ENGSCI++

Scientific Machine Learning

From a trained model to a defensible claim

Aman Bhargava

First working edition — July 2026

For engineers who can already build systems and now need to make empirical claims that survive contact with statistics, messy data, and production.

Orientation

Audience

This book assumes that you can ship code. You are comfortable with functions, tests, APIs, data structures, version control, and debugging. You have seen probability, linear algebra, calculus, and basic machine learning, although the details may be rusty. The target reader is fast enough that a conventional introductory tour would be boring and experienced enough to be dangerous with a dataframe and a model-selection loop.

The objective is therefore not to catalogue algorithms. It is to make you good at answering a harder question:

Principle. What, exactly, does this experiment justify us in believing?

A model can compile, converge, score well, and still fail to support its stated claim. Scientific machine learning is the discipline of preventing that gap. It joins statistical reasoning, numerical optimization, data engineering, software architecture, and experimental design.

What this book means by "scientific"

The word does not mean academic, slow, or detached from products. A workflow is scientific when its claims are explicit, its evidence is generated by a process that could have falsified them, and its artifacts let another competent person audit or repeat the reasoning. This standard applies equally to a paper, an internal experiment, a medical classifier, an A/B test, or a model proposed for production.

The unit of quality is not the model file. It is the chain

$$\text{question} \to \text{estimand} \to \text{data} \to \text{split} \to \text{pipeline} \to \text{metric} \to \text{uncertainty} \to \text{claim}.$$

Every arrow is a possible source of invalidity. Strong engineering makes the chain inspectable; strong statistics makes its interpretation defensible.

How to read

Chapters 1–6 form the main argument. Chapter 7 applies it to an educational reconstruction of a biomedical classification capstone. Chapter 8 is a compact field guide for design and review. The appendix restores the mathematical facts most often needed in practice.

Boxes carry four recurring signals:

Principle	A compact invariant worth remembering.
Failure mode	A plausible workflow that produces misleading evidence.
Audit	A concrete question or check for an existing system.
Shipping rule	An implementation constraint that turns reasoning into durable software.

Source and reconstruction policy

The mathematical foundations draw on the original EngSci Abridged notes for matrix algebra, optimization, probabilistic reasoning, and information theory, cross-checked against standard references [1, 2, 3, 5, 14]. Repository projects are used as evidence and case material, not as a table of contents. Their clean-room ingest distinguishes committed sources, dirty working trees, stored notebook output, supplied course material, and later audit conclusions [4].

The biomedical case study is explicitly idealized. Historical design choices and observed failure modes are retained when educationally important, but the pipeline is reconstructed into the version that should have been built. Such changes are marked [idealized] and never presented as historical results.

The standard

This is a working edition, not an excuse for lower standards. Equations should be dimensionally coherent. Algorithms should state their invariants. Metrics should name their denominators. Experimental claims should identify the unit of independence. Figures and numbers should have provenance. Unknowns should remain unknown.

Shipping rule. If a result cannot be traced to a dataset version, split manifest, pipeline configuration, code revision, and metric definition, it is a rumor with decimals.

Contents

Chapter 1

Begin with the claim

1.1 Prediction is not the question

"Predict cancer," "classify text," and "forecast demand" are product descriptions, not experimental questions. Before touching a model, specify the objects that connect a system output to a real-world claim.

Definition 1.1 (Unit of observation)**.** The entity represented by one row or example: a patient, image, device, transaction, document, or time window.

Definition 1.2 (Unit of independence)**.** The smallest grouping that can reasonably be treated as independent for splitting and uncertainty. It is often larger than a row: patient rather than blood draw, user rather than event, machine rather than sensor window.

Definition 1.3 (Target population)**.** The population, environment, and time period over which the final claim is intended to hold.

Definition 1.4 (Estimand)**.** The quantity the experiment is designed to estimate. Examples include expected log loss for future patients at a named hospital, sensitivity at 95% specificity, or the change in conversion caused by a treatment policy.

An estimand includes more than a metric name. It fixes a population, prediction time, outcome horizon, intervention or observation regime, and loss. "AUC" is not an estimand until these are specified.

1.1.1 The prediction contract

A useful design artifact is a one-page contract:

Decision	What action will use the prediction, and who bears the error?
Unit	What entity receives one prediction? What entities are dependent?
Features	What information exists at prediction time?
Target	What event or quantity is observed, when, and how reliably?
Population	Where, when, and for whom should performance generalize?
Loss	What operational costs distinguish false positives, false negatives, delay, and abstention?
Evidence threshold	What result would justify proceeding, and what result would stop the project?

Audit. Read every feature as a timestamped fact. Could it genuinely exist at the instant the model is supposed to predict? If not, the experiment answers a different question.

1.2 Risk, samples, and generalization

Let observations $Z = (X, Y) \sim P$, model $f \in \mathcal{F}$, and loss $\ell(f(X), Y)$. The population risk is

$$R_P(f) = \mathbb{E}_{Z \sim P}[\ell(f(X), Y)]. \tag{1.1}$$

Given a sample $\mathcal{D} = \{z_i\}_{i=1}^n$, empirical risk is

$$\widehat{R}_{\mathcal{D}}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i). \tag{1.2}$$

Training minimizes some empirical objective. Science asks what that procedure tells us about R_Q, where Q is the target deployment distribution. If $P \neq Q$, more precise optimization of $\widehat{R}_P$ need not improve the claim about R_Q.

The learning algorithm is itself a function

$$\mathcal{A} : \mathcal{D} \mapsto f_{\mathcal{D}}.$$

Evaluation must therefore assess the complete data-dependent procedure, not a fixed function chosen independently of the test set. Feature selection, preprocessing, threshold choice, and hyperparameter search belong inside $\mathcal{A}$ [24].

Principle. The object under evaluation is the whole training procedure. A model selected after looking at test performance is no longer independent of that test set.

1.3 Association, prediction, and intervention

Predictive evidence supports statements of the form "this procedure predicts well under this distribution." It does not automatically identify causes or the effect of changing a variable.

Suppose hospital site S affects scanner calibration X and disease prevalence Y. A model may exploit S-encoded artifacts and predict Y within a mixed-site dataset. It does not follow that the measured biomarker is biologically informative, nor that performance transfers to a new site.

Three questions should not be conflated:

1. **Descriptive:** What patterns occur in the observed sample?
2. **Predictive:** How well does a specified procedure predict new observations from a target distribution?
3. **Causal:** What would happen under an intervention?

Each requires different assumptions. A high-capacity predictor can be useful without being causal; trouble starts when a predictive result is narrated as a mechanism [17].

1.4 Baselines and falsification

A baseline is not ceremonial. It is a rival explanation for the result. Useful baselines include:

- prevalence or historical mean;
- a small linear or logistic model;
- a policy already in operation;
- features known to be available and robust;
- a shuffled-label or broken-feature negative control;
- an oracle upper bound, clearly labeled and never confused with a deployable model.

If a complex model barely beats logistic regression, the correct result may be that representation complexity is unnecessary. Correcting leakage has erased apparent advantages of complex models in published scientific domains [13].

Failure mode. Choosing the baseline after seeing results turns it into rhetoric. Specify the comparison set and decision threshold before running the final evaluation.

1.5 Claim ladder

Evidence supports progressively stronger statements:

1. The code produced this stored output.
2. The implementation computes the intended statistic.
3. The statistic estimates performance for unseen units from the sampled population.
4. The sampled population represents the deployment population.
5. The model improves a decision after accounting for costs and workflow.
6. The improvement persists under monitoring and distribution change.

A single retrospective predictive experiment often establishes only levels 2–3. This is useful. The failure is not modest evidence; it is skipping levels without stating the additional assumptions.

Chapter 2

Data is part of the model

2.1 The observed table is generated

A dataset is not a neutral matrix. It is the output of sampling, sensors, human decisions, joins, missingness, filtering, labeling, and time. Write this as a data-generating process:

$$U \longrightarrow X^* \longrightarrow X, \qquad U \longrightarrow Y^* \longrightarrow Y,$$

where U contains latent state and selection effects, stars denote quantities of real interest, and (X, Y) are recorded proxies. The model only sees the right-hand side. Scientific validity depends on the arrows.

2.2 Rows, entities, events, and time

A common structural mistake is treating rows as independent when they share an entity or future information.

Let row i have group g_i and event time t_i. A group-safe split requires

$$\{g_i : i \in I_{\text{train}}\} \cap \{g_i : i \in I_{\text{test}}\} = \varnothing. \tag{2.1}$$

A time-safe split additionally respects the prediction clock: training facts must precede the period represented by evaluation. Random splitting is valid only when exchangeability across rows is a defensible approximation [19].

Data	Row	Independence unit	Usually defensible split
Clinical assays	specimen	patient	grouped, often site/time held out
Clickstream	event	user/session	grouped and chronological
Predictive maintenance	window	machine	machine-held-out plus forward time
Documents	paragraph	document/source	source-held-out
Images	crop	original scene/subject	group before augmentation

Failure mode. A random row split can place nearly identical measurements from one entity on both sides. The model is then tested on recognition of the entity, not generalization to a new entity.

2.3 Split before fit

Any operation that learns state from data is a model component:

- missing-value imputation;
- standardization or quantile transforms;
- vocabulary and feature selection;
- PCA, UMAP, or learned embeddings;
- resampling and synthetic-data generation;
- target encoding;
- threshold and hyperparameter selection.

Let T be a fitted transformation. Leakage-safe evaluation computes

$$\widehat{T} = \mathrm{fit}(T, \mathcal{D}_{\mathrm{train}}), \qquad X'_a = \mathrm{transform}(\widehat{T}, X_a), \quad a \in \{\mathrm{train}, \mathrm{val}, \mathrm{test}\}.$$

It never refits T using validation or test observations. "Unsupervised" does not make a transformation harmless: the held-out feature distribution is still information. A learned prototype, class-average waveform, dictionary, basis, channel choice, or normalization state is fitted state. For a new-subject claim, split subjects first and construct, for example,

$$T_{\mathrm{train}} = \frac{1}{|I_+|} \sum_{i \in I_+} x_i, \qquad I_+ = \{i \in I_{\mathrm{train}} : y_i = 1\}.$$

Held-out examples may be projected onto T_{train}; they may not define it.

Principle. Split before every fit, not merely before fitting the final classifier.

This rule is the simplest defense against a broad family of leakage errors documented across scientific ML [13].

2.3.1 Three nested information boundaries

1. **Outer test boundary:** protects the final performance estimate.
2. **Inner validation boundary:** protects model and hyperparameter selection.
3. **Training boundary:** confines fitted preprocessing and augmentation.

When data are scarce, cross-validation reuses observations across fits, but the information boundary still applies separately inside every fold.

2.4 Missingness is information

Let M denote the missingness pattern and partition the complete data into observed and missing components X_{obs} and X_{mis}. Classical missing-data terminology distinguishes the following mechanisms for the full missingness indicator under a specified full-data model [20]:

- **MCAR:** $M \perp (X_{\mathrm{obs}}, X_{\mathrm{mis}})$;
- **MAR:** $M \perp X_{\mathrm{mis}} \mid X_{\mathrm{obs}}$;
- **MNAR:** the MAR conditional independence does not hold.

The names are less important than the causal question: why is the field absent? In operational systems, missingness often encodes workflow, cost, access, or clinical suspicion. An imputer can convert this policy signal into a shortcut.

For a numeric feature, a transparent baseline is median imputation plus a missingness indicator. More sophisticated imputers are justified when they improve the target estimand under fold-local evaluation, not because their filled table looks plausible.

Audit. For every column, record type, unit, valid range, timestamp semantics, missingness mechanism, transformation, and whether it is available at prediction time. This is the feature contract.

2.5 Labels and proxies

Labels can be delayed, noisy, selectively observed, or defined by the same process that generates features. Let true outcome Y^* pass through a labeling channel $P(Y \mid Y^*, X, S)$. If that channel varies by site or demographic group, evaluation against Y measures agreement with the channel as well as the underlying outcome.

Useful checks include:

- duplicate-label consistency for the same entity and horizon;
- inter-rater agreement where multiple raters exist;
- label prevalence by time, site, and acquisition pathway;
- delayed-label censoring and differential follow-up;
- manual review of high-confidence disagreements.

2.6 Distribution shift

One useful factorization-based vocabulary for training–deployment mismatch is [18]:

Covariate shift: $P(X)$ changes while $P(Y \mid X)$ is stable.

Label shift: $P(Y)$ changes while $P(X \mid Y)$ is stable.

Concept shift: $P(Y \mid X)$ changes.

Selection shift: inclusion in the dataset depends differently on X, Y, or latent variables.

Terminology—especially "concept shift"—varies across fields; the marginal and conditional factorizations displayed above are the definitions used here.

These are assumptions about distributions, not diagnoses produced by a drift dashboard. A held-out hospital, device, geography, or future period often tests the intended claim more directly than another random fold.

2.7 Dataset contracts and immutable manifests

Shipping rule. Persist split membership by stable entity ID. A random seed alone is not a split manifest: library versions, ordering, filtering, and upstream data can all change the realized partition.

Datasheets motivate systematic documentation of a dataset's motivation, composition, collection, uses, distribution, and maintenance [11]. The immutable, executable fields below are EngSci++ house policy for connecting that documentation to a build.

A reproducible dataset version should be an immutable manifest rather than a mutable folder name. At minimum record:

- source versions and cryptographic hashes;
- schema, units, keys, and uniqueness constraints;
- row counts before and after every filter/join;
- label definition and cutoff time;
- entity/group identifier and split assignment;
- code revision and configuration producing the table, together with rights, consent, retention, and access constraints.

Chapter 3

Models, objectives, and optimization

3.1 A model is a constrained explanation

A useful model combines a representation, an objective, an optimization procedure, and assumptions about uncertainty. Separate these explicitly:

$$x \xrightarrow{\phi} z \xrightarrow{f_\theta} \widehat{y}, \qquad \widehat{\theta} = \arg\min_\theta \left[\frac{1}{n} \sum_{i=1}^{n} \ell(f_\theta(\phi(x_i)), y_i) + \lambda \Omega(\theta) \right].$$

The representation ϕ, loss ℓ, regularizer Ω, and search procedure all encode inductive bias. "We used a neural network" omits most of the procedure.

3.2 Likelihood, loss, and regularization

For a probabilistic model $p_\theta(y \mid x)$, maximum likelihood minimizes negative log likelihood:

$$\widehat{\theta}_{\mathrm{ML}} = \arg\min_\theta - \sum_{i=1}^{n} \log p_\theta(y_i \mid x_i). \tag{3.1}$$

With prior $p(\theta)$, maximum a posteriori estimation adds $-\log p(\theta)$. A zero-mean isotropic Gaussian prior yields an ℓ_2 penalty; a Laplace prior yields ℓ_1. Regularization is therefore both a geometric constraint and a probabilistic assumption [14, 2].

For linear regression with $X \in \mathbb{R}^{n \times d}$ and $\lambda > 0$, ridge regression solves

$$\widehat{\beta}_\lambda = \arg\min_\beta \frac{1}{2} \|X\beta - y\|_2^2 + \frac{\lambda}{2} \|\beta\|_2^2, \qquad \widehat{\beta}_\lambda = (X^T X + \lambda I)^{-1} X^T y. \tag{3.2}$$

In code, solve the linear system; do not form the inverse. The regularized matrix improves conditioning, but scaling and rank still matter [1, 5].

For binary logistic regression, $p_i = \sigma(x_i^T \beta)$, and

$$\ell_i(\beta) = -y_i \log p_i - (1 - y_i) \log(1 - p_i). \tag{3.3}$$

This is a proper probabilistic loss [12]. Classification accuracy is not a suitable training objective because it is discontinuous and discards confidence.

3.3 Optimization is part of correctness

Gradient descent updates

$$\theta_{k+1} = \theta_k - \alpha_k \nabla L(\theta_k).$$

For convex L with L_g-Lipschitz gradient, a fixed step $0 < \alpha \leq 1/L_g$ gives a standard descent guarantee. For least squares, $L_g = \lambda_{\max}(X^T X)$ [5]. For a closed convex constraint set C, projected gradient descent uses

$$\theta_{k+1} = \Pi_C(\theta_k - \alpha_k \nabla L(\theta_k)) .$$

The wave-one MIE424 implementation provides a memorable audit case: its projectors are correct, but the least-squares gradient uses $X^T X\beta + X^T y$ instead of $X^T X\beta - X^T y$, reversing a critical term [4]. Elegant experimental scaffolding cannot rescue the wrong derivative.

Audit. For every handwritten gradient, compare dimensions, run central finite differences on a tiny random problem, and test a point with a known optimum.

For coordinate j, central difference checking uses

$$g_j^{\text{num}} = \frac{L(\theta + \epsilon e_j) - L(\theta - \epsilon e_j)}{2\epsilon}. \tag{3.4}$$

Compare with a scale-aware error such as $\|g - g^{\text{num}}\|/(\|g\| + \|g^{\text{num}}\| + \delta)$.

3.4 Capacity, bias, and variance

Increasing capacity can reduce approximation bias while increasing sensitivity to the sampled data and to optimization. The useful question is not whether a model is "powerful," but whether its inductive bias and sample complexity fit the evidence available.

Learning curves separate common regimes:

- high and similar train/validation loss: representation, optimization, or irreducible-noise limitation;
- low train and high validation loss: variance, leakage in selection, or shift;
- improving validation loss with more data: acquisition may dominate architecture work;
- unstable rankings across splits: evidence is insufficient to distinguish candidates.

Principle. Model complexity is purchased with evidence. If the experiment cannot distinguish two procedures reliably, prefer the simpler operational object unless another constraint decides the choice.

3.5 Hyperparameters are trained parameters

Any value selected using outcomes—regularization, architecture, preprocessing, feature count, augmentation, threshold—is part of the trained procedure. If λ is selected from candidates Λ, then evaluation must include the selection map

$$\widehat{\lambda}(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}) = \arg\min_{\lambda \in \Lambda} \widehat{R}_{\mathcal{D}_{\text{val}}}(f_{\lambda, \mathcal{D}_{\text{train}}}).$$

Reporting the validation error of the winning candidate as if λ were fixed underestimates selection optimism. Nested evaluation or a sealed final test set protects the estimate [24].

3.6 Probabilities, calibration, and decisions

A score becomes a decision only after a threshold and cost model. A probability predictor is calibrated when

$$\mathbb{E}[Y \mid \widehat{p}(X)] = \widehat{p}(X) \qquad \text{almost surely.}$$

Discrimination asks whether positives rank above negatives; calibration asks whether numerical probabilities mean what they say. A model can excel at one and fail the other. Reliability bins are finite-sample estimates of this conditional relationship, not the definition itself [7].

For false-positive cost C_{10} and false-negative cost C_{01}, under simple binary costs predict positive when

$$\widehat{p}(Y = 1 \mid x) > \frac{C_{10}}{C_{10} + C_{01}}. \tag{3.5}$$

This threshold assumes calibrated posterior probabilities, two exhaustive actions, zero cost for correct decisions, and constant false-positive and false-negative costs.

Thresholds should be selected on validation data for a declared operating condition, then frozen before test evaluation.

Failure mode. Using the test set to choose a threshold converts the test set into a validation set, even when the model weights never change.

Chapter 4

Evaluation without self-deception

4.1 Build the split around the claim

The split is a simulation of deployment. A random split asks about new exchangeable rows. A grouped split asks about new entities. A forward split asks about a later period. A site-held-out split asks about a new institution. Choose the split whose failure boundary matches the claim [19].

The common three-way design is

Training: fit model and preprocessing parameters.

Validation: select procedures, hyperparameters, and thresholds.

Test: estimate performance once for the frozen procedure.

When data are limited, nested cross-validation replaces a single validation split. The outer loop estimates the complete selection procedure; the inner loop chooses within each outer-training set. Reusing one cross-validation loop for both choice and reporting is optimistically biased [24].

4.1.1 Nested group evaluation

For each outer fold:

1. hold out entire groups as outer test;
2. run group-aware inner folds only inside outer training;
3. fit every transform separately inside each inner fold;
4. select one complete configuration from inner validation results;
5. refit that configuration on the complete outer-training set;
6. predict the outer-test groups exactly once.

Aggregate the out-of-fold predictions across outer folds before computing the primary metric. Keep fold and group identifiers for auditing.

4.1.2 Worked example: when a visit recognizes its patient

Consider a fixed, fully synthetic cohort of 60 patients with four visits per patient. Each assay contains weak disease signal and a stable patient-specific fingerprint. The fingerprint is independent of disease for a newly generated patient, but it can identify sibling visits from a patient already represented in training. This construction is deliberately idealized: it isolates an evaluation failure rather than modelling biology.

We compare two complete nested procedures. $N0$ assigns visits to folds; D assigns whole patients to folds. Otherwise they are matched: every imputer, scaler, and selector is fit inside its permitted fold; both procedures search the same 15 pipelines; inner selection minimizes patient-level Brier loss; each outer fit predicts held-out visits once; and four visit probabilities are averaged before scoring patient g:

$$\bar{p}_g = \frac{1}{4}\sum_{v=1}^{4} p_{gv}, \qquad \widehat{R}_{\text{Brier}} = \frac{1}{60}\sum_{g=1}^{60}(\bar{p}_g - y_g)^2.$$

The split therefore changes the question, not merely the implementation. Under an idealized independent five-way row allocation, a held-out visit has a training sibling with probability

$$1 - \left(\frac{1}{5}\right)^3 = \frac{124}{125}.$$

This is a mechanism calculation, not the observed overlap of the constrained fold manifest. In the frozen manifest, sibling visits crossed the $N0$ training boundary. The resulting synthetic patient Brier scores were 0.0073 for $N0$ and 0.2414 for D, against the designed $p = 0.5$ baseline of 0.25.

Two full-workflow controls locate the shortcut. Across 100 balanced patient-label permutations of this same cohort, with all 15 candidates refit each time, $N0$ remained below 0.25 in 100/100 runs (median 0.0021), whereas D did so in 5/100 (median 0.2901). Independently reassigning all stable fingerprint vectors raised $N0$ Brier from 0.0073 to 0.1734. These diagnostics support an identity shortcut; they are neither a calibrated permutation test nor proof of sole causation.

Failure mode. A visit-IID resample is a change of estimand, not a more precise patient analysis. The separate $N4$ counterexample treats 240 visits as independent and yields visit-level Brier 0.0458, with pseudo-range 0.0208–0.0750. It is not plotted on the patient-Brier axis and is not an uncertainty interval.

All numbers above describe one frozen, synthetic, idealized cohort. The patient-resampling ranges in the executable companion only reweight fixed OOF predictions; they are not confidence intervals for algorithmic risk. No outer Monte Carlo was run, so expected risk, the expected $N0 - D$ gap, and selection stability across new cohorts remain unknown. Inspired by failure modes found during the capstone audit, this is not a capstone rerun or corrected performance estimate, and it establishes no biological, causal, external, or clinical validity.

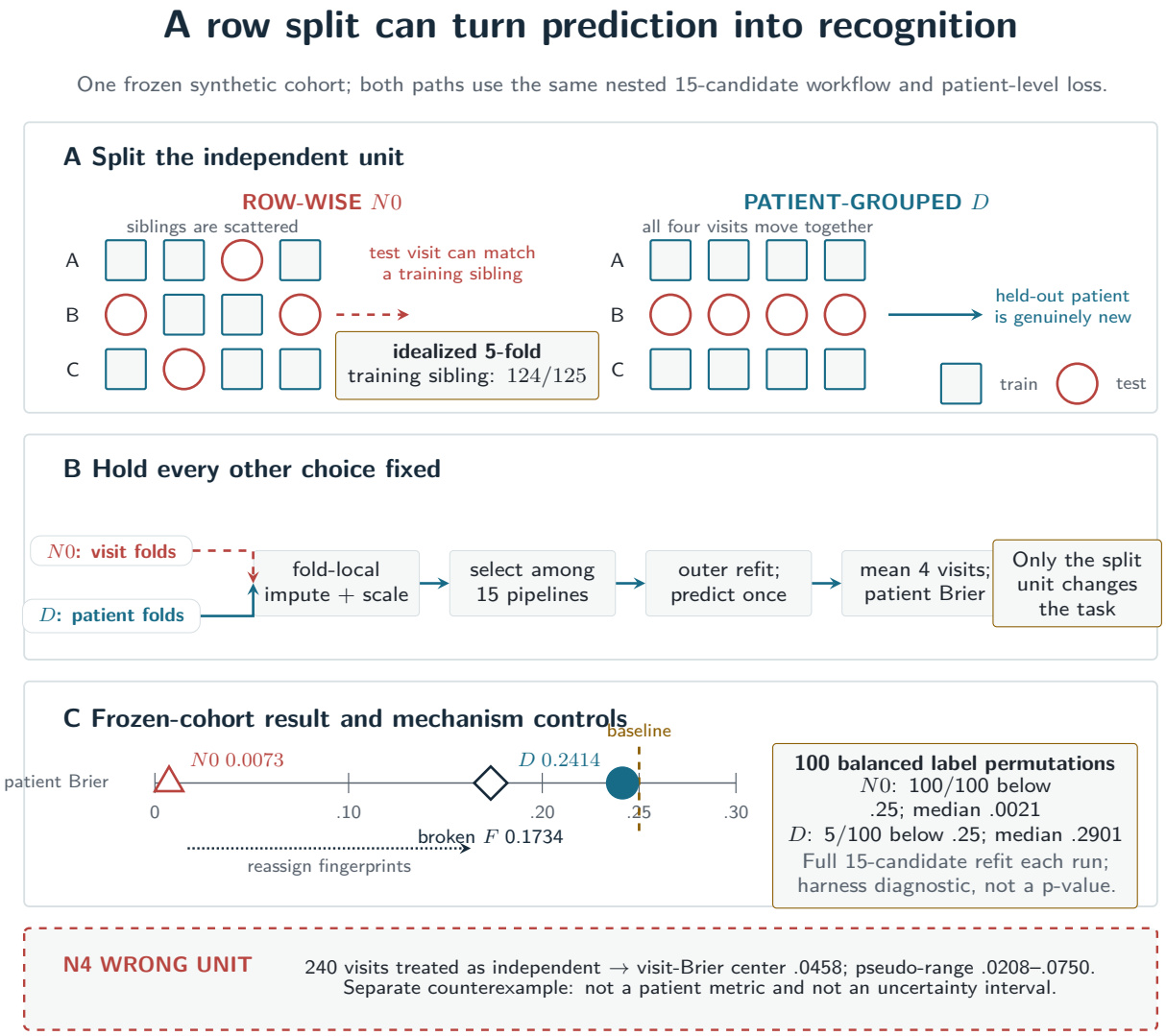

Figure 4.1: **Repeated-measure leakage in a synthetic, idealized worked example.** Row-wise nested cross-validation lets sibling visits cross the training boundary and therefore measures a known-patient task; patient-grouped nested cross-validation measures the intended new-patient task. Both paths use the same 15-candidate family, fold-local transforms, patient-level selection loss, and four-visit aggregation. On this fixed 60×4 synthetic cohort, patient Brier was 0.0073 for $N0$ and 0.2414 for D, versus the designed 0.25 baseline. One hundred full-workflow patient-label permutations and the fingerprint reassignment are harness diagnostics, not a p-value or general effect estimate. The $N4$ visit-level pseudo-range is deliberately quarantined because its unit is not patient Brier. These results are not typical-performance, clinical, or capstone performance claims; the example is inspired by audited failure modes, and all data and numerical results are synthetic and idealized for instruction.

4.2 Confusion metrics name their denominators

For positive class 1:

$$\text{precision} = \frac{TP}{TP+FP}, \qquad \text{recall} = \text{TPR} = \frac{TP}{TP+FN}, \tag{4.1}$$

$$\text{specificity} = \text{TNR} = \frac{TN}{TN+FP}, \qquad \text{FPR} = \frac{FP}{FP+TN}. \tag{4.2}$$

Balanced accuracy is

$$\frac{1}{2}(\text{TPR} + \text{TNR}).$$

Accuracy weights classes by their observed prevalence. This may match deployment cost, or may conceal total failure on a rare class. State the class mapping and show counts alongside rates.

Audit. Unit-test every metric on an asymmetric hand-computed confusion matrix. Symmetric examples allow swapped labels and wrong denominators to pass.

The historical capstone utilities provide exactly this warning: a false-positive rate divided by the positive count and contradictory class semantics produced plausible charts with invalid meanings [4].

4.3 Threshold curves

For score $s(x)$ and threshold t, an ROC curve plots TPR against FPR as t varies. A precision–recall curve plots precision against recall. ROC AUC has the ranking interpretation

$$\text{AUC} = \mathbb{P}(s(X^+) > s(X^-)) + \tfrac{1}{2}\mathbb{P}(s(X^+) = s(X^-)).$$

This probability interpretation uses independent draws from the positive and negative score distributions, with ties receiving the displayed half weight [9].

AUC is threshold-free but not decision-free: it averages ranking performance over operating regions that may be irrelevant. Under severe imbalance, precision–recall quantities expose the burden of false positives more directly [21].

Never construct an "ROC" curve by rescaling a fixed set of class probabilities without actually sweeping a decision threshold and recomputing the confusion matrix.

4.4 Proper scoring rules

Hard decisions discard information. For predicted probability p_i, binary log loss and Brier score are

$$\text{LogLoss} = -\frac{1}{n}\sum_i [y_i \log p_i + (1-y_i)\log(1-p_i)], \tag{4.3}$$

$$\text{Brier} = \frac{1}{n}\sum_i (p_i - y_i)^2. \tag{4.4}$$

Both are proper: in expectation, honest probabilities minimize the score [12]. Report an uncertainty-aware calibration curve or a reliability table with per-bin counts; binning or smoothing choices are part of the estimator, and useful resolution requires adequate validation data [7]. Compare against prevalence and simple calibrated baselines.

4.5 Subgroups and aggregation

Overall performance may hide weak subgroups. Decomposable losses are weighted averages of subgroup losses, whereas nonlinear summaries such as F1 and AUC do not generally decompose that way. Pre-specify meaningful slices: site, device, time, demographic group, missingness pattern, acquisition pathway, and label quality. For each slice report sample size and uncertainty; do not mine dozens of tiny subgroups for dramatic noise.

Macro averaging gives each class equal weight; micro averaging pools decisions; weighted averaging uses class support. None is universally correct. The estimand decides.

4.6 Negative controls and invariants

Before trusting a headline result, require tests that should fail:

- shuffled labels should eliminate real predictive signal;
- an identifier-only model should not perform well;
- training on future-only features should be rejected by schema checks;
- duplicate rows must not cross split boundaries;
- metrics should be invariant to row order;
- probability columns must follow the estimator's explicit class order;
- a tiny synthetic dataset should recover a known optimum.

Principle. A good evaluation harness contains tests designed to embarrass it.

4.7 Test-set governance

Audit. Historical ERP-Classifier—stored artifacts inspected; training not rerun.

Unit.	Subjects were concatenated before an unseeded random pooled-example split; this cannot estimate new-subject performance.
Fit.	A label-derived target template used pooled data before splitting; the logistic branch also scaled training and validation independently.
Choice.	The reported configuration is the unique global maximum of chirplet-MLP validation accuracy across 864 committed rows: 54 JSON files $\times$ 16 channels.
Measure.	Accuracy after class balancing, with unmatched branch splits and no subject-level estimates or uncertainty.
Claim.	Artifact lineage is reproducibly traceable; superiority for new subjects is unsupported. The paper was coauthored, and its dataset and foundational method are third-party. No source data, code, figures, tables, or manuscript text are reproduced.

Shipping rule. The final report should be generated from immutable out-of-sample predictions, not by rerunning an interactive notebook with hidden state.

Chapter 5

Uncertainty and comparison

5.1 Name the randomness

Variation can come from sampled entities, measurement noise, labeling, partition choice, initialization, minibatch order, and nondeterministic hardware. A seed probes only the mechanisms it controls. Five seeds on one fixed split do not represent five new samples from the deployment population.

For experimental result M, a useful conceptual decomposition is

$$M = M(P, \mathcal{D}, S, A, H),$$

where P is population, $\mathcal{D}$ sampled data, S split, A algorithmic randomness, and H analyst choices. An interval must say which of these vary and which remain fixed.

5.2 Fold variation is not a standard error

Cross-validation folds share most training observations, so fold scores are correlated. Their standard deviation divided by $\sqrt{k}$ is not generally a valid standard error for deployment performance. Small samples can yield large uncertainty even when fold scores look stable [25].

The wave-one capstone compiler pools repeated results from nominal folds that reuse the same seeded partition. This is pseudo-replication, not independent evidence [4].

Failure mode. More rows in a results table do not imply more independent information. Identify the sampling unit before computing a standard error.

5.3 Intervals tied to an estimand

For m independent group-level observations of a metric-like quantity, the familiar interval

$$\bar{M} \pm t_{1-\alpha/2,m-1}\frac{s}{\sqrt{m}}$$

requires independence and a plausible sampling model. It is not repaired by forcing a small nonzero standard error when variance is zero; that manufactures precision and can push bounded metrics outside $[0, 1]$.

For a fixed classifier tested on n independent binary outcomes, uncertainty for accuracy is binomial-like. For grouped or repeated observations, resample the independent groups. A nonparametric group bootstrap is:

1. sample group IDs with replacement;
2. include all predictions belonging to each selected group;
3. recompute the metric;
4. use the bootstrap distribution for intervals and diagnostics.

The resampling unit and scheme must mimic the sampling process behind the estimand; behavior depends on the number and size balance of clusters [10].

This conditions on the trained procedure unless training is repeated inside each resample. State which version you computed.

5.4 Paired comparison

Two models evaluated on the same units should be compared through a paired analysis, not overlapping marginal intervals. For a decomposable loss per independent group L_{ag} and L_{bg}, define $D_g = L_{ag} - L_{bg}$ and estimate uncertainty for $\mathbb{E}[D_g]$. Pairing removes shared sample difficulty. For nondecomposable statistics such as AUC, F1, or a calibration summary, resample the same groups for both models and recompute the complete statistic on each paired resample.

Report practical magnitude, not only a null-hypothesis decision. A tiny average improvement may be irrelevant relative to latency, calibration, maintenance, or subgroup regressions.

5.5 Selection uncertainty

If model A wins one split and model B another, the instability is evidence. Record selection frequency across outer folds, rank distributions, and the performance cost of choosing a simpler candidate. Do not select the single highest noisy mean and erase the comparison history.

Principle. Uncertainty belongs to the claim, not as an error bar pasted onto a number after model selection.

5.6 Reproducibility levels

Distinguish:

Computational repeatability: same artifacts and environment reproduce the output.

Robustness: reasonable analytic choices do not reverse the conclusion.

Replication: new data and an independent workflow support the claim.

The National Academies uses *reproducibility* for obtaining consistent computational results from the same inputs and methods, and *replicability* for consistent results across studies addressing the same question [15]. Here "computational repeatability" is the narrow first category, while robustness is an EngSci++ intermediate category.

Deterministic code helps diagnose differences, but identical reruns cannot establish generalization. Conversely, small nondeterminism is not a scientific failure if its distribution is measured and the conclusion is stable.

Chapter 6

The experiment is a data system

6.1 Pure stages and explicit artifacts

Treat the workflow as a directed acyclic graph of typed transformations:

$$\text{raw} \to \text{validated} \to \text{split manifest} \to \text{fitted pipeline} \to \text{predictions} \to \text{metrics} \to \text{report}.$$

Each edge should have a schema and each node should be reproducible from named inputs and configuration. This prevents interactive notebook state from becoming an invisible dependency.

Artifact	Minimum contents
Dataset manifest	source hashes, schema, row/group counts, rights, construction revision
Split manifest	stable IDs, groups, role/fold, policy, seed as metadata
Pipeline spec	ordered transforms, fit scope, versions, hyperparameters
Run manifest	code revision, environment lock, inputs, configuration, start/end status
Predictions	stable IDs, truths, scores, class order, model/run ID
Metric report	definitions, denominators, slices, uncertainty method

6.2 Fit and transform are different operations

An ML pipeline needs a lifecycle contract:

```
split = load_frozen_split_manifest()
pipe = Pipeline([imputer, encoder, scaler, reducer, estimator])
pipe.fit(rows[split.train])

val_rows = rows[split.validation]
val_y = val_rows.target
val_pred = pipe.predict_proba(val_rows)
threshold = choose_threshold(
    val_y, val_pred, objective=decision_contract
)

test_pred = pipe.predict_proba(rows[split.test])
write_predictions(test_pred, threshold, run_manifest)
```

The same fitted object must transform validation, test, and production records. Offline/online skew appears when training and serving implement transformations independently. The decision

contract may also require stable identifiers, groups, sample weights, class order, and error costs; pass every such input explicitly rather than hiding it inside the threshold routine.

6.3 Caching without epistemic corruption

Caches are performance tools, not sources of truth. A cache key must include all inputs that affect its value: data hashes, code revision, transform configuration, fit partition, and relevant library versions. A filename such as `imputed.csv` cannot establish which rows influenced the imputer.

Shipping rule. Cached fitted state must be partition-specific. If an imputation cache was fit globally, no downstream split can undo the leakage.

6.4 Tests for scientific software

Add domain tests beyond ordinary unit tests:

- schema and range checks at ingestion;
- primary-key uniqueness and join-cardinality assertions;
- no group overlap across split roles;
- no feature timestamp after prediction time;
- finite-difference gradients and tiny analytic solutions;
- class-order and probability-simplex checks;
- metric checks against hand calculations and a trusted library;
- pipeline serialization round trips in which every intended trainable component appears in optimizer and state manifests and receives the expected gradient or update on a tiny batch;
- deterministic reconstruction of reports from saved predictions.

Property tests are especially valuable: permuting row order should not change a metric; duplicating one group should affect a group-weighted metric predictably; changing a test feature must not alter fitted training state.

Breck et al.'s production-readiness rubric supports treating data, model, infrastructure, and monitoring tests as first-class ML controls [6]; the exact tests above are EngSci++ synthesis for scientific workflows.

6.5 Configuration is experimental state

Configuration spread across notebook cells, environment variables, and filename conventions cannot be audited. Use a validated schema; reject unknown keys; log defaults; store the resolved configuration with the result. Sculley et al. identify configuration, data dependencies, feedback loops, and undeclared consumers as characteristic sources of ML technical debt [22].

6.6 Notebooks have a role

Notebooks are excellent for inspection and explanation. They are poor as the only implementation of a pipeline. Move reusable logic into importable modules, make notebooks consume immutable artifacts, and clear or regenerate output in a controlled build.

Principle. The report is a view over evidence artifacts. It should not secretly create the evidence it claims to summarize.

6.7 From experiment to production

Pre-deployment evaluation estimates a bounded historical claim. Production adds latency, missing features, feedback loops, human adaptation, and shift. Define:

- input/schema monitoring and feature availability;
- score and decision distributions by slice;
- delayed outcome joins and calibration monitoring;
- abstention and safe fallback behavior;
- model/data lineage for every prediction;
- rollback triggers and ownership.

The NIST AI RMF supplies a voluntary lifecycle framework for assigning ownership, measuring risks, and responding to observed failures [23]; citing it does not assert certification or regulatory compliance.

Monitoring can expose violated observable assumptions quickly, but it cannot by itself identify every form of concept shift or establish safety. The deployment contract should say which observed changes require retraining, recalibration, investigation, or shutdown.

Chapter 7

Case study: a defensible biomarker pipeline

7.1 Provenance and educational reconstruction

This chapter is based on the Oxford Cancer Analytics capstone repositories and their wave-one forensic ingest [4]. The historical project combined biomarker tables, missing-data handling, categorical encoding, dimensionality reduction, KDE augmentation, multiple classifiers, parameter search, repeated seeds, and extensive result visualization.

That breadth makes it an unusually valuable teaching case. It also produced several validity failures: repeated patient IDs were split by row; imputation was fit before the held-out split; nominal folds repeated one seeded partition; all candidates touched the same test set; metric implementations had class and denominator errors; and confidence intervals pooled dependent observations.

No historical biomedical performance number is endorsed here. The pipeline below is an idealized reconstruction [idealized]: it preserves the problem and useful architecture while applying the rules developed in this book.

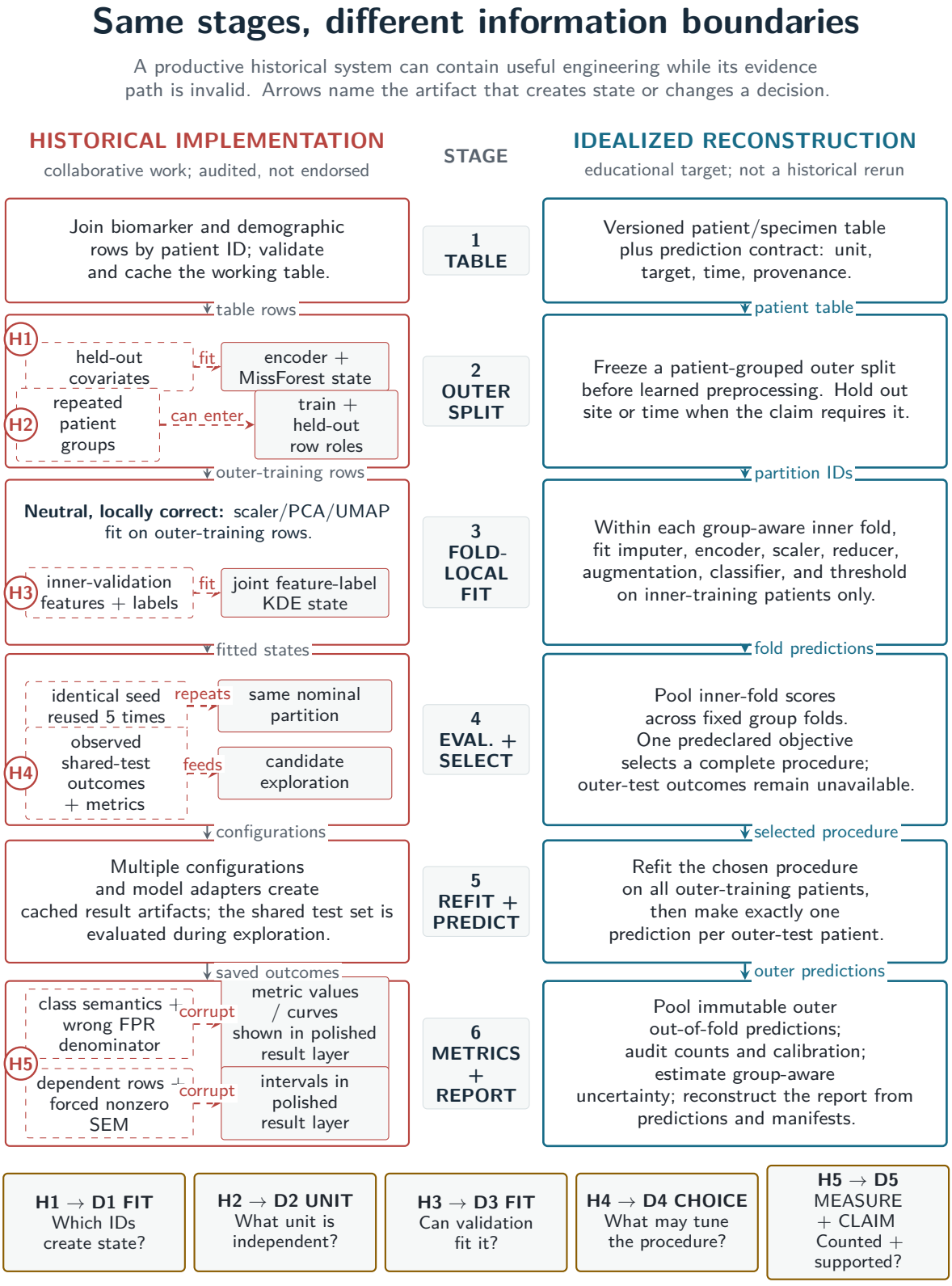

Figure 7.1: Historical information dependencies are contrasted with an idealized defensible reconstruction. The reconstruction is educational: it is not a rerun of the collaborative historical project and does not imply improved historical performance. The observed-test-metric arrow records the architectural problem created when shared test outcomes are available during exploration; it does not claim that a documented final winner was selected from those scores.

7.2 Step 1: write the clinical prediction contract

An example contract might be:

> At the time a blood panel is available, estimate the probability that a new patient belongs to a predeclared cancer-vs-control target population. Evaluate patient-level discrimination, calibration, and sensitivity at a clinically chosen specificity. Do not claim screening utility, causal biomarker relevance, or transport to other hospitals without separate evidence.

This immediately forces unresolved questions: Are repeated rows multiple assays or duplicated records? Which sites and acquisition pathways exist? How are controls selected? What does a missing tumour stage mean? Which biomarkers are available in the intended workflow? These are dataset questions, not details to be hidden by an imputer.

7.3 Step 2: construct the immutable patient table

Build a versioned long-form source table, then a modeling view with:

- stable patient and specimen identifiers;
- acquisition time, site, and assay batch;
- raw value, unit, detection-limit flags, and missingness reason;
- target definition and provenance;
- one explicit rule for multiple specimens per patient;
- an audit log of filters and joins.

The historical table had 1,817 rows but 1,188 unique patient IDs. The correct split unit is therefore patient unless documentation establishes another independence structure.

7.4 Step 3: freeze outer patient splits

Create a patient-grouped, label-stratified outer design. If site or time transport is part of the claim, reserve an external site or future period. Store the realized IDs, not only a seed.

Label stratification requires one declared patient-level label. It may be impossible to preserve both label balance and a strict site/time transport test; the target claim decides which constraint dominates (Chapter 2).

Within each outer-training set, create patient-grouped inner folds. Every subsequent fit—imputer, encoder, scaler, reducer, augmenter, classifier, and threshold—occurs inside the relevant inner-training boundary.

7.5 Step 4: establish boring baselines

Before augmentation or representation learning:

1. prevalence-only probability;
2. age/sex or other declared operational covariates;
3. logistic regression with median-plus-indicator imputation;
4. regularized logistic regression on biomarkers;

5. a tree ensemble using the same leakage-safe inputs.

Use identical outer predictions and metrics for every procedure. If complex models do not reliably beat the linear baseline, that is a useful conclusion.

7.6 Step 5: make preprocessing a fitted object

For outer fold o and inner fold j, define

$$\widehat{T}_{oj} = \mathrm{fit}(T, X_{oj}^{\mathrm{train}}), \qquad \widetilde{X}_{oj}^{\mathrm{train}} = \mathrm{transform}(\widehat{T}_{oj}, X_{oj}^{\mathrm{train}}),$$
$$\widetilde{X}_{oj}^{\mathrm{valid}} = \mathrm{transform}(\widehat{T}_{oj}, X_{oj}^{\mathrm{valid}}).$$

Only the fitted state $\widehat{T}_{oj}$, learned from inner-training rows, transforms the corresponding validation fold. After selection, refit the chosen pipeline on all outer-training patients and transform outer-test patients once. Cache keys include outer fold, inner fold, input hash, and configuration.

Missing values outside a model's supported domain should not be silently clipped into plausibility. Range and unit checks precede imputation.

7.7 Step 6: augmentation must respect support

The historical mixed-data KDE concatenated continuous features and encoded labels, sampled in Euclidean space, then truncated and clipped categorical coordinates. Stored output included negative biomarker values and invalid label combinations. Preserve that result as a negative control.

A defensible augmentation study must specify what distribution it approximates. Possible approaches include class-conditional models with explicit support, transformations appropriate to positive concentrations, and categorical models rather than Gaussian smoothing of integer codes. Fit augmentation only on the inner-training partition and compare against no augmentation.

Failure mode. Synthetic data do not create independent evidence. They may change an estimator's inductive bias; they do not increase the number of observed patients.

7.8 Step 7: choose once, test once

For each outer fold:

1. score every configuration on inner validation predictions;
2. choose using a predeclared objective and tie-break rule;
3. refit the chosen procedure on all outer-training patients;
4. write one prediction record per outer-test patient;
5. never use the outer result to revise that fold's candidate set.

The final estimate uses pooled outer predictions. A genuinely sealed external cohort, if available, is evaluated only after the procedure and report template are frozen.

Pooled outer predictions estimate the fold-trained selection procedure, not a final model trained on all available data; deployment refitting and its uncertainty are separate questions.

7.9 Step 8: report what the evidence supports

For a clinical prediction study, the following case-specific minimum is aligned with TRIPOD+AI reporting guidance [8]:

- cohort flow and patient counts, not only row counts;
- missingness and prevalence by split/site/time;
- immutable configuration and split manifests;
- patient-level confusion counts and declared primary metrics;
- calibration and decision-relevant operating points;
- paired differences from baselines with group-aware uncertainty;
- outer-fold variability and selection stability;
- subgroup results with denominators;
- all known deviations, failed checks, and negative controls.

The conclusion should be narrow: this procedure achieved an estimated level of performance for these held-out patients under this data construction. Discrimination and calibration alone do not establish clinical utility; prospective validity, decision impact, and transport require additional studies.

Principle. Idealization is legitimate pedagogy only when the boundary is visible. We improve the pipeline; we do not retroactively improve its historical results.

7.10 What the original project teaches

The capstone's enduring contribution is not a particular score. It shows why serious software can still generate invalid evidence. Its abstractions, model adapters, search tooling, cached artifacts, and visualization made the pipeline productive; absent statistical information boundaries, the same productivity scaled the wrong experiment.

The fix is not to abandon engineering. It is to make the scientific contract an architectural invariant.

Chapter 8

Field guide

8.1 Design review

Before implementation, require written answers:

1. What decision and target population define the claim?
2. What is one independent unit? What correlations remain?
3. Which features exist at prediction time?
4. How are labels generated, delayed, censored, or corrupted?
5. Which split best simulates deployment?
6. What is fit inside each information boundary?
7. What baseline could falsify the need for complexity?
8. Which metric and operating point match the decision cost?
9. What randomness does the uncertainty statement cover?
10. What result stops the project?

8.2 Code review

- Stable IDs survive from raw inputs to predictions.
- Group overlap and post-prediction timestamps are asserted impossible.
- Preprocessing exposes explicit `fit` and `transform` state.
- Fitted state is never shared across folds.
- Model class order is stored with probability columns.
- Handwritten objectives and gradients have numerical checks.
- Metrics have asymmetric hand-calculated unit tests.
- Configurations are validated, resolved, and stored.
- Caches are content-addressed by inputs and fit partition.
- Reports consume frozen predictions rather than retraining models.

8.3 Result review

- Counts and denominators accompany every rate.
- Train, validation, and test roles are unambiguous.
- Hyperparameters and thresholds never use the final test outcome.
- Comparisons are paired on the same independent units.
- Intervals name their resampling unit and conditioning assumptions.
- Subgroup analyses show sample sizes and were not selectively reported.
- Calibration and operating points accompany ranking metrics.
- Negative controls behaved as expected.
- Stored artifacts reconstruct every table and figure.
- The conclusion does not climb past its evidence on the claim ladder.

8.4 The compact workflow

1. Write the prediction contract and estimand.
2. Audit the data-generating process, keys, time, and labels.
3. Freeze group-aware outer splits and manifests.
4. Build the simplest end-to-end baseline pipeline.
5. Put every fitted transformation inside inner evaluation.
6. Unit-test objectives, gradients, class semantics, and metrics.
7. Compare complete procedures using paired outer predictions.
8. Quantify uncertainty at the unit of independence.
9. Generate the report from immutable predictions and manifests.
10. State the narrowest claim that the evidence actually supports.

Shipping rule. A trustworthy ML system makes invalid experimental states difficult to represent: no ungrouped split without an explicit override, no transform without fit provenance, no probability without class order, no metric without a denominator, and no report without a run manifest.

8.5 Final perspective

Good scientific ML feels slower at the first notebook and dramatically faster at the tenth decision. Explicit contracts prevent dead-end experiments. Frozen artifacts make disagreements local. Simple baselines reveal whether complexity is earning its keep. Valid evaluation lets a negative result become durable knowledge instead of an inconvenient chart.

The goal is not ritual purity. It is leverage: building systems whose empirical outputs can safely carry real decisions.

Appendix A

Mathematical quick reference

A.1 Expectation and covariance

For random vector $X \in \mathbb{R}^d$,

$$\mu = \mathbb{E}[X], \qquad \Sigma = \mathrm{Cov}(X) = \mathbb{E}[(X-\mu)(X-\mu)^T].$$

For constant matrix A and vector b,

$$\mathbb{E}[AX+b] = A\mu + b, \qquad \mathrm{Cov}(AX+b) = A\Sigma A^T.$$

Independence implies zero covariance when moments exist; zero covariance does not generally imply independence. Jointly Gaussian variables are an important exception.

A.2 Bayes and log odds

$$p(\theta \mid x) = \frac{p(x \mid \theta)p(\theta)}{p(x)}, \qquad \log \frac{p(Y=1 \mid x)}{p(Y=0 \mid x)} = \log \frac{p(x \mid Y=1)}{p(x \mid Y=0)} + \log \frac{p(Y=1)}{p(Y=0)}.$$

Working in the log domain turns products into sums and avoids numerical underflow. Use log-sum-exp for stable normalization.

A.3 Gradients worth remembering

For compatible dimensions,

$$\nabla_x (a^T x) = a, \tag{A.1}$$

$$\nabla_x \tfrac{1}{2}\|Ax-b\|_2^2 = A^T(Ax-b), \tag{A.2}$$

$$\nabla_x \tfrac{1}{2} x^T Q x = \tfrac{1}{2}(Q+Q^T)x, \tag{A.3}$$

$$\nabla_x \tfrac{\lambda}{2}\|x\|_2^2 = \lambda x. \tag{A.4}$$

Check shapes before algebra: the gradient of a scalar with respect to $x \in \mathbb{R}^d$ must lie in $\mathbb{R}^d$.

A.4 Convexity

A set C is convex when $\lambda x + (1-\lambda)y \in C$ for all $x, y \in C$ and $\lambda \in [0,1]$. A function f is convex when

$$f(\lambda x + (1-\lambda)y) \le \lambda f(x) + (1-\lambda) f(y).$$

For differentiable convex f, every tangent plane is a global lower bound:

$$f(y) \ge f(x) + \nabla f(x)^T (y - x).$$

For twice differentiable f, a positive-semidefinite Hessian throughout a convex domain is equivalent to convexity. Local minima of convex problems are global; strict convexity gives at most one minimizer [5, 1].

A.5 Information quantities

For discrete X, entropy and conditional entropy are

$$H(X) = -\sum_x p(x) \log p(x), \qquad H(Y \mid X) = -\sum_{x,y} p(x,y) \log p(y \mid x).$$

Mutual information is

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} = H(Y) - H(Y \mid X).$$

It is nonnegative and equals zero exactly when X and Y are independent under the stated distribution. Estimated entropy and mutual information can be severely biased in sparse finite samples; estimator choice, support assumptions, and smoothing are part of the claim [16, 3].

Bibliography

[1] Aman Bhargava. ECE367: Matrix algebra and optimization, 2021. Engineering Science, Abridged; authored LaTeX notes.

[2] Aman Bhargava. ECE368: Probabilistic reasoning, 2021. Engineering Science, Abridged; authored LaTeX notes.

[3] Aman Bhargava. EE/Ma/CS 126a: Information theory, 2022. Engineering Science, Abridged; authored LaTeX notes.

[4] Aman Bhargava and EngSci++ ingest reviewers. Undergraduate repository ingest: Wave 1, 2026. Internal clean-room evidence wiki with repository, commit, code-line, notebook-cell, and PDF-page provenance.

[5] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[6] Eric Breck, Shanqing Cai, Eric Nielsen, Michael Salib, and D. Sculley. The ML test score: A rubric for ML production readiness and technical debt reduction. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 1123–1132. IEEE, 2017.

[7] Ben Van Calster, David J. McLernon, Maarten van Smeden, Laure Wynants, and Ewout W. Steyerberg. Calibration: The achilles heel of predictive analytics. *BMC Medicine*, 17(1):230, 2019. On behalf of Topic Group 'Evaluating diagnostic tests and prediction models' of the STRATOS initiative.

[8] Gary S. Collins, Karel G. M. Moons, Paula Dhiman, Richard D. Riley, Andrew L. Beam, Ben Van Calster, Marzyeh Ghassemi, Xiaoxuan Liu, Johannes B. Reitsma, Maarten van Smeden, Anne-Laure Boulesteix, Jennifer Catherine Camaradou, Leo Anthony Celi, Spiros Denaxas, Alastair K. Denniston, Ben Glocker, Robert M. Golub, Hugh Harvey, Georg Heinze, Michael M. Hoffman, André Pascal Kengne, Emily Lam, Naomi Lee, Elizabeth W. Loder, Lena Maier-Hein, Bilal A. Mateen, Melissa D. McCradden, Lauren Oakden-Rayner, Johan Ordish, Richard Parnell, Sherri Rose, Karandeep Singh, Laure Wynants, and Patricia Logullo. TRIPOD+AI statement: Updated guidance for reporting clinical prediction models that use regression or machine learning methods. *BMJ*, 385:e078378, 2024.

[9] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[10] C. A. Field and A. H. Welsh. Bootstrapping clustered data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(3):369–390, 2007.

[11] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé, III, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021.

[12] Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

[13] Sayash Kapoor and Arvind Narayanan. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9):100804, 2023.

[14] Kevin P. Murphy. *Probabilistic Machine Learning: An Introduction*. MIT Press, 2022.

[15] National Academies of Sciences, Engineering, and Medicine. *Reproducibility and Replicability in Science*. National Academies Press, Washington, DC, 2019.

[16] Liam Paninski. Estimation of entropy and mutual information. *Neural Computation*, 15(6):1191–1253, 2003.

[17] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2 edition, 2009.

[18] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence, editors. *Dataset Shift in Machine Learning*. Neural Information Processing Series. MIT Press, 2008.

[19] David R. Roberts, Volker Bahn, Simone Ciuti, Mark S. Boyce, Jane Elith, Gurutzeta Guillera-Arroita, Severin Hauenstein, José J. Lahoz-Monfort, Boris Schröder, Wilfried Thuiller, David I. Warton, Brendan A. Wintle, Florian Hartig, and Carsten F. Dormann. Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, 40(8):913–929, 2017.

[20] Donald B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

[21] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3):e0118432, 2015.

[22] D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems 28*, 2015.

[23] Elham Tabassi. Artificial intelligence risk management framework (AI RMF 1.0). Technical Report NIST AI 100-1, National Institute of Standards and Technology, Gaithersburg, MD, 2023.

[24] Sudhir Varma and Richard Simon. Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, 7:91, 2006.

[25] Gaël Varoquaux. Cross-validation failure: Small sample sizes lead to large error bars. *NeuroImage*, 180:68–77, 2018.