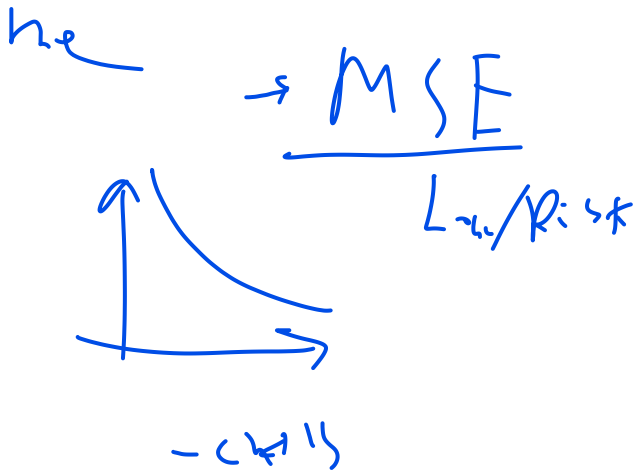
Outline



Why even bother choosing a metric?

A spam filter sees 100 emails: 99 legit, 1 spam.

A lazy model labels *everything* "legit":

- **Accuracy: 99%.** Looks excellent.
- Spam caught: **zero**. The model is useless.

The model didn't change - only the *number we chose to look at*. Accuracy quietly hid the one mistake that mattered.

A metric is a **definition of success**. Pick the wrong one and a worthless model looks great.

Same trap in regression

Two delivery-time models, evaluated on the same orders. *Identical average error.*

- **Model A:** every estimate is off by about 10 minutes.
- **Model B:** usually spot-on, but once in a while off by 3 hours.

Same mean error, completely different customer experience. If the *worst case* is what burns you, you care about the **maximum absolute deviation**, not the average.

A pricing model is the mirror image: being off by 50k AMD is a disaster on a 200k flat (25%) and a rounding error on a 5M flat (1%). There, **percentage error** is the honest metric.

Takeaway: there is no universal "accuracy." The metric encodes *which mistake you cannot afford*.

Loss: what the model optimizes

Loss is the function minimized *while the model is being fit*.

- It drives the fitting algorithm, so for gradient-based methods it must be **differentiable**.
- Examples: **MSE** for ordinary least squares, **log loss** for logistic regression.

You rarely look at the loss value itself. It exists to *steer the optimizer*, not to be reported.

Metric: what you judge the model by

Metric is the number you use to *evaluate* a trained model. It need not be differentiable.

A metric does **two jobs**:

1. **Reporting** - the number you hand to stakeholders ("typical error is 20 kAMD").
2. **Selecting** - the score your cross-validation and hyperparameter search optimize. *The metric you tune against is the model you end up with.*

Loss and metric *can* coincide (often do in regression), but needn't: you might **train** on MSE for its smooth math, then **tune and report** MAE because stakeholders think in "typical error."

The question every metric answers: what kind of mistake hurts you most? One big error or many small ones? Errors on cheap items or expensive ones? Over- or under-prediction?

Magnitude metrics

How big are the errors, in the units you actually care about? Every metric here answers that - they only disagree about how much a single big miss should count.

MSE, RMSE, MAE: how the penalty shapes differ

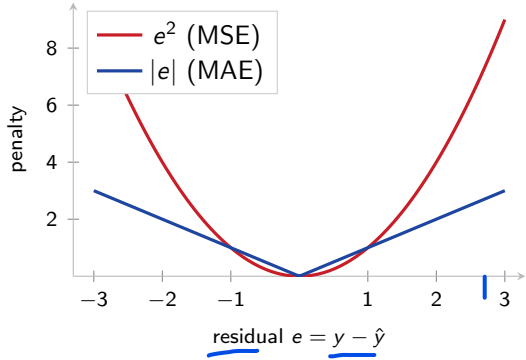
MSE (Mean Squared Error), **RMSE** (Root Mean Squared Error):

$$\text{MSE} = \frac{1}{n}\sum (y_i - \hat{y}_i)^2, \quad \text{RMSE} = \sqrt{\text{MSE}}$$

MAE (Mean Absolute Error):

$$\text{MAE} = \frac{1}{n}\sum |y_i - \hat{y}_i|$$

- **Squared** penalty: one error of 10 costs 100; ten errors of 1 cost 10 total. *Big errors dominate.*
- **Absolute** penalty: those two cases cost the same. *Robust* to outliers.
- Report **RMSE** (units of y); minimize **MSE** internally (smoother).



The squared penalty curves upward fast; the absolute penalty grows at a constant rate.

MAE and MedAE: robust variants

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \qquad \text{MedAE} = \text{median}_i\ |y_i - \hat{y}_i|$$

- **MAE** (Mean Absolute Error). Units of y. Linear penalty - an outlier of 100 adds 100, not 10 000.
- **MedAE** (Median Absolute Error). The middle residual. Even more robust: half your predictions are at least this good.
- Where each shines: *MAE* for outlier-heavy targets (real estate, finance). *MedAE* for "typical case" reporting: "half my predictions land within 30 kAMD."

Predict first: which metric notices the outlier?

10 rent predictions. Nine are off by 1 kAMD; one is off by 100 kAMD (a luxury flat the model missed).

Before looking: rank MSE, RMSE, MAE, MedAE from most to least disturbed by that one outlier.

Predict first: which metric notices the outlier?

10 rent predictions. Nine are off by 1 kAMD; one is off by 100 kAMD (a luxury flat the model missed).

Before looking: rank MSE, RMSE, MAE, MedAE from most to least disturbed by that one outlier.

Metric	Value	Interpretation
MSE	$\frac{9 \cdot 1^2 + 1 \cdot 100^2}{10} = 1000.9$	dominated by the outlier
RMSE	$\sqrt{1000.9} \approx 31.6$	"typical error 32 kAMD" (misleading)
MAE	$\frac{9 \cdot 1 + 100}{10} = 10.9$	visible but moderate
MedAE	median $= 1$	outlier completely invisible

Same model, four different stories. The metric you pick *is* the conclusion you report.

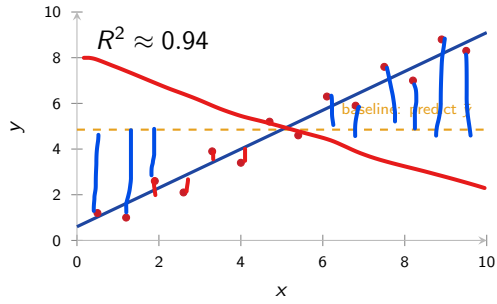
Relative metrics

A raw error of "20" means nothing on its own - 20 kAMD or 20 EUR? These metrics normalize, so the number travels across problems and scales.

R^2: fraction of variance explained

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

- Compares your model (blue) to the dumbest baseline (orange dashed): *always predict the mean* $\bar{y}$.
- $R^2 = 1$: perfect. $R^2 = 0$: no better than $\bar{y}$.
- **Scale-free** - no units, so it reads the same whether y is in AMD or EUR (unlike MAE/RMSE).

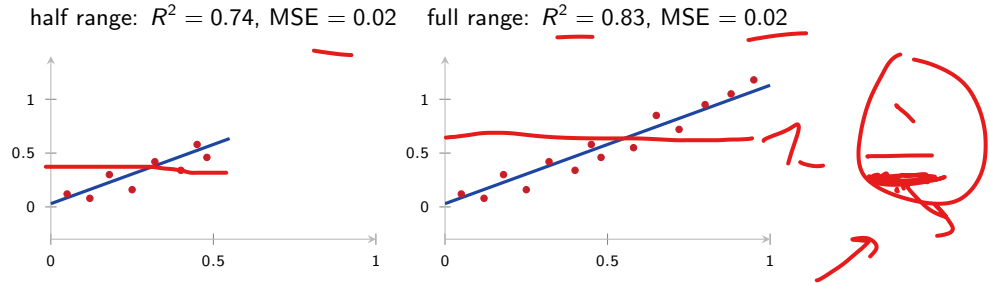


Tight scatter around the line ⇒ high R^2.

R^2 **can go negative on test data.** It just means your model does worse than predicting the training mean. Not a bug - a warning. Investigate.

Better $R^2 \neq$ better fit

Same model, same MSE, two views of the data. R^2 depends on the *range* of y; MSE depends on the *magnitude* of residuals.



Why? $R^2 = 1 - \text{SSE/SST}$. Extending the range inflates SST much faster than SSE $\Rightarrow$ the ratio shrinks $\Rightarrow$ R^2 rises, even though the fit didn't improve. **Don't compare R^2 across datasets with different y-ranges.** Pair it with RMSE/MAE.

Adjusted and generalized R^2

Adjusted R^2 - penalizes useless features:

$$R^2_{\text{adj}} = 1 - (1 - R^2)\,\frac{n-1}{n-p-1}$$

Plain R^2 *never* drops when you add a feature, even pure noise. Adjusted R^2 *can* drop, so it won't reward bloating the model with junk.

Generalized R^2 - same skeleton, any loss, any baseline:

$$R^2_{\text{gen}} \;=\; 1 - \frac{\text{Loss}_{\text{model}}}{\text{Loss}_{\text{baseline}}}$$

Classical R^2 is the special case (loss = SSE, baseline = constant mean). Swap in MAE, or a smarter baseline, and the same "fraction of the gap closed" reading still holds.

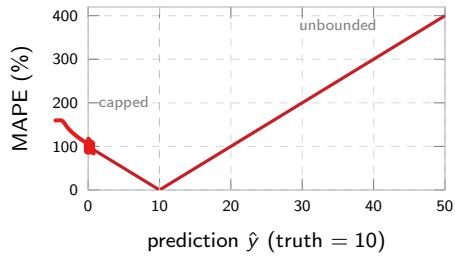
MAPE: the popular but flawed default

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{|y_i|}$$

MAPE = Mean Absolute Percentage Error. Popular because it reports as a percentage; stakeholders love it.

Three known problems:

1. **Undefined when $y_i = 0$.** Any zero target $\Rightarrow$ division by zero $\Rightarrow$ infinite MAPE.
2. **Asymmetric.** Under-prediction capped at 100%; over-prediction unbounded (plot, fixed truth $y = 10$).
3. **Biases models toward under-prediction** when used as a loss.



Asymmetric around $y = 10$: under- vs over-prediction are penalized unequally.

SMAPE: the symmetric fix

$$\text{SMAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \frac{2\,|y_i - \hat{y}_i|}{|y_i| + |\hat{y}_i|}$$

SMAPE = Symmetric Mean Absolute Percentage Error.

- Symmetric: a 50% over-prediction and a 50% under-prediction get the same score.
- Bounded in $[0\%, 200\%]$ - no infinities.
- Defined when $y_i = 0$ (as long as $\hat{y}_i \neq 0$).
- Still quirky (a missed zero pins it near 200%), but better-behaved than MAPE.

MASE (Mean Absolute Scaled Error): MAE of the model divided by the MAE of a naive "predict last value" forecast. Scale-free, handles zeros, the standard for time-series competitions (M4, M5).

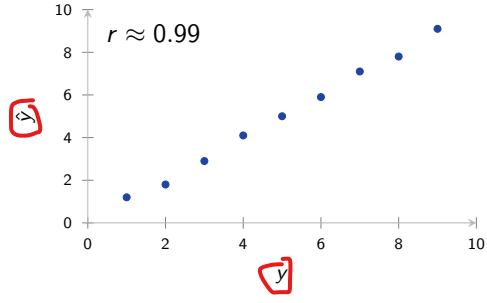
Rank and association

Sometimes you don't need the exact value, only the right order - which listing is pricier, not by how much. These measures score the ordering.

Pearson correlation: do predictions move with the truth?

$$r = \frac{\mathrm{cov}(y, \hat{y})}{\sigma_y \, \sigma_{\hat{y}}} \in [-1, 1]$$

- Measures **linear co-movement**: when y goes up, does $\hat{y}$ go up by a proportional amount?
- **Bounded** in $[-1, 1]$ and scale-free - which is exactly what makes it usable: unlike MAE/RMSE, 0.9 reads as "strong" on its own, no units to look up. A clean *screening* tool for which raw feature tracks the target.
- $r = 1$: perfect straight-line agreement. 0: none. -1: perfectly inverted.



Points hug a straight line ⇒ r near 1.

Catch: r only sees *straight-line* structure. If $\hat{y}$ tracks y through a bent-but-always-increasing curve, r drops below 1 even though the ranking is perfect.

Spearman: correlation on ranks

Spearman's ρ is just Pearson computed on the *ranks* of y and $\hat{y}$ instead of their raw values.

- Sees any **monotonic** relationship, not only linear ones - a curved-but-increasing fit scores $\rho = 1$.
- Robust to outliers and to monotone transforms (log, sqrt) of either variable.
- Use it when you care about **getting the order right** (ranking listings, ordinal targets), not the exact values.

Kendall's τ is a third option (fraction of correctly-ordered pairs minus incorrectly-ordered ones); same spirit, more robust to ties, slower to compute.

Predict first: when correlation lies

Your model's predictions on a 3-row test set:

y_{true}	100	200	300
$\hat{y}$	1100	1200	1300

Predict: Pearson r between y and $\hat{y}$? MAE? Would you deploy this?

Predict first: when correlation lies

Your model's predictions on a 3-row test set:

y_{true}	100	200	300
$\hat{y}$	1100	1200	1300

Predict: Pearson r between y and $\hat{y}$? MAE? Would you deploy this?

- **Pearson** $r = 1.0$**.** Predictions are perfectly linearly related to truth - just shifted by $+1000$.
- **MAE** $= 1000$**.** Predictions are uniformly terrible.
- No.

Correlation measures association, not accuracy. Use correlations for screening features. Use MAE/RMSE for evaluating models.

AIC and BIC: model comparison, not evaluation

$$\text{AIC} = 2k - 2\ln(\hat{L}) \qquad \text{BIC} = k\ln(n) - 2\ln(\hat{L})$$

AIC = Akaike Information Criterion, **BIC** = Bayesian Information Criterion; k = number of parameters, n = sample size, $\hat{L}$ = maximized likelihood.

- Both compare *candidate models on the same data*. Lower is better.
- They balance **fit quality** ($-2\ln\hat{L}$) against **model complexity** (k).
- Require a probabilistic model so $\hat{L}$ is defined - available from `statsmodels` OLS, not directly from sklearn estimators.
- These are *model selection* criteria, not accuracy metrics. They tell you which model to pick; they do not tell you how accurate it is on unseen data.

AIC vs. BIC: when each wins

Both penalize complexity, but **BIC penalizes harder** once $n \geq 8$ (because $\ln n > 2$).

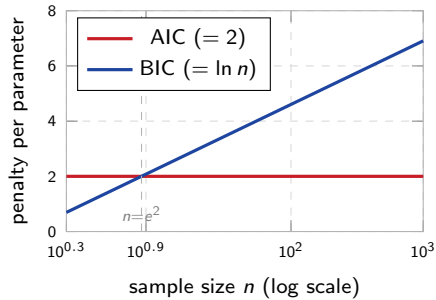
AIC ($2k$ per param):

- Better for *prediction* - keeps useful features.
- Asymptotically equivalent to leave-one-out CV.

BIC ($\ln(n) \cdot k$ per param):

- Better for *model identification* - conservative.
- Consistent: as $n \to \infty$, BIC picks the true model.

Practical: report both. When they disagree, BIC is more conservative. Pair with CV for honest accuracy.



They cross at $n = e^2 \approx 7.4$. Beyond that, BIC's complexity tax is harsher.

Residual diagnostics

A metric collapses every mistake into one number. A plot shows you the shape of those mistakes - and points at what to fix.

The five diagnostic plots

We'll use five standard plots, in the order you'd actually look at them:

- **Predicted vs. actual.** The intuitive overview - where and how predictions drift.
- **Residuals vs. fitted.** Are residuals patterned? (missed nonlinearity, heteroscedasticity)
- **Scale-Location.** Does the residual *spread* depend on the prediction?
- **Normal QQ.** Are residuals roughly Gaussian? (matters for OLS inference)
- **Leverage / Cook's distance.** Which observations have outsized influence?

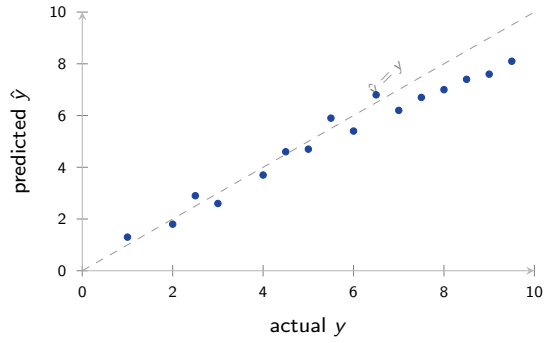
Look at all of them. A model with $R^2 = 0.97$ can still be broken in ways only these plots reveal.

Predicted vs. actual: the first plot to draw

Plot $\hat{y}_i$ against y_i with the reference line $\hat{y} = y$. Perfect predictions sit exactly on it.

- Points **above** the line: over-prediction. **Below**: under-prediction.
- A **bend** away from the line at one end = systematic bias there (often the model compresses the extremes).
- A **widening funnel** = error grows with the target (heteroscedasticity).

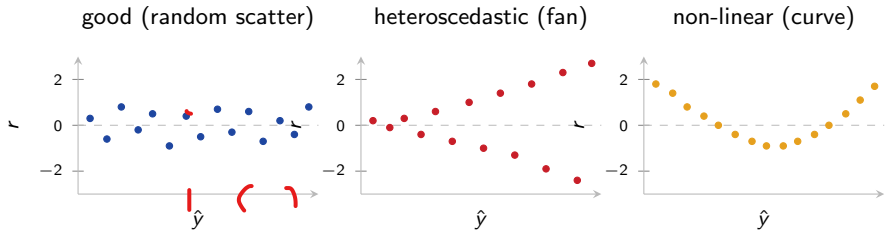
The single most intuitive diagnostic, and the easiest to show a non-technical stakeholder.



High-y points fall below the line: the model under-predicts the top end.

Residuals vs. fitted

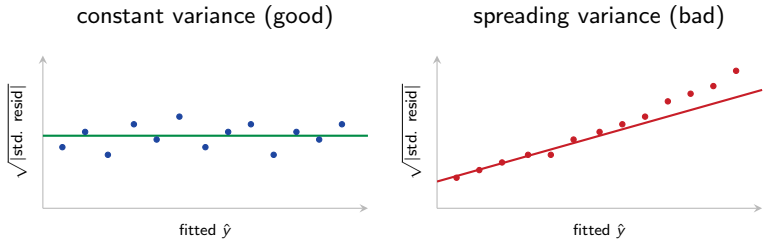
Residual $r_i = y_i - \hat{y}_i$ plotted against the prediction $\hat{y}_i$. Should look like random scatter around 0.



Fan shape: variance grows with prediction (heteroscedasticity). Fix: log-transform y, weighted least squares, or robust standard errors.
Curve: you're missing a non-linear term. Fix: add polynomial features, interaction terms, or switch to a tree.

Scale-Location: a sharper read on spreading variance

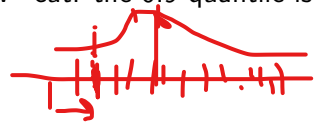
Plots $\sqrt{|\text{standardized residual}|}$ against the fitted value. It strips out the *sign* of the error and isolates its *size*, so a trend in spread is easier to see than in residuals-vs-fitted.



- **Flat horizontal band**: constant variance (homoscedastic) - what you want.
- **Rising line**: the model's errors grow with its predictions. Fix: log-transform y, weighted least squares, or robust errors.

Normal QQ plot: is the error Gaussian?

A **quantile** is a "what fraction falls below" cut: the 0.9 quantile is the value 90% of the data sits under.

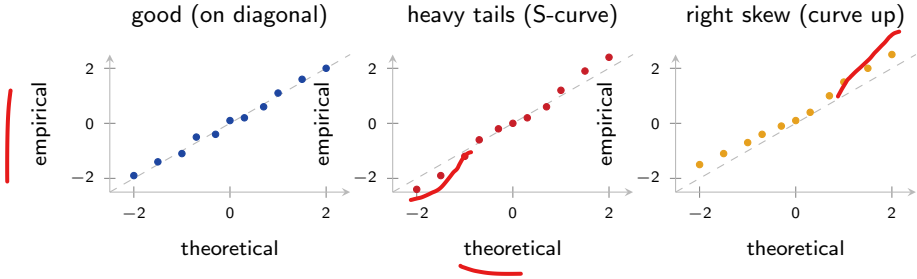


How the plot is built:

1. Sort the residuals from smallest to largest.
2. For each, ask: if these were truly Normal, where *should* this ranked value land? That's its theoretical quantile.
3. Plot actual (sorted residual) on one axis vs. theoretical (Normal quantile) on the other.

If the residuals really are Normal, every point lands on the **45° diagonal**: the value they have equals the value Normal predicts. Departures from the line are departures from Normality - and the *shape* of the departure tells you which way.

Reading the QQ plot



- **On the line**: residuals are Gaussian.
- **S-curve** (ends peel below-left and above-right): *heavy tails* - more extreme errors than Normal predicts.
- **Upward bow**: *right skew* - a long tail of large positive residuals.

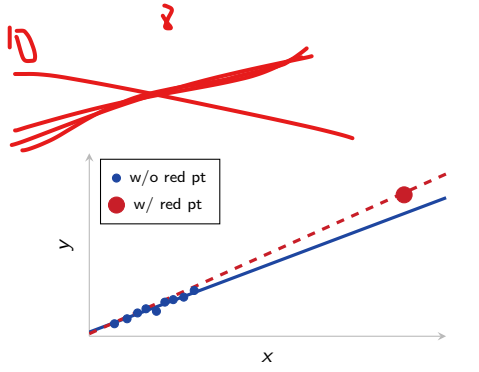
Why care? OLS *point predictions* don't need Normal residuals. OLS *inference* - CIs, t- and F-tests, p-values - does. Heavy tails or skew $\Rightarrow$ your p-values are wrong even when the predictions are fine.

Leverage and Cook's distance

Leverage h_i. How unusual is $\mathbf{x}^{(i)}$? Diagonal of the hat matrix $\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$. High leverage $\Rightarrow$ this point can pull the fit strongly toward itself.

Cook's distance D_i. How much would $\hat{\boldsymbol{\theta}}$ change if you removed observation i? Combines leverage AND residual size.

$$D_i \;=\; \frac{r_i^2}{p \cdot \mathrm{MSE}} \cdot \frac{h_i}{(1-h_i)^2}$$



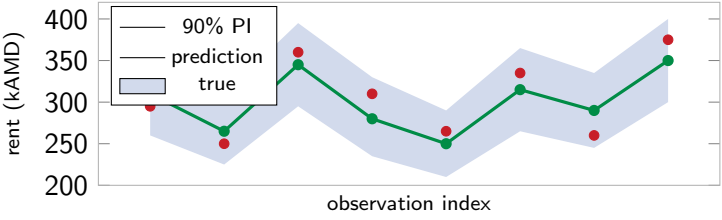
The high-leverage red point pulls the line toward itself.

Rule of thumb: $D_i > 4/n$ is suspicious. Investigate before deleting. High-Cook's points are often (a) data-entry errors, (b) genuinely unusual but real cases (a Yerevan 200 m^2 penthouse in training), or (c) telling you the model class is too rigid.

Point predictions aren't enough

"This apartment will rent for 310 kAMD" is a *point prediction*. It says nothing about uncertainty.

"90% prediction interval: [265, 355] kAMD" is **honest** - it communicates what the model knows and doesn't know.



A well-calibrated 90% interval should contain the truth about 90% of the time on held-out data. The modern tool is *conformal prediction* (`MAPIE`), which gives coverage guarantees regardless of model.

Recap

- A metric is a **definition of success** - it decides which model wins your tuning, not just what you report.
- **Magnitude** (MSE, RMSE, MAE, MedAE): how big are the errors? They differ entirely in how they treat outliers.
- **Relative** (R^2, adjusted, MAPE, SMAPE): scale-free. R^2 can go negative on test; MAPE blows up near zero; SMAPE tames it.
- **Correlation** measures association, not accuracy - a perfect Pearson can hide a constant offset of 1000.
- **AIC / BIC**: compare candidate models, don't measure accuracy. Pair with CV.
- **Diagnostics** (predicted-vs-actual, residuals-vs-fitted, scale-location, QQ, leverage/Cook's) show *what kind* of wrong - one number can't.
- **Prediction intervals** turn a point guess into an honest range.

One model, many metrics. Report - and tune on - the ones that match what the decision actually costs.