

# Performance Metrics for Machine Learning

Aayush Bajaj

July 10, 2026

## Contents

|          |                                                    |          |
|----------|----------------------------------------------------|----------|
| <b>1</b> | <b>the confusion matrix</b>                        | <b>1</b> |
| <b>2</b> | <b>roc curves and auc</b>                          | <b>2</b> |
| 2.1      | the probabilistic meaning of auc . . . . .         | 3        |
| <b>3</b> | <b>precision–recall curves and class imbalance</b> | <b>3</b> |
| <b>4</b> | <b>calibration and the brier score</b>             | <b>4</b> |
| <b>5</b> | <b>regression metrics</b>                          | <b>4</b> |
| <b>6</b> | <b>micro versus macro averaging</b>                | <b>5</b> |
| <b>7</b> | <b>everything from scratch</b>                     | <b>5</b> |
| <b>8</b> | <b>choosing, quickly</b>                           | <b>7</b> |
| <b>9</b> | <b>see also</b>                                    | <b>7</b> |

a model is only as good as the number you judge it by, and most of the classic modelling disasters are really metric disasters — a fraud detector with 99.9% accuracy that never flags anything, a medical test tuned to a roc curve nobody deployed at the published threshold. this page is the field guide: what each metric measures, what it silently assumes, and which one to reach for when the classes are lopsided, the probabilities matter, or the target is continuous.

*rule of thumb: before admiring any metric, compute what the trivial constant predictor scores on it*

## 1 the confusion matrix

every threshold classifier on a binary problem induces four counts. fix the convention: **positive** is the rare/interesting class.

|          | predicted +         | predicted −         |
|----------|---------------------|---------------------|
| actual + | true positive (tp)  | false negative (fn) |
| actual − | false positive (fp) | true negative (tn)  |

the zoo of derived metrics is just different marginalisations of this table:

| metric                    | formula                               | reads as                                 |
|---------------------------|---------------------------------------|------------------------------------------|
| accuracy                  | $(tp + tn)/n$                         | fraction correct                         |
| precision (ppv)           | $tp/(tp + fp)$                        | of the flagged, how many are real?       |
| recall (sensitivity, tpr) | $tp/(tp + fn)$                        | of the real, how many were flagged?      |
| specificity (tnr)         | $tn/(tn + fp)$                        | of the negatives, how many were spared?  |
| false positive rate       | $fp/(fp + tn) = 1 - tnr$              | fraction of negatives falsely flagged    |
| f1                        | harmonic mean of precision and recall | one number for the precision/recall pair |

three of these deserve equations. the **f-beta** family interpolates between precision and recall:

$$F_\beta = \frac{(1 + \beta^2) \text{precision} \cdot \text{recall}}{\beta^2 \text{precision} + \text{recall}}, \quad (1)$$

where  $\beta$  is how many times more you care about recall than precision —  $F_1$  is the balanced case,  $F_2$  favours recall (screening tests),  $F_{0.5}$  favours precision (spam filters). being a harmonic mean,  $F_\beta$  punishes imbalance between the two: you cannot buy a good  $F_1$  with precision alone.

*f1 has blind spots: it ignores tn entirely, so it is not invariant to swapping which class you call 'positive'*

**matthews correlation coefficient** uses all four cells — it is literally the pearson correlation between the predicted and actual binary labels:

$$\text{MCC} = \frac{tp \cdot tn - fp \cdot fn}{\sqrt{(tp + fp)(tp + fn)(tn + fp)(tn + fn)}} \in [-1, 1], \quad (2)$$

with 0 for chance-level prediction and  $-1$  for perfect disagreement. of the single-number summaries it is the hardest to fool with imbalance, because every marginal appears in the denominator.

**accuracy** is the one to distrust first: with 10% positives, “always predict negative” scores 0.90. any accuracy must be compared against the majority-class baseline  $\max(\pi, 1 - \pi)$ , where  $\pi$  is the prevalence.

## 2 roc curves and auc

a scoring classifier  $\hat{p}(x)$  plus a sweep of thresholds  $t$  traces the **receiver operating characteristic**: the curve of  $(fpr(t), tpr(t))$  as  $t$  slides from  $+\infty$  (bottom-left) to  $-\infty$  (top-right).

*the name is wwii radar jargon — the ‘operator’ trading missed bombers against false alarms*

- the diagonal  $tpr = fpr$  is chance: a classifier that scores by coin toss.
- the point  $(0, 1)$  is perfection; curves closer to the top-left dominate.
- the roc is invariant under any strictly increasing transform of the scores — it sees only the *ranking*. calibration is invisible here (see below).

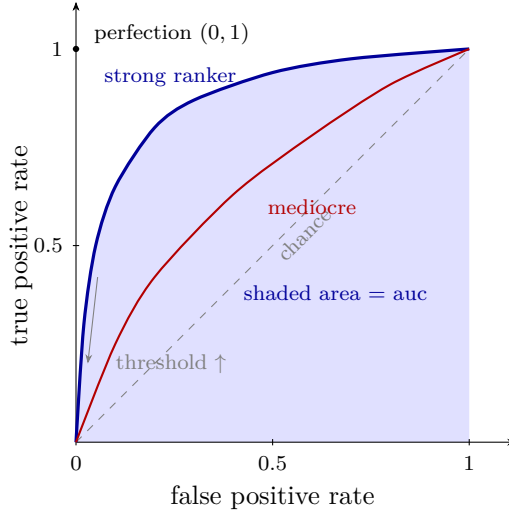


Figure 1: roc space: a strong ranker ( $\text{auc} \approx 0.92$ , shaded), a mediocre one ( $\text{auc} \approx 0.71$ ), and the chance diagonal. the threshold slides from  $+\infty$  at  $(0,0)$  to  $-\infty$  at  $(1,1)$ .

## 2.1 the probabilistic meaning of auc

the area under the roc curve has an exact interpretation that makes it worth reporting:

$$\text{AUC} = \mathbb{P}(\hat{p}(X^+) > \hat{p}(X^-)) + \frac{1}{2} \mathbb{P}(\hat{p}(X^+) = \hat{p}(X^-)), \quad (3)$$

for an independently drawn positive  $X^+$  and negative  $X^-$ : the probability that a random positive outranks a random negative. equivalently, AUC is the normalised mann–whitney  $U$  statistic of the two score samples — which is exactly how you compute it in  $O(n \log n)$  via ranks (code below).<sup>1</sup> chance gives 0.5; a perfect ranker gives 1; anything below 0.5 means you wired the sign backwards.

auc summarises ranking quality across *all* thresholds — which is also its weakness: deployment happens at *one* threshold, and auc happily averages over operating regions you will never use.

## 3 precision–recall curves and class imbalance

roc’s axes are both *class-conditional* rates: tpr conditions on positives, fpr on negatives. neither sees the prevalence  $\pi$ . precision does:

$$\text{precision} = \frac{\pi \cdot \text{tpr}}{\pi \cdot \text{tpr} + (1 - \pi) \text{fpr}}, \quad (4)$$

by bayes’ rule. with  $\pi = 0.001$  and a “great” roc point  $(\text{fpr}, \text{tpr}) = (0.01, 0.9)$ , precision is  $\approx 8\%$ : eleven false alarms per true hit. the roc curve looks superb; the deployed system is a nuisance.

hence the **pr curve** — precision against recall as the threshold sweeps — for rare-positive problems (fraud, retrieval, rare disease):

<sup>1</sup>the equivalence: auc integrates tpr over fpr; substituting the empirical distributions turns the integral into the fraction of positive–negative pairs ranked correctly, which is  $U/(n_+n_-)$  for the mann–whitney  $U$ . ties contribute half, matching the trapezoidal rule on the empirical roc.

- the chance baseline in pr space is the horizontal line at  $\pi$  (a random scorer’s precision at any recall), so the plot shows *lift over prevalence* directly, where roc’s diagonal hides it.
- the pr curve is *not* monotone and interpolation between points is nonlinear — summarise with average precision (area under it) rather than trapezoids.
- a curve dominates in roc space if and only if it dominates in pr space (davis & goadrich, 2006), but *differences are magnified* near low recall in pr space when  $\pi$  is small — exactly the operating region rare-event systems live in.

rule: report roc when classes are roughly balanced or when class-conditional trade-offs are the object; report pr when positives are rare and flagging cost is the object.

## 4 calibration and the brier score

a classifier can rank perfectly and still lie about probabilities.  $\hat{p}$  is **calibrated** when

$$\mathbb{P}(Y = 1 \mid \hat{p}(X) = q) = q \quad \text{for all } q \in [0, 1] : \quad (5)$$

of all the days the forecaster says “70% rain”, it rains on 70%. check it with a **reliability diagram** — bin predictions by  $\hat{p}$ , plot observed frequency per bin against bin centre; calibrated models hug the diagonal.

the **brier score** measures probability quality as plain squared error,

$$\text{BS} = \frac{1}{n} \sum_{i=1}^n (\hat{p}_i - y_i)^2, \quad (6)$$

and admits the murphy decomposition  $\text{BS} = \text{reliability} - \text{resolution} + \text{uncertainty}$ : a calibration penalty, minus a reward for confident discrimination, plus the irreducible  $\bar{y}(1 - \bar{y})$  of the base rate. brier (like log loss) is a **proper scoring rule**: its expectation is uniquely minimised by reporting the true conditional probability — you cannot game it by hedging or exaggerating (gneiting and raftery 2007, *jsa* 102(477), is the definitive treatment). auc and accuracy are not proper; a model selected on them alone may need post-hoc recalibration (platt scaling, isotonic regression) before its probabilities mean anything.

## 5 regression metrics

| metric | formula                                                     | minimised by       | character                                  |
|--------|-------------------------------------------------------------|--------------------|--------------------------------------------|
| rmse   | $\sqrt{\frac{1}{n} \sum_i (y_i - \hat{y}_i)^2}$             | conditional mean   | quadratic loss-geometry: outliers dominate |
| mae    | $\frac{1}{n} \sum_i  y_i - \hat{y}_i $                      | conditional median | linear tails: robust, but kinked at zero   |
| $R^2$  | $1 - \sum_i (y_i - \hat{y}_i)^2 / \sum_i (y_i - \bar{y})^2$ | conditional mean   | rmse rescaled by target variance           |

the “minimised by” column is the loss-geometry fact that matters (derived properly in **loss functions**): choosing rmse versus mae is choosing *which functional of the conditional distribution you are estimating* — the mean or the median. under skewed noise they genuinely differ, and the metric silently picks for you.

- always  $\text{rmse} \geq \text{mae}$  (jensen), with equality only when all errors have equal magnitude; a large gap flags heavy-tailed residuals.

- $R^2$  compares against the constant-mean baseline:  $R^2 = 0$  means “no better than predicting  $\bar{y}$ ”, and it is *negative* whenever the model is worse than that. on held-out data this happens more often than people admit. it is also not comparable across datasets with different target variances (Hastie, Trevor and Tibshirani, Robert and Friedman, Jerome, 2009).
- mae’s median-seeking makes it the honest choice when over- and under-prediction cost the same per unit; rmse when large errors are disproportionately costly.

## 6 micro versus macro averaging

with  $K$  classes, per-class precision/recall/f1 must be aggregated, and the two standard recipes answer different questions:

- **micro**: pool the  $K$  confusion matrices into one (sum tp, fp, fn across classes), then compute the metric. every *instance* votes equally, so frequent classes dominate. for single-label multiclass prediction, micro-precision = micro-recall = micro-f1 = accuracy — all four collapse.<sup>2</sup>
- **macro**: compute the metric per class, then take the unweighted mean over classes. every *class* votes equally: a 50-sample rare class counts as much as a 50,000-sample common one. macro-f1 is the standard summary when rare-class competence is the point.
- **weighted macro**: per-class metrics averaged with class-frequency weights — a compromise that mostly behaves like micro.

the gap between micro and macro is itself a diagnostic: micro  $\gg$  macro means the model is coasting on frequent classes.

## 7 everything from scratch

confusion-matrix zoo, rank-based auc, brier, and the regression metrics — on a synthetic problem with 10% positives where scores are gaussian ( $\mathcal{N}(1,1)$  for positives,  $\mathcal{N}(0,1)$  for negatives), so the true auc is  $\Phi(1/\sqrt{2}) \approx 0.760$ . the posterior  $\hat{p}_{\text{cal}}$  is computed by bayes’ rule (calibrated by construction);  $\hat{p}_{\text{sig}} = \sigma(s)$  ignores the prevalence (miscalibrated on purpose).

```

1 import numpy as np
2 from math import erf, sqrt
3
4 rng = np.random.default_rng(1)
5
6 # synthetic scores: 10% positives; score ~ N(1,1) for +, N(0,1) for -
7 n, prev = 20000, 0.10
8 y = (rng.uniform(size=n) < prev).astype(int)
9 s = np.where(y == 1, rng.normal(1.0, 1.0, n), rng.normal(0.0, 1.0, n))
10
11 phi = lambda z: np.exp(-z**2 / 2) / np.sqrt(2 * np.pi)

```

<sup>2</sup>every false positive for one class is a false negative for another, so pooled fp = pooled fn, forcing pooled precision = pooled recall = accuracy.

## references

Hastie, Trevor and Tibshirani, Robert and Friedman, Jerome (2009). *The Elements of Statistical Learning*, Springer.

```

12 p_cal = prev * phi(s - 1) / (prev * phi(s - 1) + (1 - prev) * phi(s)) # bayes
   ↪ posterior
13 p_sig = 1 / (1 + np.exp(-s)) # miscalibrated: ignores prevalence
14
15 def confusion(y, yhat):
16     tp = int(np.sum((yhat == 1) & (y == 1))); fp = int(np.sum((yhat == 1) & (y
   ↪ == 0)))
17     fn = int(np.sum((yhat == 0) & (y == 1))); tn = int(np.sum((yhat == 0) & (y
   ↪ == 0)))
18     return tp, fp, fn, tn
19
20 print("thresh  acc  prec  rec  spec  f1  mcc")
21 for t in (0.10, 0.25, 0.50):
22     tp, fp, fn, tn = confusion(y, (p_cal >= t).astype(int))
23     acc = (tp + tn) / n
24     prec = tp / (tp + fp)
25     rec = tp / (tp + fn)
26     spec = tn / (tn + fp)
27     f1 = 2 * prec * rec / (prec + rec)
28     mcc = (tp * tn - fp * fn) / np.sqrt(
29         float(tp + fp) * (tp + fn) * (tn + fp) * (tn + fn))
30     print(f"{t:>6.2f} {acc:>6.3f} {prec:>6.3f} {rec:>6.3f} "
31           f"{spec:>6.3f} {f1:>6.3f} {mcc:>6.3f}")
32
33 def auc_rank(y, p): # mann-whitney rank statistic
34     order = np.argsort(p)
35     ranks = np.empty(n); ranks[order] = np.arange(1, n + 1)
36     npos, nneg = y.sum(), n - y.sum()
37     return (ranks[y == 1].sum() - npos * (npos + 1) / 2) / (npos * nneg)
38
39 auc_theory = 0.5 * (1 + erf(1 / sqrt(2) / sqrt(2))) # Phi(1/sqrt(2))
40 print(f"\nauc: posterior = {auc_rank(y, p_cal):.4f} sigmoid = {auc_rank(y,
   ↪ p_sig):.4f}"
41       f" theory = {auc_theory:.4f}")
42
43 print(f"brier: calibrated = {np.mean((p_cal - y)**2):.4f}"
44       f" miscalibrated = {np.mean((p_sig - y)**2):.4f}"
45       f" predict-prevalence = {np.mean((prev - y)**2):.4f}")
46
47 # regression metrics from scratch, prediction = true conditional mean
48 x = rng.uniform(0, 1, 500)
49 ytrue = 2 * x + rng.normal(0, 0.3, 500)
50 yhat = 2 * x
51 rmse = np.sqrt(np.mean((ytrue - yhat) ** 2))
52 mae = np.mean(np.abs(ytrue - yhat))
53 r2 = 1 - np.sum((ytrue - yhat) ** 2) / np.sum((ytrue - ytrue.mean()) ** 2)
54 print(f"regression: rmse = {rmse:.4f} mae = {mae:.4f} r2 = {r2:.4f}")

```

| thresh | acc   | prec  | rec   | spec  | f1    | mcc   |
|--------|-------|-------|-------|-------|-------|-------|
| 0.10   | 0.696 | 0.208 | 0.707 | 0.695 | 0.322 | 0.255 |
| 0.25   | 0.877 | 0.360 | 0.265 | 0.946 | 0.305 | 0.243 |
| 0.50   | 0.899 | 0.588 | 0.039 | 0.997 | 0.073 | 0.133 |

auc: posterior = 0.7646 sigmoid = 0.7646 theory = 0.7602  
 brier: calibrated = 0.0820 miscalibrated = 0.2752 predict-prevalence = 0.0916  
 regression: rmse = 0.2934 mae = 0.2367 r2 = 0.7892

four object lessons in one printout:

- **the accuracy trap:** at threshold 0.5 accuracy is 0.899 — while recall is 0.039. the majority-class baseline is 0.90, so this “90%-accurate” model is *worse than doing nothing* by the only metric it advertises. mcc (0.13) is not fooled.

- **thresholds move everything:** precision and recall trade off threefold across three thresholds; a metric quoted without its threshold is an unfinished sentence.
- **auc is transform-blind:** the calibrated posterior and the naive sigmoid score identical auc (0.7646, matching  $\Phi(1/\sqrt{2})$ ) because one is a monotone transform of the other.
- **brier is not:** the calibrated posterior scores 0.082, beating even the “predict the prevalence” straw man (0.092), while the same-auc sigmoid scores 0.275. ranking quality and probability quality are different axes; brier sees the second, auc only the first.

## 8 choosing, quickly

- balanced classes, hard decisions: accuracy is fine; mcc if you want insurance.
- imbalanced, flagging costs matter: pr curve + average precision, f-beta at the deployment threshold.
- comparing rankers across thresholds: roc-auc.
- probabilities feed a downstream decision: brier or log loss, plus a reliability diagram; recalibrate if needed.
- regression: rmse when big misses are catastrophic, mae when they are merely proportional; always sanity-check  $R^2$  against zero.
- multiclass with rare classes you care about: macro-f1, and report the micro-macro gap.

and estimate all of them on data the model never trained on — the machinery for that lives in [cross validation](#).

## 9 see also

- [cross validation](#) — how to estimate these metrics without flattering yourself
- [loss functions](#) — metrics you can differentiate: the training-time counterparts
- [the bias-variance decomposition](#) — what sits underneath a test-set rmse
- [the no free lunch theorem](#) — why no single metric (or model) rules them all