Data Science for Mathematicians

Lesson 04: Model Evaluation and Statistical Inference

Department of Mathematics and Computer Science

Outline

1. Beyond Model Fitting
2. The Bias-Variance Tradeoff
3. Metrics for Regression Models
4. Metrics for Classification Models
5. Estimating Generalization Error

From Fitting to Evaluating

In previous lessons we derived OLS and Naive Bayes—both focused on **fitting** models to data.

The central question now:

Having fit a model, is it any good?

From Fitting to Evaluating

In previous lessons we derived OLS and Naive Bayes—both focused on **fitting** models to data.

The central question now:

Having fit a model, is it any good?

- Low training error $\neq$ good model
- The dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$ is a **sample**, not the population
- The estimator $\hat{\beta} = (X^T X)^{-1} X^T y$ is a **random variable**—it depends on which sample we drew

Training Error vs. Generalization Error

- **Training error:** Performance on the data used to fit the model

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2$$

Training Error vs. Generalization Error

- **Training error:** Performance on the data used to fit the model

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2$$

- **Generalization error:** Performance on *new*, *unseen data* from the same process

Training Error vs. Generalization Error

- **Training error:** Performance on the data used to fit the model

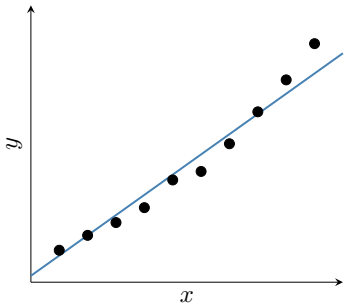
$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2$$

- **Generalization error:** Performance on *new, unseen data* from the same process

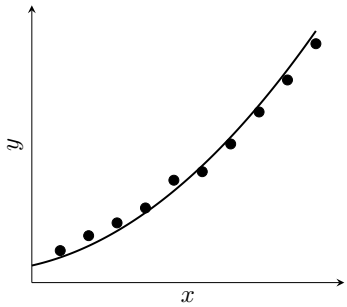
Our true objective: minimize **generalization error**, not training error.

Memorization vs. Generalization

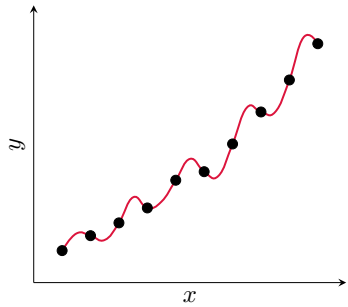
Overfitting: the model fits the noise, not the signal.



Underfitting
(High Bias)



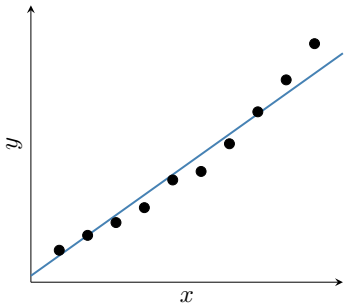
Good Fit



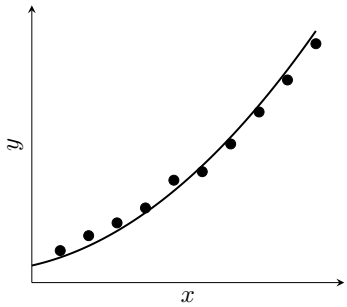
Overfitting
(High Variance)

Memorization vs. Generalization

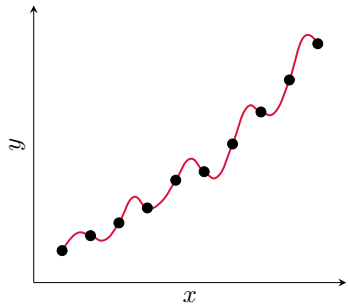
Overfitting: the model fits the noise, not the signal.



Underfitting
(High Bias)



Good Fit



Overfitting
(High Variance)

Goal: capture stable, repeatable patterns while ignoring stochastic noise.

Lecture Roadmap

Three pillars of model evaluation:

1. **Theoretical Framework:** Bias-Variance Decomposition
   - Why does error arise? What are its components?

Lecture Roadmap

Three pillars of model evaluation:

1. **Theoretical Framework:** Bias-Variance Decomposition
   - Why does error arise? What are its components?
2. **Practical Metrics:** MSE, R^2, Precision, Recall, F_1, AUC
   - How do we measure error in practice?

Lecture Roadmap

Three pillars of model evaluation:

1. **Theoretical Framework:** Bias-Variance Decomposition
   - Why does error arise? What are its components?
2. **Practical Metrics:** MSE, R^2, Precision, Recall, F_1, AUC
   - How do we measure error in practice?
3. **Estimation Procedure:** Cross-Validation
   - How do we reliably estimate generalization error?

The Data Generating Process

Definition: Data Generating Process

We assume observed data arises from:

$$y = f(x) + \epsilon$$

where $f : \mathbb{R}^p \to \mathbb{R}$ is the true (unknown) function and ϵ is random noise with:

1. $\mathbb{E}[\epsilon] = 0$ (unbiased noise)
2. $\text{Var}(\epsilon) = \sigma^2$ (constant, finite variance)

The Data Generating Process

Definition: Data Generating Process

We assume observed data arises from:

$$y = f(x) + \epsilon$$

where $f : \mathbb{R}^p \to \mathbb{R}$ is the true (unknown) function and ϵ is random noise with:

1. $\mathbb{E}[\epsilon] = 0$ (unbiased noise)
2. $\text{Var}(\epsilon) = \sigma^2$ (constant, finite variance)

- f is the **signal** we wish to learn
- ϵ is the **noise** (measurement error, unmodeled variables, randomness)
- Our estimator $\hat{f}(x; D)$ is a **random quantity**—it depends on the training set D

Expected Prediction Error

Consider a new, unseen point (x_0, y_0) with $y_0 = f(x_0) + \epsilon_0$.

Definition: Expected Prediction Error

The expected squared prediction error at x_0:

$$\mathbb{E}\left[(y_0 - \hat{f}(x_0))^2\right]$$

where the expectation is over:

- the random training dataset D
- the random noise ϵ_0 in the test point

Expected Prediction Error

Consider a new, unseen point (x_0, y_0) with $y_0 = f(x_0) + \epsilon_0$.

Definition: Expected Prediction Error

The expected squared prediction error at x_0:

$$\mathbb{E}\left[(y_0 - \hat{f}(x_0))^2\right]$$

where the expectation is over:

- the random training dataset D
- the random noise ϵ_0 in the test point

Can we decompose this error into interpretable components?

The Bias-Variance Decomposition

Theorem: Bias-Variance Decomposition

For $y = f(x) + \epsilon$ with $\mathbb{E}[\epsilon] = 0$, $\text{Var}(\epsilon) = \sigma^2$:

$$\mathbb{E}[(y_0 - \hat{f}(x_0))^2] = \underbrace{\left(\mathbb{E}[\hat{f}(x_0)] - f(x_0)\right)^2}_{\text{Squared Bias}} + \underbrace{\text{Var}(\hat{f}(x_0))}_{\text{Variance}} + \underbrace{\sigma^2}_{\text{Irreducible Error}}$$

The Bias-Variance Decomposition

Theorem: Bias-Variance Decomposition

For $y = f(x) + \epsilon$ with $\mathbb{E}[\epsilon] = 0$, $\text{Var}(\epsilon) = \sigma^2$:

$$\mathbb{E}[(y_0 - \hat{f}(x_0))^2] = \underbrace{\left(\mathbb{E}[\hat{f}(x_0)] - f(x_0)\right)^2}_{\text{Squared Bias}} + \underbrace{\text{Var}(\hat{f}(x_0))}_{\text{Variance}} + \underbrace{\sigma^2}_{\text{Irreducible Error}}$$

Three distinct, additive sources of error:

- **Bias**^{2}: systematic error from model assumptions
- **Variance**: sensitivity to the particular training set
- $\boldsymbol{\sigma^2}$: noise inherent to the process (cannot be reduced)

Proof Sketch: Setup

Write $\hat{f} = \hat{f}(x_0; D)$ and $f = f(x_0)$. Add and subtract $\mathbb{E}[\hat{f}]$:

$$\mathbb{E}[(y_0 - \hat{f})^2] = \mathbb{E}\left[\left((y_0 - \mathbb{E}[\hat{f}]) + (\mathbb{E}[\hat{f}] - \hat{f})\right)^2\right]$$

$$= \underbrace{\mathbb{E}[(y_0 - \mathbb{E}[\hat{f}])^2]}_{\text{Term 1}} + \underbrace{\mathbb{E}[(\hat{f} - \mathbb{E}[\hat{f}])^2]}_{\text{Term 2}} + \underbrace{2\mathbb{E}[(y_0 - \mathbb{E}[\hat{f}])(\mathbb{E}[\hat{f}] - \hat{f})]}_{\text{Term 3}}$$

Proof Sketch: Setup

Write $\hat{f} = \hat{f}(x_0; D)$ and $f = f(x_0)$. Add and subtract $\mathbb{E}[\hat{f}]$:

$$\begin{aligned}\mathbb{E}[(y_0 - \hat{f})^2] &= \mathbb{E}\left[\left((y_0 - \mathbb{E}[\hat{f}]) + (\mathbb{E}[\hat{f}] - \hat{f})\right)^2\right] \\ &= \underbrace{\mathbb{E}[(y_0 - \mathbb{E}[\hat{f}])^2]}_{\text{Term 1}} + \underbrace{\mathbb{E}[(\hat{f} - \mathbb{E}[\hat{f}])^2]}_{\text{Term 2}} + \underbrace{2\mathbb{E}[(y_0 - \mathbb{E}[\hat{f}])(\mathbb{E}[\hat{f}] - \hat{f})]}_{\text{Term 3}}\end{aligned}$$

Term 2 $= \mathrm{Var}(\hat{f})$ (by definition)

Term 3 $= 0$ (y_0 and $\hat{f}$ are independent; $\mathbb{E}_D[\mathbb{E}[\hat{f}] - \hat{f}] = 0$)

Proof Sketch: Term 1

For Term 1, substitute $y_0 = f + \epsilon_0$:

$$\begin{aligned}
\mathbb{E}[(y_0 - \mathbb{E}[\hat{f}])^2] &= \mathbb{E}\left[\left((f - \mathbb{E}[\hat{f}]) + \epsilon_0\right)^2\right] \\
&= (f - \mathbb{E}[\hat{f}])^2 + 2(f - \mathbb{E}[\hat{f}])\underbrace{\mathbb{E}[\epsilon_0]}_{=0} + \underbrace{\mathbb{E}[\epsilon_0^2]}_{=\sigma^2} \\
&= \underbrace{(f - \mathbb{E}[\hat{f}])^2}_{\text{Bias}^2} + \sigma^2
\end{aligned}$$

Combining all terms:

$$\mathbb{E}[(y_0 - \hat{f})^2] = \text{Bias}^2(\hat{f}) + \text{Var}(\hat{f}) + \sigma^2 \qquad \square$$

Proof Sketch: Term 1

For Term 1, substitute $y_0 = f + \epsilon_0$:

$$\mathbb{E}[(y_0 - \mathbb{E}[\hat{f}])^2] = \mathbb{E}\left[\left((f - \mathbb{E}[\hat{f}]) + \epsilon_0\right)^2\right]$$

$$= \underbrace{(f - \mathbb{E}[\hat{f}])^2}_{\text{Bias}^2} + \sigma^2$$

Combining all terms:

$$\mathbb{E}[(y_0 - \hat{f})^2] = \text{Bias}^2(\hat{f}) + \text{Var}(\hat{f}) + \sigma^2 \qquad \square$$

Proof Sketch: Term 1

For Term 1, substitute $y_0 = f + \epsilon_0$:

$$\begin{aligned}
\mathbb{E}[(y_0 - \mathbb{E}[\hat{f}])^2] &= \mathbb{E}\left[\left((f - \mathbb{E}[\hat{f}]) + \epsilon_0\right)^2\right] \\
&= (f - \mathbb{E}[\hat{f}])^2 + 2(f - \mathbb{E}[\hat{f}])\underbrace{\mathbb{E}[\epsilon_0]}_{=0} + \underbrace{\mathbb{E}[\epsilon_0^2]}_{=\sigma^2} \\
&= \underbrace{(f - \mathbb{E}[\hat{f}])^2}_{\text{Bias}^2} + \sigma^2
\end{aligned}$$

Proof Sketch: Term 1

For Term 1, substitute $y_0 = f + \epsilon_0$:

$$\begin{aligned}
\mathbb{E}[(y_0 - \mathbb{E}[\hat{f}])^2] &= \mathbb{E}\left[\left((f - \mathbb{E}[\hat{f}]) + \epsilon_0\right)^2\right] \\
&= (f - \mathbb{E}[\hat{f}])^2 + 2(f - \mathbb{E}[\hat{f}])\underbrace{\mathbb{E}[\epsilon_0]}_{=0} + \underbrace{\mathbb{E}[\epsilon_0^2]}_{=\sigma^2} \\
&= \underbrace{(f - \mathbb{E}[\hat{f}])^2}_{\text{Bias}^2} + \sigma^2
\end{aligned}$$

Combining all terms:

$$\mathbb{E}[(y_0 - \hat{f})^2] = \text{Bias}^2(\hat{f}) + \text{Var}(\hat{f}) + \sigma^2 \qquad \square$$

Component 1: Irreducible Error

Definition: Irreducible Error

$$\sigma^2 = \text{Var}(\epsilon) = \mathbb{E}[\epsilon^2]$$

A **uniform lower bound** on the expected prediction error for *any* estimator $\hat{f}$.

- Inherent to the data-generating process
- Arises from measurement error, unmodeled variables, intrinsic randomness
- **No model can eliminate it**—the price of modeling a stochastic world

Component 2: Bias

Definition: Bias

$$\text{Bias}(\hat{f}(x_0)) = \mathbb{E}_D[\hat{f}(x_0)] - f(x_0)$$

$$\text{Bias}^2(\hat{f}(x_0)) = \left(\mathbb{E}_D[\hat{f}(x_0)] - f(x_0)\right)^2$$

Component 2: Bias

Definition: Bias

$$\text{Bias}(\hat{f}(x_0)) = \mathbb{E}_D[\hat{f}(x_0)] - f(x_0)$$

$$\text{Bias}^2(\hat{f}(x_0)) = \left(\mathbb{E}_D[\hat{f}(x_0)] - f(x_0)\right)^2$$

- Systematic error from the model's simplifying assumptions
- Discrepancy between the **average prediction** (over all possible datasets) and the **true value**
- High bias $\Rightarrow$ model is too rigid $\Rightarrow$ **underfitting**
- Example: fitting a linear model to a cubic relationship

Component 3: Variance

Definition: Variance of an Estimator

$$\text{Var}(\hat{f}(x_0)) = \mathbb{E}_D\left[\left(\hat{f}(x_0) - \mathbb{E}_D[\hat{f}(x_0)]\right)^2\right]$$

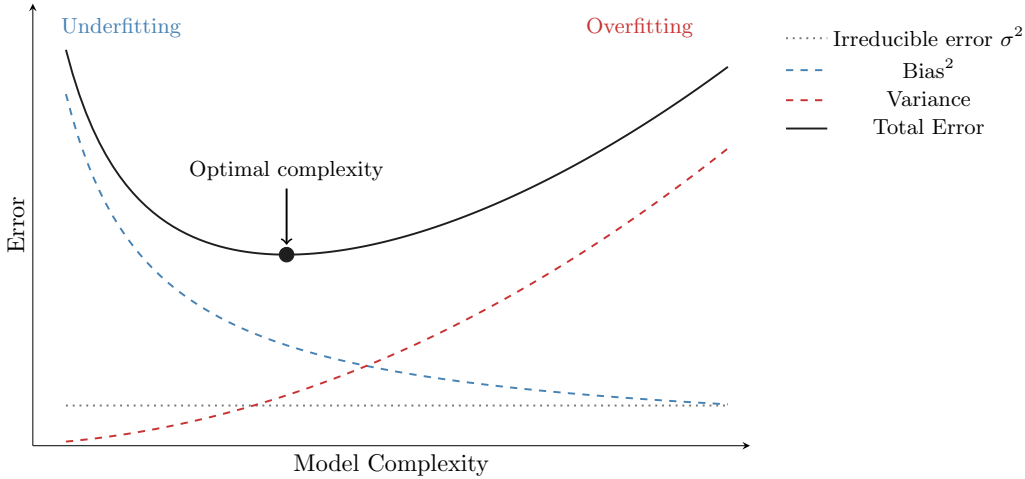
Component 3: Variance

Definition: Variance of an Estimator

$$\text{Var}(\hat{f}(x_0)) = \mathbb{E}_D\left[\left(\hat{f}(x_0) - \mathbb{E}_D[\hat{f}(x_0)]\right)^2\right]$$

- Sensitivity to the particular training dataset
- How much does $\hat{f}(x_0)$ **fluctuate** across different training samples?
- High variance $\Rightarrow$ model is too flexible $\Rightarrow$ **overfitting**
- Example: a 20th-degree polynomial changes shape dramatically with each training set

The Tradeoff Curve



A slightly biased model can have lower total error than an unbiased but high-variance one.

Why Simpler Models Can Win

The decomposition justifies preferring a simpler, slightly *wrong* model:

	Simple Model	**Complex Model**
Bias^2	High	Low
Variance	Low	High
Total Reducible Error	Can be lower	Can be higher

Why Simpler Models Can Win

The decomposition justifies preferring a simpler, slightly *wrong* model:

	Simple Model	**Complex Model**
Bias^2	High	Low
Variance	Low	High
Total Reducible Error	Can be lower	Can be higher

Regularization (Ridge, LASSO) intentionally introduces bias to reduce variance:

$$\text{Reducible Error} = \text{Bias}^2 + \text{Var}$$

The goal is not to eliminate bias, but to minimize the **sum**.

Mean Squared Error (MSE)

Definition: Mean Squared Error

Given n observations (y_i) and predictions $(\hat{y}_i)$:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Mean Squared Error (MSE)

Definition: Mean Squared Error

Given n observations (y_i) and predictions ($\hat{y}_i$):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- Direct empirical analogue of the theoretical expected prediction error
- OLS minimizes MSE on training data *by construction*
- **Limitation:** units are squared (e.g., $\$^2$)

Root Mean Squared Error (RMSE)

Definition: Root Mean Squared Error

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

- Same units as y — directly interpretable
- Represents the "typical magnitude" of prediction error

Root Mean Squared Error (RMSE)

Definition: Root Mean Squared Error

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

- Same units as y — directly interpretable
- Represents the "typical magnitude" of prediction error

Example

MSE $= 25{,}000{,}000$ \$^{2} is hard to interpret.
RMSE $=$ \$5,000 immediately tells you the typical error scale.

The Coefficient of Determination: Setup

MSE and RMSE give **absolute** error. We often want a **relative** measure: how much better is our model than a trivial baseline?

The Coefficient of Determination: Setup

MSE and RMSE give **absolute** error. We often want a **relative** measure: how much better is our model than a trivial baseline?

Definition: Total Sum of Squares

$$\mathrm{SS}_{\mathrm{tot}} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

Error of the baseline **mean-only model** $\hat{y}_i = \bar{y}$ for all i.

The Coefficient of Determination: Setup

MSE and RMSE give **absolute** error. We often want a **relative** measure: how much better is our model than a trivial baseline?

Definition: Total Sum of Squares

$$\mathrm{SS}_{\mathrm{tot}} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

Error of the baseline **mean-only model** $\hat{y}_i = \bar{y}$ for all i.

Definition: Residual Sum of Squares

$$\mathrm{SS}_{\mathrm{res}} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Variation left **unexplained** by the model.

The Coefficient of Determination (R^2)

Definition: Coefficient of Determination

$$\boxed{\mathrm{R}^2 = 1 - \frac{\mathrm{SS}_{\mathrm{res}}}{\mathrm{SS}_{\mathrm{tot}}}}$$

The Coefficient of Determination (R^2)

Definition: Coefficient of Determination

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

Interpretation: proportion of variance **explained** by the model.

- $R^2 = 1.0$: perfect fit ($SS_{res} = 0$)
- $R^2 = 0.0$: no better than predicting $\bar{y}$ ($SS_{res} = SS_{tot}$)
- $R^2 < 0$: **worse** than predicting $\bar{y}$

Critical Limitations of R^2

Lemma

R^2 is **non-decreasing** when a new predictor is added to a linear model.

Critical Limitations of R^2

Lemma

R^2 is **non-decreasing** when a new predictor is added to a linear model.

Proof.

Adding predictor X_2 enlarges the parameter space. OLS minimizes $\mathrm{SS}_{\mathrm{res}}$ over a *larger* set, so $\mathrm{SS}_{\mathrm{res}}$ can only decrease (or stay the same). Since $\mathrm{SS}_{\mathrm{tot}}$ is fixed, R^2 cannot decrease. □

Critical Limitations of R^2

Lemma

R^2 is **non-decreasing** when a new predictor is added to a linear model.

Proof.

Adding predictor X_2 enlarges the parameter space. OLS minimizes $\mathrm{SS}_{\mathrm{res}}$ over a *larger* set, so $\mathrm{SS}_{\mathrm{res}}$ can only decrease (or stay the same). Since $\mathrm{SS}_{\mathrm{tot}}$ is fixed, R^2 cannot decrease. □

Consequence: R^2 *actively encourages overfitting*—adding pure noise variables increases R^2.

Further Limitations of R^2

A high R^2 does **not** mean a good model:

- **Does not validate model assumptions:** A misspecified model (e.g., linear fit to nonlinear data) can still have high R^2

Further Limitations of R^2

A high R^2 does **not** mean a good model:

- **Does not validate model assumptions:** A misspecified model (e.g., linear fit to nonlinear data) can still have high R^2
- **Does not measure predictive accuracy:** Two models with the same MSE can have very different R^2 values if $\mathrm{SS}_{\mathrm{tot}}$ differs

Further Limitations of R^2

A high R^2 does **not** mean a good model:

- **Does not validate model assumptions:** A misspecified model (e.g., linear fit to nonlinear data) can still have high R^2
- **Does not measure predictive accuracy:** Two models with the same MSE can have very different R^2 values if $\mathrm{SS}_{\mathrm{tot}}$ differs
- **Not comparable across transformations:** R^2 for predicting y vs. $\log(y)$ are on different scales

Further Limitations of R^2

A high R^2 does **not** mean a good model:

- **Does not validate model assumptions:** A misspecified model (e.g., linear fit to nonlinear data) can still have high R^2
- **Does not measure predictive accuracy:** Two models with the same MSE can have very different R^2 values if $\mathrm{SS}_{\mathrm{tot}}$ differs
- **Not comparable across transformations:** R^2 for predicting y vs. $\log(y)$ are on different scales

The uncritical pursuit of high R^2 is a common anti-pattern in data analysis.

Adjusted R^2

Definition: Adjusted R^2

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

where n = number of samples, p = number of predictors.

Adjusted R^2

Definition: Adjusted R^2

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

where n = number of samples, p = number of predictors.

- Penalizes model complexity through the ratio $\frac{n-1}{n-p-1}$
- Only increases if the new variable reduces SS_{res} *enough* to overcome the penalty
- More suitable for comparing models with **different numbers of predictors**
- Rewards **parsimony**

The Pitfall of Accuracy

Definition: Accuracy

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

The Pitfall of Accuracy

Definition: Accuracy

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

Example: Credit Card Fraud Detection

Dataset: 1,000,000 transactions, 100 fraudulent (0.01%).

A model that *always predicts "not fraud"*:

$$\text{Accuracy} = \frac{999{,}900}{1{,}000{,}000} = 99.99\%$$

The Pitfall of Accuracy

Definition: Accuracy

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

Example: Credit Card Fraud Detection

Dataset: 1,000,000 transactions, 100 fraudulent (0.01%).

A model that *always predicts "not fraud"*:

$$\text{Accuracy} = \frac{999{,}900}{1{,}000{,}000} = 99.99\%$$

Yet it detects **zero** fraud cases—completely useless!

The Confusion Matrix

Definition: Confusion Matrix

For binary classification with true labels $y_i \in \{0, 1\}$ and predictions $\hat{y}_i \in \{0, 1\}$:

$$\text{TP} = |\{i : y_i = 1,\ \hat{y}_i = 1\}| \qquad \text{FN} = |\{i : y_i = 1,\ \hat{y}_i = 0\}|$$
$$\text{FP} = |\{i : y_i = 0,\ \hat{y}_i = 1\}| \qquad \text{TN} = |\{i : y_i = 0,\ \hat{y}_i = 0\}|$$

The Confusion Matrix

Definition: Confusion Matrix

For binary classification with true labels $y_i \in \{0, 1\}$ and predictions $\hat{y}_i \in \{0, 1\}$:

$$\mathrm{TP} = |\{i : y_i = 1,\ \hat{y}_i = 1\}| \qquad \mathrm{FN} = |\{i : y_i = 1,\ \hat{y}_i = 0\}|$$
$$\mathrm{FP} = |\{i : y_i = 0,\ \hat{y}_i = 1\}| \qquad \mathrm{TN} = |\{i : y_i = 0,\ \hat{y}_i = 0\}|$$

	Predicted +	**Predicted −**
Actual +	TP	FN
Actual −	FP	TN

$$\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN} = n \quad \text{and} \quad \text{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{n}$$

Confusion Matrix: Medical Test Example

Example

$n = 200$ patients: 50 sick, 150 healthy. Model predicts positive for 60.

	Predicted +	**Predicted −**
Actual +	TP $= 40$	FN $= 10$
Actual −	FP $= 20$	TN $= 130$

Confusion Matrix: Medical Test Example

Example

$n = 200$ patients: 50 sick, 150 healthy. Model predicts positive for 60.

	Predicted +	**Predicted −**
Actual +	TP $= 40$	FN $= 10$
Actual −	FP $= 20$	TN $= 130$

- 40/50 sick patients correctly identified
- 20 false alarms among healthy patients
- Verify: $40 + 130 + 20 + 10 = 200$

Precision

Definition: Precision

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

"Of all instances predicted positive, what fraction was correct?"

Precision

Definition: Precision

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

"Of all instances predicted positive, what fraction was correct?"

Example

$$\text{Precision} = \frac{40}{40 + 20} = \frac{2}{3} \approx 0.667$$

Only 2/3 of flagged patients actually have the disease. The remaining 1/3 are false alarms.

Precision

Definition: Precision

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

"Of all instances predicted positive, what fraction was correct?"

Example

$$\text{Precision} = \frac{40}{40 + 20} = \frac{2}{3} \approx 0.667$$

Only 2/3 of flagged patients actually have the disease. The remaining 1/3 are false alarms.

Optimize for precision when FP is costly (e.g., spam filter—don't lose legitimate emails).

Recall

Definition: Recall

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

"Of all actual positives, what fraction did the model identify?"

Recall

Definition: Recall

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

"Of all actual positives, what fraction did the model identify?"

Example

$$\text{Recall} = \frac{40}{40 + 10} = \frac{4}{5} = 0.80$$

The model catches 80% of sick patients, but misses 10 (1 in 5 go undiagnosed).

Recall

Definition: Recall

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

"Of all actual positives, what fraction did the model identify?"

Example

$$\text{Recall} = \frac{40}{40 + 10} = \frac{4}{5} = 0.80$$

The model catches 80% of sick patients, but misses 10 (1 in 5 go undiagnosed).

Optimize for recall when FN is costly (e.g., disease screening—don't miss sick patients).

The Precision-Recall Tradeoff

Most classifiers output a score $p(y{=}1|x)$, thresholded at τ:

$$\hat{y} = \begin{cases} 1 & \text{if } p(y{=}1|x) > \tau \\ 0 & \text{otherwise} \end{cases}$$

The Precision-Recall Tradeoff

Most classifiers output a score $p(y{=}1|x)$, thresholded at τ:

$$\hat{y} = \begin{cases} 1 & \text{if } p(y{=}1|x) > \tau \\ 0 & \text{otherwise} \end{cases}$$

Increase τ (e.g., 0.9):

- More conservative
- Fewer FP $\Rightarrow$ **higher precision**
- More FN $\Rightarrow$ **lower recall**

Decrease τ (e.g., 0.1):

- More liberal
- Fewer FN $\Rightarrow$ **higher recall**
- More FP $\Rightarrow$ **lower precision**

The Precision-Recall Tradeoff

Most classifiers output a score $p(y{=}1|x)$, thresholded at τ:

$$\hat{y} = \begin{cases} 1 & \text{if } p(y{=}1|x) > \tau \\ 0 & \text{otherwise} \end{cases}$$

Increase τ (e.g., 0.9):

- More conservative
- Fewer FP $\Rightarrow$ **higher precision**
- More FN $\Rightarrow$ **lower recall**

Decrease τ (e.g., 0.1):

- More liberal
- Fewer FN $\Rightarrow$ **higher recall**
- More FP $\Rightarrow$ **lower precision**

The optimal threshold is a **domain decision**, not a purely statistical one.

Why the Harmonic Mean?

Lemma: Property of the Harmonic Mean

For positive a, b:

$$H(a,b) = \frac{2ab}{a+b}$$

is always closer to $\min\{a,b\}$ than the arithmetic mean $A(a,b) = \frac{a+b}{2}$.

Why the Harmonic Mean?

Lemma: Property of the Harmonic Mean

For positive a, b:

$$H(a, b) = \frac{2ab}{a + b}$$

is always closer to $\min\{a, b\}$ than the arithmetic mean $A(a, b) = \frac{a+b}{2}$.

Why this matters:

- The harmonic mean is **low if either input is low**
- A model cannot score well by having perfect recall but terrible precision
- This makes it ideal for combining Precision and Recall

The F_1-Score

Definition: F_1-Score

The harmonic mean of Precision and Recall:

$$\boxed{\mathrm{F}_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}}$$

The F_1-Score

Definition: F_1-Score

The harmonic mean of Precision and Recall:

$$\boxed{F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}}$$

Example

With Precision $= 2/3$ and Recall $= 4/5$:

$$F_1 = \frac{2 \cdot \frac{2}{3} \cdot \frac{4}{5}}{\frac{2}{3} + \frac{4}{5}} = \frac{\frac{16}{15}}{\frac{22}{15}} = \frac{16}{22} = \frac{8}{11} \approx 0.727$$

The F_1-Score

Definition: F_1-Score

The harmonic mean of Precision and Recall:

$$\boxed{\mathrm{F}_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}}$$

Example

With Precision $= 2/3$ and Recall $= 4/5$:

$$\mathrm{F}_1 = \frac{2 \cdot \frac{2}{3} \cdot \frac{4}{5}}{\frac{2}{3} + \frac{4}{5}} = \frac{\frac{16}{15}}{\frac{22}{15}} = \frac{16}{22} = \frac{8}{11} \approx 0.727$$

Compare: arithmetic mean ≈ 0.733. The F_1 is pulled toward the **lower** value.

The F_β-Score

Definition: F_β-Score

For $\beta > 0$, the weighted harmonic mean:

$$\mathrm{F}_\beta = \frac{(1+\beta^2) \cdot \text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}}$$

The F_β-Score

Definition: F_β-Score

For $\beta > 0$, the weighted harmonic mean:

$$\mathrm{F}_\beta = \frac{(1+\beta^2) \cdot \text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}}$$

- $\beta = 1$: equal weight $\Rightarrow$ recovers F_1
- $\beta > 1$: **recall weighted more** (penalizes FN)
- $\beta < 1$: **precision weighted more** (penalizes FP)

F_β-Score: Choosing β by Context

With Precision $= 2/3$, Recall $= 4/5$:

Metric	Value	Use Case
$F_{0.5}$	≈ 0.690	Spam filter (precision matters)
F_1	≈ 0.727	Balanced
F_2	≈ 0.769	Disease screening (recall matters)

F_β-Score: Choosing β by Context

With Precision $= 2/3$, Recall $= 4/5$:

Metric	Value	Use Case
$F_{0.5}$	≈ 0.690	Spam filter (precision matters)
F_1	≈ 0.727	Balanced
F_2	≈ 0.769	Disease screening (recall matters)

Since Recall $>$ Precision for this model:

- Higher β rewards the model's stronger recall $\Rightarrow$ higher score
- Lower β penalizes the model's weaker precision $\Rightarrow$ lower score

The ROC Curve

Definition: ROC Curve

Plot of TPR vs. FPR as threshold τ varies:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \quad \mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}$$

The ROC Curve

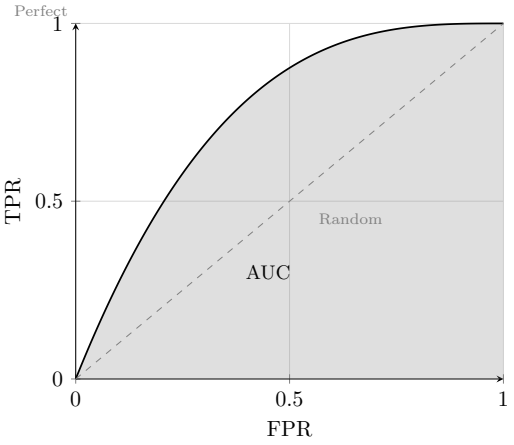
Definition: ROC Curve

Plot of TPR vs. FPR as threshold τ varies:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

Key reference points:

- $\tau = 1$: all negative $\Rightarrow (0, 0)$
- $\tau = 0$: all positive $\Rightarrow (1, 1)$
- Diagonal = random guessing
- Top-left corner = perfect



Area Under the ROC Curve (AUC)

Definition: AUC

$$\mathrm{AUC} = \int_0^1 \mathrm{TPR}\big(\mathrm{FPR}^{-1}(t)\big)\, dt$$

Area Under the ROC Curve (AUC)

Definition: AUC

$$\mathrm{AUC} = \int_0^1 \mathrm{TPR}\big(\mathrm{FPR}^{-1}(t)\big)\, dt$$

Probabilistic interpretation:

$$\mathrm{AUC} = \mathbb{P}\big(X^+ > X^-\big)$$

where X^+ and X^- are scores of a random positive and negative instance.

Area Under the ROC Curve (AUC)

Definition: AUC

$$\mathrm{AUC} = \int_0^1 \mathrm{TPR}\big(\mathrm{FPR}^{-1}(t)\big)\, dt$$

Probabilistic interpretation:

$$\mathrm{AUC} = \mathbb{P}\big(X^+ > X^-\big)$$

where X^+ and X^- are scores of a random positive and negative instance.

AUC	**Interpretation**
1.0	Perfect classifier
0.5	Random guessing (no discriminative power)
< 0.5	Worse than random (invert predictions)

ROC Curve: Worked Example

Example

10 patients, $P = 5$, $N = 5$, sorted by score:

τ	TP	FP	TPR	FPR
> 0.95	0	0	0.0	0.0
0.95	1	0	0.2	0.0
0.90	2	0	0.4	0.0
0.82	2	1	0.4	0.2
0.65	4	1	0.8	0.2
0.40	5	2	1.0	0.4
≤ 0.10	5	5	1.0	1.0

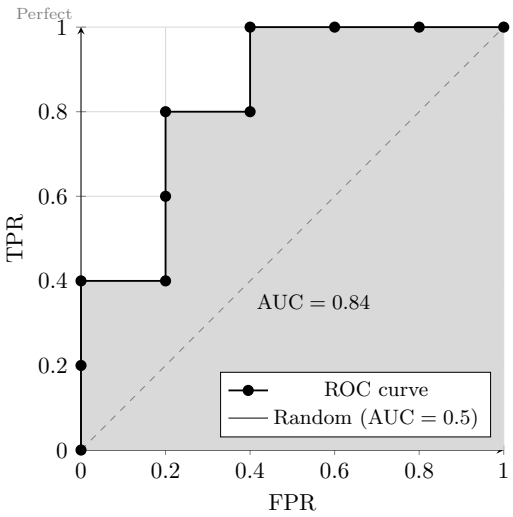
ROC Curve: Worked Example

Example

10 patients, $P = 5$, $N = 5$, sorted by score:

τ	TP	FP	TPR	FPR
> 0.95	0	0	0.0	0.0
0.95	1	0	0.2	0.0
0.90	2	0	0.4	0.0
0.82	2	1	0.4	0.2
0.65	4	1	0.8	0.2
0.40	5	2	1.0	0.4
≤ 0.10	5	5	1.0	1.0

$\mathrm{AUC} = \mathbf{0.84}$



ROC vs. Precision-Recall Curves

- The ROC curve can be **overly optimistic** under class imbalance
- FPR uses $\mathrm{FP} + \mathrm{TN}$ as denominator—a large TN count masks many false positives

ROC vs. Precision-Recall Curves

- The ROC curve can be **overly optimistic** under class imbalance
- FPR uses $\mathrm{FP} + \mathrm{TN}$ as denominator—a large TN count masks many false positives
- Precision-Recall curves **ignore** TN entirely
- Preferred when the positive class is rare and FP cost relative to TP matters

ROC vs. Precision-Recall Curves

- The ROC curve can be **overly optimistic** under class imbalance
- FPR uses $\mathrm{FP} + \mathrm{TN}$ as denominator—a large TN count masks many false positives
- Precision-Recall curves **ignore** TN entirely
- Preferred when the positive class is rare and FP cost relative to TP matters

	ROC/AUC	**PR Curve**
Balanced classes	Appropriate	Appropriate
Imbalanced classes	Can be misleading	Preferred

The Flaw of Evaluating on Training Data

Training error is a **systematically optimistic** estimate of generalization error.

- A complex model can achieve near-perfect training scores yet fail on new data
- Training error $\neq$ generalization error
- We need a procedure to estimate performance on **unseen data**

The Simple Validation Set

Split data into two disjoint parts:

1. Train $\hat{f}$ on D_{train} (e.g., 80%)
2. Evaluate on D_{val} (e.g., 20%)

The Simple Validation Set

Split data into two disjoint parts:

1. Train $\hat{f}$ on D_{train} (e.g., 80%)
2. Evaluate on D_{val} (e.g., 20%)

Two critical drawbacks:

1. **High variance:** The estimate depends heavily on the random split
2. **Data inefficiency:** The model is trained on less data $\Rightarrow$ systematically worse

Validation Set: Instability Example

Example

Dataset: $n = 10$, linear model. Two different 80/20 splits:

Split A (val $= \{(5, 11), (6, 12)\}$):

$$\mathrm{MSE}^{(A)} \approx 0.71$$

Validation Set: Instability Example

Example

Dataset: $n = 10$, linear model. Two different 80/20 splits:

Split A (val $= \{(5, 11), (6, 12)\}$):

$$\mathrm{MSE}^{(A)} \approx 0.71$$

Split B (val $= \{(9, 19), (10, 24)\}$):

$$\mathrm{MSE}^{(B)} \approx 8.35$$

Validation Set: Instability Example

Example

Dataset: $n = 10$, linear model. Two different 80/20 splits:

Split A (val $= \{(5, 11), (6, 12)\}$):

$$\mathrm{MSE}^{(A)} \approx 0.71$$

Split B (val $= \{(9, 19), (10, 24)\}$):

$$\mathrm{MSE}^{(B)} \approx 8.35$$

A **12$\times$ difference** from the same model on the same data—the estimate is unreliable!

k-Fold Cross-Validation

Definition: k-Fold Cross-Validation

1. **Partition** D into k disjoint folds $D_1, \ldots, D_k$
2. **For each** $i = 1, \ldots, k$:
   - Train $\hat{f}_i$ on $D \setminus D_i$
   - Evaluate metric M_i on D_i
3. **Average:**

$$\boxed{\mathrm{CV}_k = \frac{1}{k} \sum_{i=1}^{k} M_i}$$

k-Fold Cross-Validation

Definition: k-Fold Cross-Validation

1. **Partition** D into k disjoint folds $D_1, \ldots, D_k$
2. **For each** $i = 1, \ldots, k$:
   - Train $\hat{f}_i$ on $D \setminus D_i$
   - Evaluate metric M_i on D_i
3. **Average:**

$$\boxed{\mathrm{CV}_k = \frac{1}{k} \sum_{i=1}^{k} M_i}$$

Every data point is used for validation **exactly once**. Common choices: $k = 5$ or $k = 10$.

k-Fold CV: Visual Representation

$$\mathrm{CV}_k = \frac{1}{k} \sum_{i=1}^{k} M_i$$

5-Fold CV: Worked Example

Example

Same dataset ($n = 10$), linear model, 5 folds of size 2:

Fold	Validation set	OLS on remaining	M_i
1	$\{(1,3),\ (2,5)\}$	$\hat{y} \approx -0.86 + 2.29x$	2.06
2	$\{(3,7),\ (4,8)\}$	$\hat{y} \approx 0.29 + 2.14x$	0.40
3	$\{(5,11),\ (6,12)\}$	$\hat{y} \approx 0.29 + 2.15x$	0.71
4	$\{(7,15),\ (8,16)\}$	$\hat{y} \approx 0.08 + 2.21x$	1.68
5	$\{(9,19),\ (10,24)\}$	$\hat{y} \approx 1.11 + 1.89x$	8.35

5-Fold CV: Worked Example

Example

Same dataset ($n = 10$), linear model, 5 folds of size 2:

Fold	Validation set	OLS on remaining	M_i
1	$\{(1,3),\ (2,5)\}$	$\hat{y} \approx -0.86 + 2.29x$	2.06
2	$\{(3,7),\ (4,8)\}$	$\hat{y} \approx 0.29 + 2.14x$	0.40
3	$\{(5,11),\ (6,12)\}$	$\hat{y} \approx 0.29 + 2.15x$	0.71
4	$\{(7,15),\ (8,16)\}$	$\hat{y} \approx 0.08 + 2.21x$	1.68
5	$\{(9,19),\ (10,24)\}$	$\hat{y} \approx 1.11 + 1.89x$	8.35

$$\mathrm{CV}_5 = \frac{2.06 + 0.40 + 0.71 + 1.68 + 8.35}{5} = \mathbf{2.64}$$

More representative than either Split A (0.71) or Split B (8.35).

Bias and Variance of the CV Estimator

The CV score CV_k is itself an **estimator**—it has its own bias and variance.

Bias of CV_k:

- Each fold trains on $\frac{k-1}{k} \cdot n$ data
- Less data $\Rightarrow$ worse model $\Rightarrow$ pessimistic estimate
- Small k: high bias
- Large k: low bias

Variance of CV_k:

- Large k: training sets highly overlap
- Models $\hat{f}_i$ are correlated
- Averaging correlated estimates $\Rightarrow$ variance doesn't decrease much
- Large k: high variance

Bias and Variance of the CV Estimator

The CV score CV_k is itself an **estimator**—it has its own bias and variance.

Bias of CV_k:

- Each fold trains on $\frac{k-1}{k} \cdot n$ data
- Less data $\Rightarrow$ worse model $\Rightarrow$ pessimistic estimate
- Small k: high bias
- Large k: low bias

Variance of CV_k:

- Large k: training sets highly overlap
- Models $\hat{f}_i$ are correlated
- Averaging correlated estimates $\Rightarrow$ variance doesn't decrease much
- Large k: high variance

A **meta-level** bias-variance tradeoff in the choice of k!

Leave-One-Out Cross-Validation (LOOCV)

Definition: LOOCV

Special case $k = n$: each fold is a single observation.

$$\mathrm{CV}_n = \frac{1}{n}\sum_{i=1}^{n} e_i, \qquad e_i = L\big(y_i,\ \hat{f}_{-i}(x_i)\big)$$

where $\hat{f}_{-i}$ is trained on all data except (x_i, y_i).

Leave-One-Out Cross-Validation (LOOCV)

Definition: LOOCV

Special case $k = n$: each fold is a single observation.

$$\mathrm{CV}_n = \frac{1}{n}\sum_{i=1}^{n} e_i, \qquad e_i = L\big(y_i,\ \hat{f}_{-i}(x_i)\big)$$

where $\hat{f}_{-i}$ is trained on all data except (x_i, y_i).

- Training size $= n - 1 \Rightarrow$ approximately **zero bias**
- Training sets overlap in $n - 2$ points $\Rightarrow$ **high variance**
- Requires training n models $\Rightarrow$ **computationally expensive**

Stratified k-Fold Cross-Validation

Definition: Stratified k-Fold CV

Variant where each fold preserves the **class proportions** of D:

$$\frac{|\{(x_i, y_i) \in D_j : y_i = c\}|}{|D_j|} \approx \frac{n_c}{n} \quad \text{for all folds } j \text{ and classes } c.$$

Stratified k-Fold Cross-Validation

Definition: Stratified k-Fold CV

Variant where each fold preserves the **class proportions** of D:

$$\frac{|\{(x_i, y_i) \in D_j : y_i = c\}|}{|D_j|} \approx \frac{n_c}{n} \quad \text{for all folds } j \text{ and classes } c.$$

Example

$n = 100$: 80 class-0, 20 class-1 (80/20 split), $k = 5$.

- **Unstratified:** some folds might get 0–1 minority instances
- **Stratified:** each fold gets exactly 16 class-0 and 4 class-1

Essential for **imbalanced classification** to get stable metric estimates.

Comparison of CV Methods

	Validation	**5/10-Fold**	**LOOCV**	**Stratified**
Training Size	$0.8n$	$\frac{k-1}{k}n$	$n-1$	$\frac{k-1}{k}n$
Bias	High	Moderate	Low	Moderate
Variance	High	Moderate	High	Moderate
Cost	Low	Moderate	High	Moderate
Best For	Large data	General	Small data	Imbalanced

Comparison of CV Methods

	Validation	**5/10-Fold**	**LOOCV**	**Stratified**
Training Size	$0.8n$	$\frac{k-1}{k}n$	$n-1$	$\frac{k-1}{k}n$
Bias	High	Moderate	Low	Moderate
Variance	High	Moderate	High	Moderate
Cost	Low	Moderate	High	Moderate
Best For	Large data	General	Small data	Imbalanced

Default recommendation: 5- or 10-fold CV (stratified for classification).

CV for Model Selection

Example: Model Selection

Choose among polynomial degrees using 5-fold CV:

	M_1	M_2	M_3	M_4	M_5	CV_5
Linear ($d = 1$)	2.06	0.40	0.71	1.68	8.35	2.64
Quadratic ($d = 2$)	0.52	0.18	0.33	0.41	1.06	**0.50**
Cubic ($d = 3$)	0.61	0.25	0.40	0.55	1.89	0.74

CV for Model Selection

Example: Model Selection

Choose among polynomial degrees using 5-fold CV:

	M_1	M_2	M_3	M_4	M_5	CV_5
Linear ($d = 1$)	2.06	0.40	0.71	1.68	8.35	2.64
Quadratic ($d = 2$)	0.52	0.18	0.33	0.41	1.06	**0.50**
Cubic ($d = 3$)	0.61	0.25	0.40	0.55	1.89	0.74

- Linear underfits; cubic overfits; **quadratic wins**
- Training MSE would favor the cubic model—CV correctly identifies overfitting!

CV for Hyperparameter Tuning

Example: Tuning Polynomial Degree

Test $d \in \{1, 2, 3, 4, 5\}$:

Degree d	CV_5 (MSE)
1	2.64
2	**0.50**
3	0.74
4	1.35
5	3.92

CV for Hyperparameter Tuning

Example: Tuning Polynomial Degree

Test $d \in \{1, 2, 3, 4, 5\}$:

Degree d	CV_5 (MSE)
1	2.64
2	**0.50**
3	0.74
4	1.35
5	3.92

- CV error is **U-shaped**: bias-variance tradeoff in action!
- Optimal: $d^* = 2$

CV for Hyperparameter Tuning

Example: Tuning Polynomial Degree

Test $d \in \{1, 2, 3, 4, 5\}$:

Degree d	CV_5 (MSE)
1	2.64
2	**0.50**
3	0.74
4	1.35
5	3.92

- CV error is **U-shaped**: bias-variance tradeoff in action!
- Optimal: $d^* = 2$

Crucial step: After selecting d^*, retrain on the *entire* dataset D using $d = d^*$.

Key Takeaways

1. **Bias-Variance Decomposition:**

$$\mathbb{E}[(y_0 - \hat{f}(x_0))^2] = \text{Bias}^2 + \text{Var} + \sigma^2$$

   Error = systematic assumptions + instability + irreducible noise

2. **Regression Metrics:** MSE/RMSE measure absolute error; R^2 measures relative explained variance but *inflates with complexity*—use $\mathrm{R}^2_{\text{adj}}$

3. **Classification Metrics:** Accuracy fails under imbalance. Use the confusion matrix to derive Precision, Recall, and F_β. AUC provides threshold-independent evaluation

4. **Cross-Validation:** Never evaluate on training data. *k*-Fold CV provides a robust, nearly unbiased estimate of generalization error

5. **After selecting a model via CV:** Retrain on the *full* dataset before deployment