
Why Fitting Examples Can Mislead Us

ECE 57000 — September 25, 2026

David I. Inouye

| Wednesday connected **prediction goals, local evidence, and evaluation**

- We defined the prediction contract using only information available when the prediction is made.
- Under an explicit i.i.d. working assumption, KNN uses nearby historical outcomes to estimate a local outcome probability.
- Training error scores evidence the predictor already used; a holdout creates untouched evidence that can stand in for unseen cases.

Today: use held-out evidence to understand how model flexibility, data, and representation affect prediction on unseen cases.

| A holdout lets observed data **play the role of unseen data**

We do not have outcomes for the genuinely unseen customers we ultimately care about. How can we estimate their error?



Reserve some observed pairs **before fitting**. Because their outcomes did not help construct f , predictions on them mimic the information boundary of an unseen case—under the same sampling assumptions.

| Cross-validation rotates **which fold is held out**

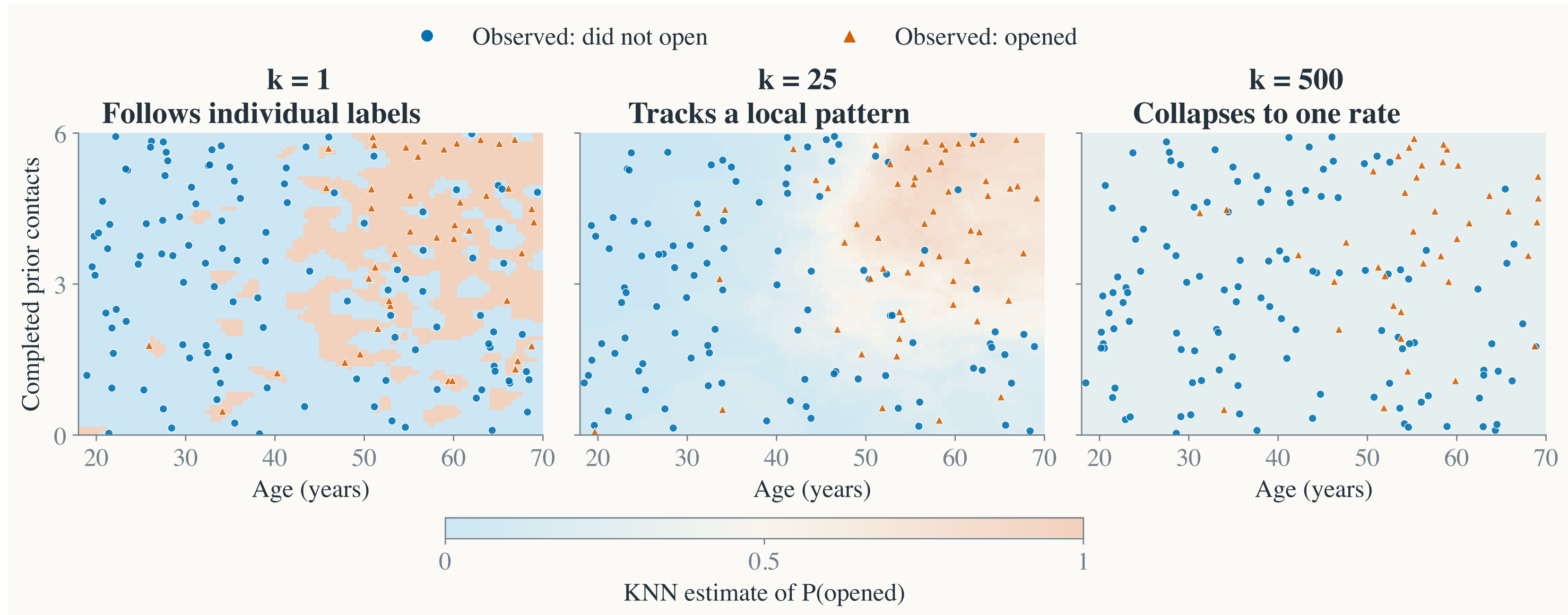
Split one observed dataset into folds. Each fold takes one turn as untouched evaluation data.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Fit 1	score e_1	train	train	train	train
Fit 2	train	score e_2	train	train	train
Fit 3	train	train	score e_3	train	train
Fit 4	train	train	train	score e_4	train
Fit 5	train	train	train	train	score e_5

K fits produce K errors, then we average: $\widehat{R}_{CV} = \frac{1}{K} \sum_{j=1}^K e_j$

Leave-one-out cross-validation is the small-data extreme: $K = n$.

| The neighbor count controls **how much local detail survives**



$k = 1$ can overfit individual outcomes; using every example erases useful local structure. Held-out error can reveal both failures.

| Bias compares the average fitted estimate with **the target relationship**

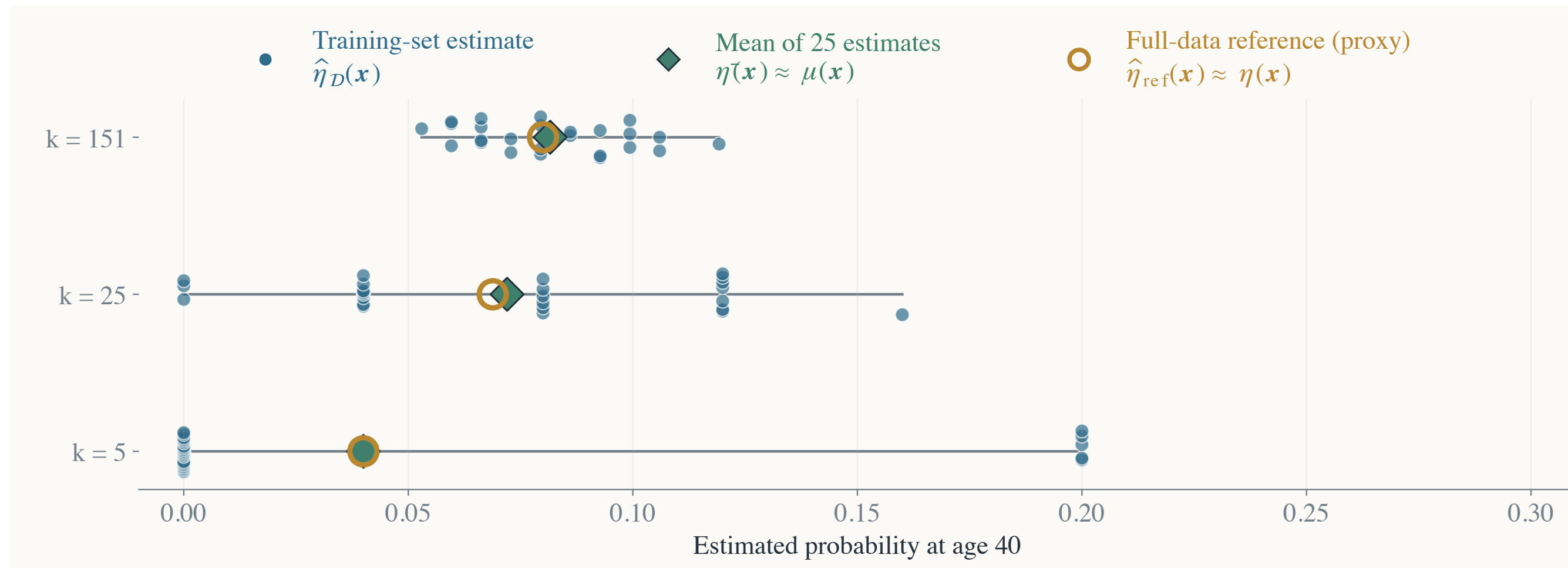
At one fixed test point $\boldsymbol{x}$, imagine refitting the method on many possible training samples. Their average prediction is where the method tends to land:

$$\mu(\boldsymbol{x}) = \mathbb{E}_{\mathcal{D}} [\hat{\eta}_{\mathcal{D}}(\boldsymbol{x})] , \quad \text{Bias}(\boldsymbol{x}) = \mu(\boldsymbol{x}) - \eta(\boldsymbol{x}).$$

Real data lets us observe how estimates vary across samples. It does not reveal the true $\eta(\boldsymbol{x})$, so formal bias remains unknown.

| Bias measures **offset**; variance measures **spread**

At **age 40** (the only feature), each blue dot uses a different disjoint training set $\mathcal{D}$ of $n = 1,647$ or $1,648$ records. Each row reuses the same 25 sets.



$$\text{Bias}(\mathbf{x}) = \mu(\mathbf{x}) - \eta(\mathbf{x})$$

$$\text{Var}_{\mathcal{D}}(\hat{\eta}_{\mathcal{D}}(\mathbf{x})) = \mathbb{E}_{\mathcal{D}} \left[(\hat{\eta}_{\mathcal{D}}(\mathbf{x}) - \mu(\mathbf{x}))^2 \right]$$

True η is unknown: the diamond-to-ring gap is a bias proxy. The reference may itself be biased.

| Expected squared error combines **bias, variance, and label noise**

Draw a new input $\mathbf{x}$ and binary outcome y from the target population, independently of the training set $\mathcal{D}$. How far is the predicted probability from the **actual outcome**, averaged over new cases and refits?

$$\underbrace{\mathbb{E}_{\mathcal{D}, \mathbf{x}, y} [(\hat{\eta}_{\mathcal{D}}(\mathbf{x}) - y)^2]}_{\text{expected squared prediction error}}$$
$$= \mathbb{E}_{\mathbf{x}} \left[\underbrace{\text{Bias}(\mathbf{x})^2}_{\text{systematic mismatch}} + \underbrace{\text{Var}_{\mathcal{D}}(\hat{\eta}_{\mathcal{D}}(\mathbf{x}))}_{\text{sample sensitivity}} + \underbrace{\eta(\mathbf{x})(1 - \eta(\mathbf{x}))}_{\text{unresolved outcome variation}} \right].$$

Squared error loss gives this exact sum. Modeling choices can trade bias against variance; the represented inputs determine the remaining outcome noise.

| Centering the outcome at η separates estimation error and noise

Fix $\mathbf{x}$. Write $\hat{\eta}_{\mathcal{D}} = \hat{\eta}_{\mathcal{D}}(\mathbf{x})$ and $\eta = \mathbb{E}[y \mid \mathbf{x}]$. The new outcome is independent of the training set given this input.

$$\begin{aligned} & \mathbb{E}_{\mathcal{D}, y}[(\hat{\eta}_{\mathcal{D}} - y)^2 \mid \mathbf{x}] \\ &= \mathbb{E}_{\mathcal{D}, y}[(\hat{\eta}_{\mathcal{D}} - \eta) + (\eta - y)]^2 \mid \mathbf{x}] && \text{(Add and subtract } \eta) \\ &= \mathbb{E}_{\mathcal{D}}[(\hat{\eta}_{\mathcal{D}} - \eta)^2] + \mathbb{E}_y[(\eta - y)^2 \mid \mathbf{x}] \\ &\quad + 2\mathbb{E}_{\mathcal{D}}[\hat{\eta}_{\mathcal{D}} - \eta] \underbrace{\mathbb{E}_y[\eta - y \mid \mathbf{x}]}_{\eta - \mathbb{E}[y \mid \mathbf{x}] = \eta - \eta = 0} && \text{(Expand; independence factors the cross term, and centering makes it zero)} \\ &= \mathbb{E}_{\mathcal{D}}[(\hat{\eta}_{\mathcal{D}} - \eta)^2] + \underbrace{\mathbb{E}[y^2 \mid \mathbf{x}] - \eta^2}_{\text{Var}(y \mid \mathbf{x})} && \text{(Expand the remaining outcome square)} \\ &= \underbrace{\mathbb{E}_{\mathcal{D}}[(\hat{\eta}_{\mathcal{D}} - \eta)^2]}_{\text{probability-estimation error}} + \underbrace{\eta(1 - \eta)}_{\text{outcome noise}} && \text{(Binary } y^2 = y, \text{ so } \eta - \eta^2 = \eta(1 - \eta)) \end{aligned}$$

| Centering the fit at μ separates **squared bias and variance**

At the same fixed x , let $\mu = \mathbb{E}_{\mathcal{D}}[\hat{\eta}_{\mathcal{D}}]$. Continue from the preceding slide, keeping its outcome-noise term.

$$\begin{aligned} & \mathbb{E}_{\mathcal{D}}[(\hat{\eta}_{\mathcal{D}} - \eta)^2] + \eta(1 - \eta) \\ &= \mathbb{E}_{\mathcal{D}}[((\hat{\eta}_{\mathcal{D}} - \mu) + (\mu - \eta))^2] + \eta(1 - \eta) \\ &= \mathbb{E}_{\mathcal{D}}[(\hat{\eta}_{\mathcal{D}} - \mu)^2] + (\mu - \eta)^2 + \eta(1 - \eta) \\ &\quad + 2(\mu - \eta) \underbrace{\mathbb{E}_{\mathcal{D}}[\hat{\eta}_{\mathcal{D}} - \mu]}_{\mathbb{E}_{\mathcal{D}}[\hat{\eta}_{\mathcal{D}}] - \mu = \mu - \mu = 0} \\ &= \underbrace{(\mu - \eta)^2}_{\text{squared bias}} + \underbrace{\mathbb{E}_{\mathcal{D}}[(\hat{\eta}_{\mathcal{D}} - \mu)^2]}_{\text{variance}} + \underbrace{\eta(1 - \eta)}_{\text{outcome noise}} \end{aligned}$$

(Add and subtract μ)

(Expand; $\mu - \eta$ is constant across training sets, and the centered fit has mean zero)

(The cross term is zero; reorder the remaining terms)

This exact sum uses **squared error loss**. Other losses need different analyses.

| Averaging over inputs gives **population squared prediction error**

Now draw a new input $\mathbf{x}$ from the target population and its outcome y . Average the pointwise identity over those inputs:

$$\underbrace{\mathbb{E}_{\mathcal{D}, \mathbf{x}, y}[(\hat{\eta}_{\mathcal{D}}(\mathbf{x}) - y)^2]}_{\text{average error on new cases, across refits}} \\ = \underbrace{\mathbb{E}_{\mathbf{x}}[\text{Bias}(\mathbf{x})^2]}_{\text{average squared bias}} + \underbrace{\mathbb{E}_{\mathbf{x}}[\text{Var}_{\mathcal{D}}(\hat{\eta}_{\mathcal{D}}(\mathbf{x}))]}_{\text{average variance}} + \underbrace{\mathbb{E}_{\mathbf{x}}[\eta(\mathbf{x})(1 - \eta(\mathbf{x}))]}_{\text{average outcome noise}}.$$

- Inputs count according to how often they occur in the **target population**. The decomposition stays exact, but its terms now summarize all inputs.
- Average **squared** bias: positive and negative biases at different inputs cannot cancel each other.

Why it matters: smoothing can reduce sample sensitivity while increasing systematic mismatch. Held-out error measures their combined effect, plus outcome noise.

| Increasing k exchanges local detail for stability

Smaller k	Larger k
uses closer evidence	reaches farther from the query
follows more sample-specific detail	averages more observed outcomes
often lower local bias, higher variance	often higher local bias, lower variance

Neither extreme is automatically best. Held-out error measures the combined effect for the population and representation we care about.

| Data and representation improve **the same local-evidence mechanism**

Change	What happens to a KNN neighborhood?
More representative training data	genuinely similar examples can be closer
Better scaling or features	distance favors differences relevant to the outcome
Larger k	more outcomes are averaged, but evidence comes from farther away

All three change the evidence inside $\hat{\eta}_k(\boldsymbol{x})$ and therefore its bias and variance.

4. How can we choose a model without fooling ourselves?

Choosing a predictor and measuring its performance are two different uses of data.

| Which neighbor count should we choose?

We fit KNN with $k = 1, 5, 25$ and choose the lowest **training error**.
Inputs are distinct, and each training point counts as its own neighbor.
Is this a reliable way to choose the best predictor for new cases?
A. Yes: evaluating every candidate on the same examples makes the comparison fair.
B. No: training error rewards reproducing outcomes the predictor already used.
C. No: we should always choose the largest k to get the most stable predictions.

B. Here, $k = 1$ gets zero training error by retrieving each point's own label. That does not establish how well it predicts a new outcome.

| A held-out **validation set** lets us compare candidate values of k

Fit each candidate using $\mathcal{D}_{\text{train}}$. Evaluate its predictions on the same separate $\mathcal{D}_{\text{validation}}$, then choose the lowest validation error.

$$\mathcal{D}_{\text{train}} \xrightarrow{\text{fit}} f_1, f_5, f_{25} \xrightarrow{\text{score on validation}} \hat{k} = \arg \min_{k \in \{1, 5, 25\}} \widehat{R}_{\text{validation}}(f_k).$$

The validation outcomes are unavailable when fitting each candidate. They give us evidence about prediction on cases outside that candidate's training set.

| What does the selected model's 4.5% error tell us?

We choose k with the lowest validation error and report its 4.5% as the selected model's estimated population error.

Across repetitions of this procedure, is the reported error lower on average, unbiased, or higher on average?

- A. Lower than the selected model's true population error on average.
- B. Unbiased for the selected model's true population error.
- C. Higher than the selected model's true population error on average.

A: optimistic selection bias. Choosing the lowest observed error also favors favorable validation noise. Validation appropriately helps choose k , but the winning score is no longer an unbiased final evaluation.

| Validation is held out of each fit, but participates in selection

What are we evaluating?

One candidate fitted at a fixed k

The selected pipeline, including the choice of k

Did validation outcomes help construct it?

No: only the training data determined that fit.

Yes: validation errors determined which candidate won.

Selection is part of learning. Information from validation enters through our choice of k , even though the KNN fitting step never reads those labels.

The same reuse occurs when validation guides features, preprocessing, or model families. It leaks into a claimed final evaluation through those choices.

| Separate validation for choosing from test data for reporting

Reserve all three roles **before development**:

Data	Role	What may it influence?
Training	Fit each candidate.	Its learned parameters or stored examples.
Validation	Compare candidates and choose k .	The selected pipeline.
Test	Score the settled pipeline once.	The reported performance estimate.

Freeze the pipeline before using the test outcomes. An independent, representative test set gives an **unbiased error estimate for that fixed pipeline**, with finite-sample uncertainty.