y_i=1$, and $(1-P_i)$ when $y_i=0$ — one formula for both cases. Multiplying across all n observations (assumed independent) gives the **likelihood function**:

$$L(b_0, b_1, \ldots) = \prod_{i=1}^{n} P_i^{y_i}(1-P_i)^{1-y_i}$$

In practice we maximise the **log-likelihood** instead (same maximiser, but numerically stable):

$$\ell(b) = \ln L = \sum_{i=1}^{n}\Big[y_i \ln P_i + (1-y_i)\ln(1-P_i)\Big]$$

Its negative is exactly the **cross-entropy loss** used in ML libraries — minimising cross-entropy = maximising log-likelihood. There is no closed-form solution, so the optimum is found **iteratively** (gradient ascent, or Newton–Raphson / IRLS for faster convergence) — this is what the *solver* parameter selects in scikit-learn's LogisticRegression. When classes are not perfectly separable, the log-likelihood is **concave**, guaranteeing a single global maximum.

5. Fully Worked Numerical Example

Problem. Five students studied for X = 1, 2, 3, 4, 5 hours; Y = 1 if they passed, 0 if not: Y = (0, 0, 1, 1, 1). Compare two candidate coefficient sets by their log-likelihood, and interpret the better model.

Hours (X)	**1**	**2**	**3**	**4**	**5**
Pass (Y)	0	0	1	1	1

Step 1 — Compute P for candidate 1: $b_o = -3.0$, $b_1 = 1.2$

For X = 3:

$$Z = -3.0 + 1.2\,(3) = 0.6$$

$$P = \frac{1}{1+e^{-0.6}} = \frac{1}{1+0.5488} = 0.6457$$

X	**1**	**2**	**3**	**4**	**5**
Z	−1.8	−0.6	0.6	1.8	3.0
P = sigmoid(Z)	0.1419	0.3543	0.6457	0.8581	0.9526
Y (actual)	0	0	1	1	1

Step 2 — Log-likelihood of candidate 1

$$\ell_1 = \sum\Big[y_i \ln P_i + (1-y_i)\ln(1-P_i)\Big] = -1.2295$$

Step 3 — Try candidate 2: $b_o = -4.0$, $b_1 = 1.6$

Repeating Steps 1–2 with the new coefficients gives a higher log-likelihood:

$$\ell_2 = -0.9340 \qquad \text{(improved coefficients: better fit)}$$

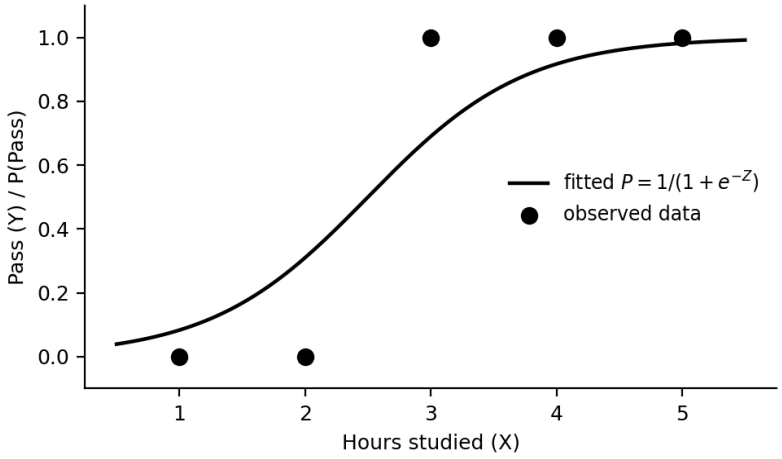

Figure 2. The fitted sigmoid curve for candidate 2 (b_0=−4.0, b_1=1.6) against the observed pass/fail data.

Conclusion: since ℓ_2 = −0.9340 > ℓ_1 = −1.2295, candidate 2 explains the data better — MLE would continue this search (via gradient ascent) until the log-likelihood stops improving.

6. A Realistic Prediction — Multiple Predictors

A retailer models purchase propensity from monthly spend (x_1, in $) and prior purchase ($x_2$, 1 = yes). Fitted model: b_0 = −6.0, b_1 = 0.05, b_2 = 0.9. For a customer with spend = 80 and a prior purchase:

$$Z = -6.0 + 0.05(80) + 0.9(1) = -1.1$$

$$P(Y = 1) = \frac{1}{1 + e^{1.1}} = 0.2497$$

Odds-ratio interpretation of b_2:

$$e^{0.9} = 2.46 \quad \text{(prior purchase roughly } 2.5\times \text{ the odds)}$$

7. Summary

- Linear regression fails for classification: unbounded output, wrong error assumptions for a binary Y.
- Logistic regression models log-odds as linear in the predictors, then converts back to P via the sigmoid.
- Coefficients are interpreted through odds ratios (e^{b_i}), not raw units.
- Fitting uses Maximum Likelihood Estimation, solved iteratively — not least squares.
- Z = 0 (P = 0.5) defines a linear decision boundary — logistic regression is a linear classifier.