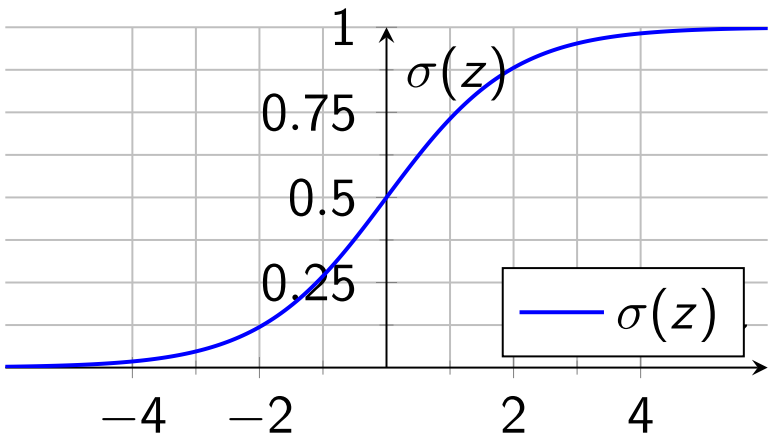
Motivation: Probabilistic Binary Decisions

- Binary label with probability $p(y = 1 \mid x)$, not just a hard class.
- Score $z = b + w^\top x$ is unbounded; we need a link to map it to $[0, 1]$.
- **Score vs probability**: z only ranks samples; $\sigma(z)$ is calibrated.
- Example: $z = 0.2 \Rightarrow p \approx 0.55$; $z = -2 \Rightarrow p \approx 0.12$.

Sigmoid Link and Geometry

- Sigmoid: $\sigma(z) = \frac{1}{1+e^{-z}}$ squashes any score to $[0, 1]$.
- $z = b + w^\top x$ is the signed distance from the decision hyperplane (scaled by $\|w\|$).
- Slope is steepest at $z = 0$; samples near the boundary are most uncertain.
- Tiny numeric check:
  $x = (1, 2), w = (0.3, 0.4), b = -0.5 \Rightarrow z = 0.6, p \approx 0.65$.



Log-Odds Are Linear

- Logit transform: $\log \frac{p}{1-p} = b + w^\top x$.
- Log-odds affine in $x \Rightarrow$ boundary $w^\top x + b = 0$ is a hyperplane (linear model).
- Monotonicity: ranking by z or by p is identical.

Likelihood for Bernoulli Labels

- For i.i.d. samples, $p_i = \sigma(b + w^\top x_i)$.
- Likelihood:
$$\mathcal{L}(w, b) = \prod_i p_i^{y_i} (1 - p_i)^{(1-y_i)}.$$
- Maximizing $\mathcal{L}$ chooses w, b that make observed labels most probable.

MLE and Log-Loss

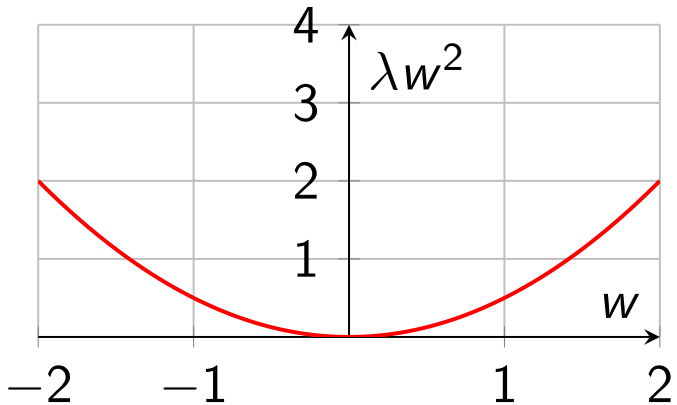
- Negative log-likelihood:

$$\text{NLL} = -\sum_i \left[y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \right].$$

- Minimizing NLL is identical to minimizing binary cross-entropy (log-loss).
- Interpretation: confident wrong predictions are heavily penalized; uncertain ones less so.

Regularization (L2)

- Add shrinkage: objective = NLL + $\lambda \|w\|^2$.
- Geometric view: L2 keeps w inside a hypersphere; all directions penalized equally.
- Mitigates overfitting and stabilizes correlated features.
- scikit-learn: ‘C‘ is inverse strength; $\lambda = \frac{1}{2C}$; smaller ‘C‘ $\Rightarrow$ stronger shrinkage.



Class Imbalance

- 'class_weight="balanced"': weight for class c is $\frac{N}{2N_c}$ (inverse frequency).
- Loss term becomes weight$_{y_i} \times$ log-loss, so minority-class errors count more.
- Helps align the effective threshold with recall/precision needs when positives are rare.

Prediction, Threshold, Ranking

- ‘predict_proba‘ returns $p = \sigma(b + w^\top x)$; default decision uses 0.5.
- Lower threshold $\Rightarrow$ fewer FN but more FP; higher threshold does the opposite.
- ROC AUC scores the ranking by p (or z) independent of any specific threshold.
- Mini diagram: features $\rightarrow z = b + w \cdot x \rightarrow \sigma(z) \rightarrow p$ $\rightarrow$ threshold $\rightarrow$ class.