

Evaluating the Quality of Data Science: An Overview

BETTER DATA. STRONGER DECISIONS. GREATER IMPACT.



Quality evaluation is a first-class discipline, not a pre-release checklist



The layered stack: data, method, model, code, impact, governance



One upstream schema change can silently degrade a model, with no error



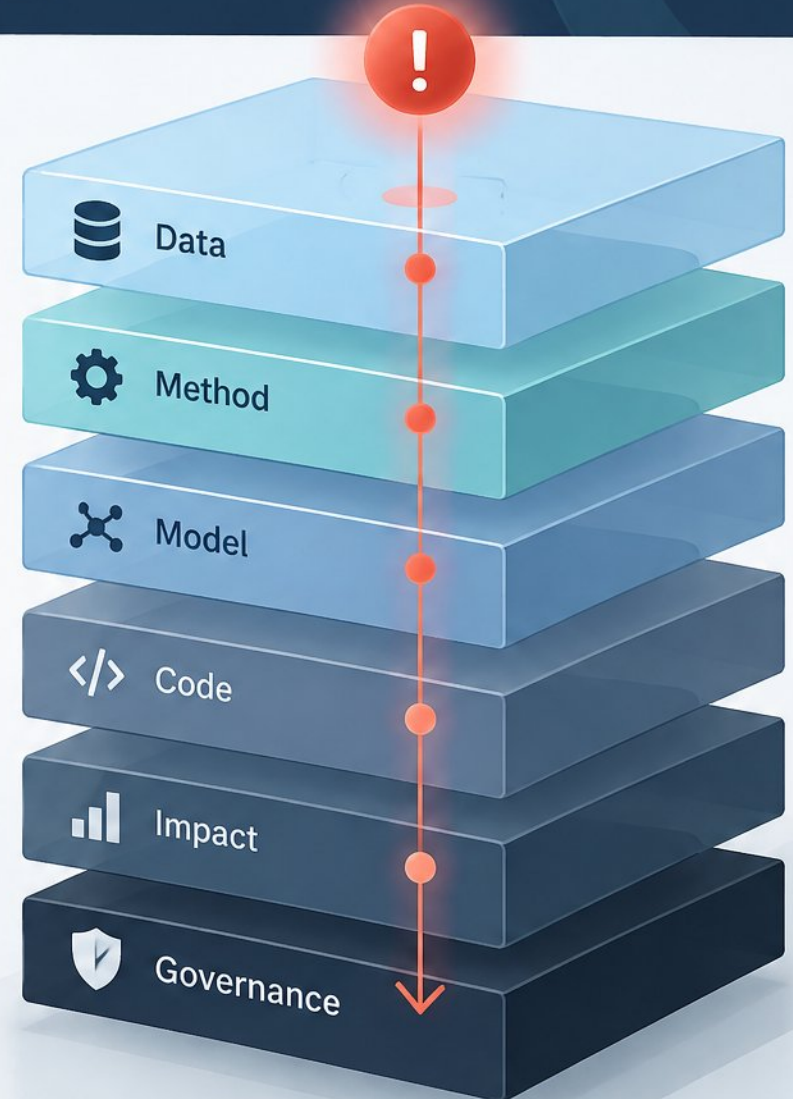
Poor quality costs rework, trust, and regulatory exposure — roughly \$12.9M per year



Every role owns a distinct layer of evidence



Roadmap: the decisions you should be able to make by the end



A Shared Vocabulary for Quality and Failure Archetypes



Validation checks requirements; verification checks correctness against a trusted record



Reproducibility: same code, same data. Replicability: new data. Robustness: different methods



Fitness for purpose: the method must answer the actual decision question



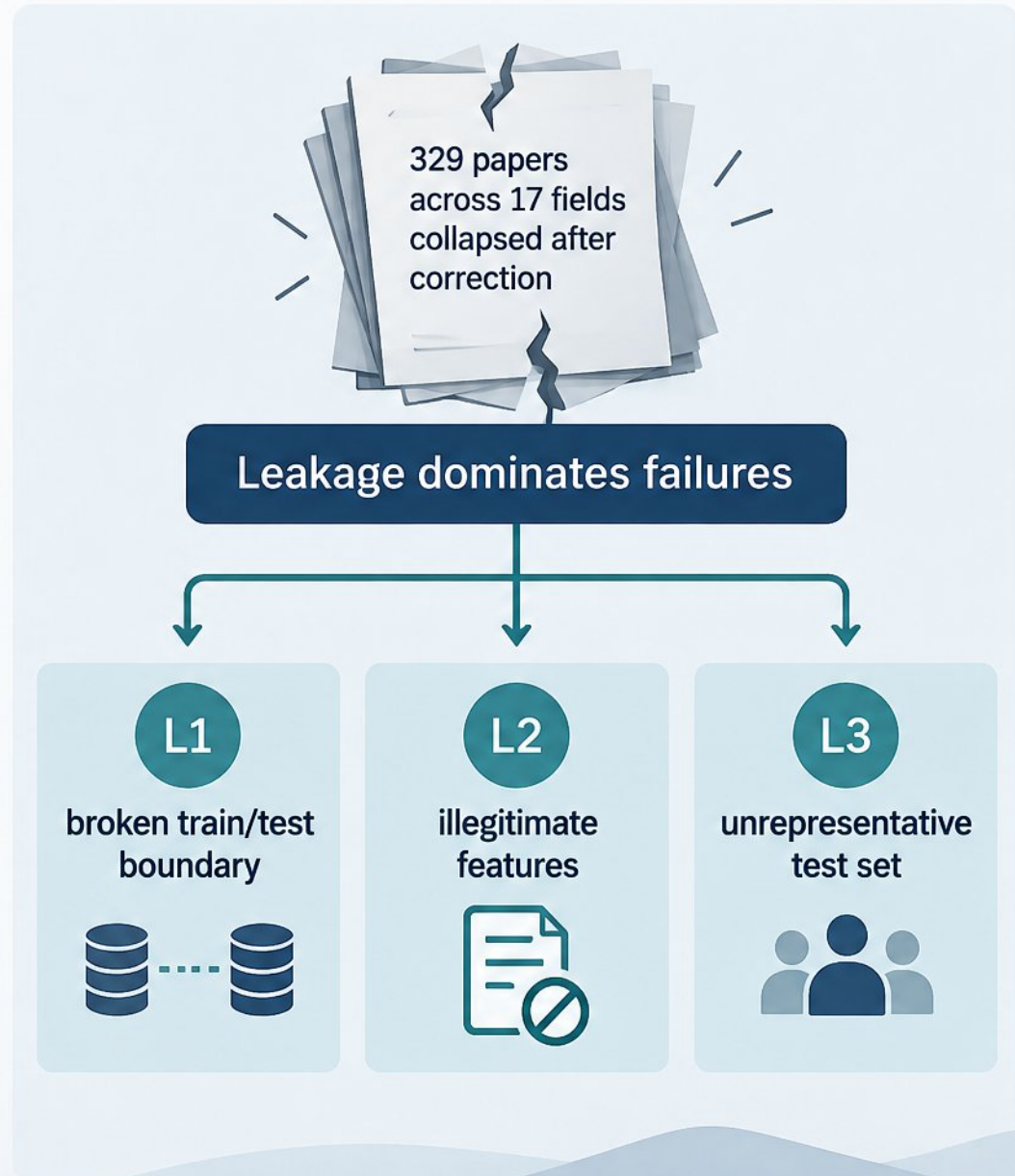
Leakage dominates failures: 329 papers across 17 fields collapsed after correction



Three categories: broken train/test boundary (L1), illegitimate features (L2), unrepresentative test set (L3)

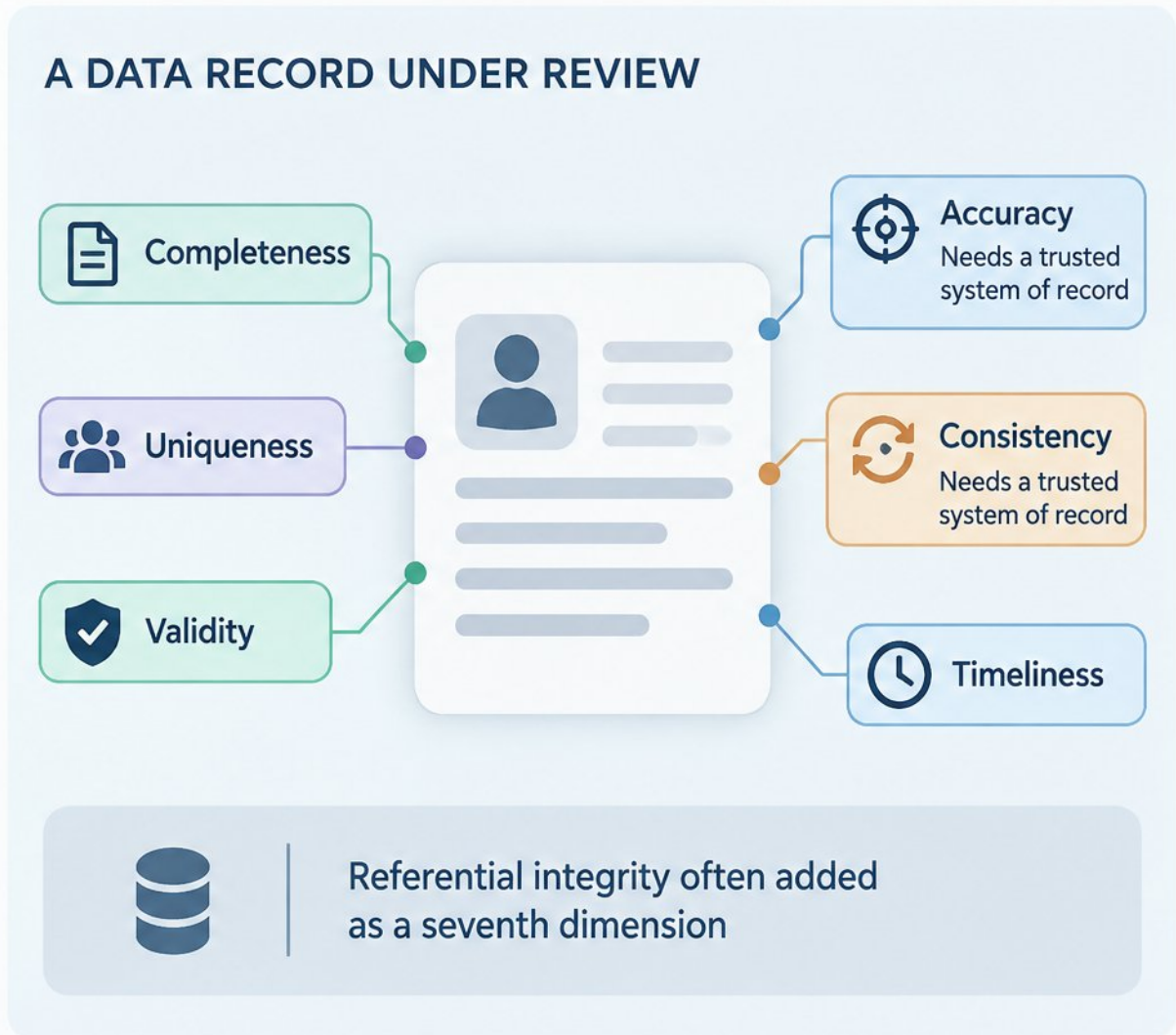


Checklists and model cards miss leakage; provenance records and adversarial review catch it



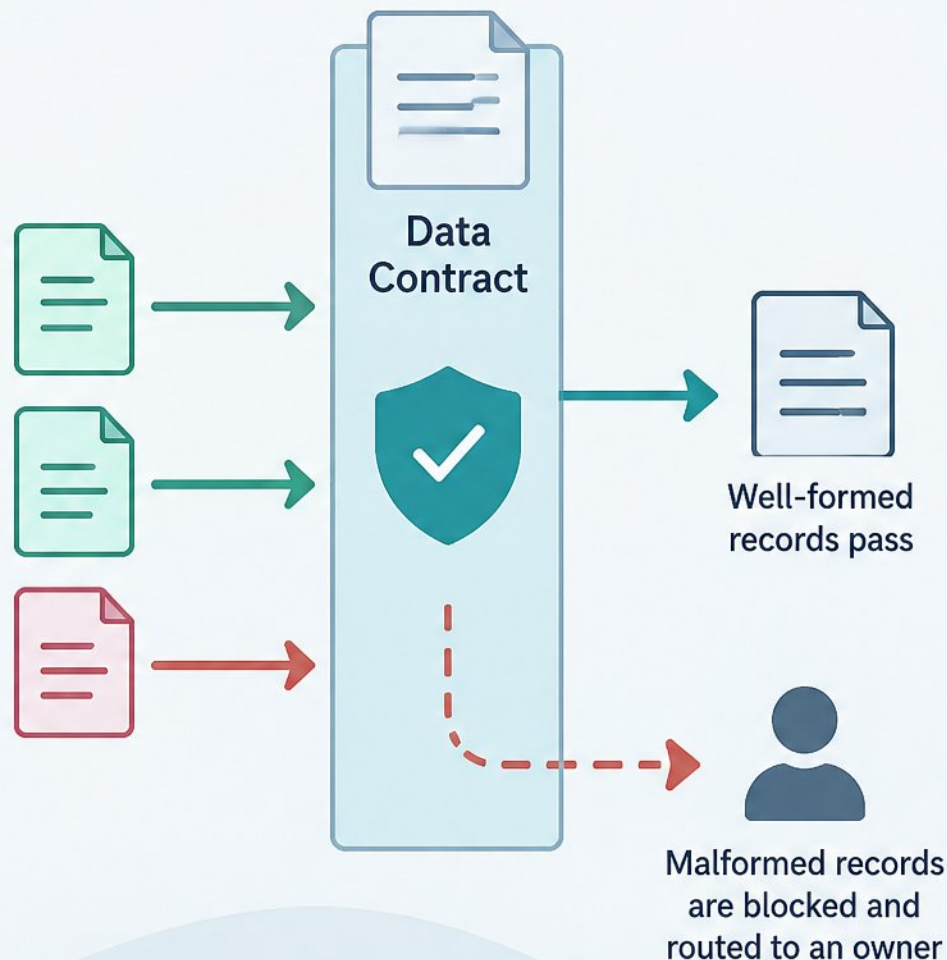
Data Quality Dimensions and Profiling

- Six measurable dimensions: completeness, uniqueness, **validity**, accuracy, consistency, timeliness
- Referential integrity often added as a seventh dimension
- Accuracy and consistency resist mechanical checks: they need a trusted system of record
- Complete, unique, and valid records can still be wrong
- Profile with column statistics, distribution checks, cross-field constraints, anomaly detection
- Separate source-system errors from pipeline-introduced errors
- Account for sampling and measurement error before declaring a defect real



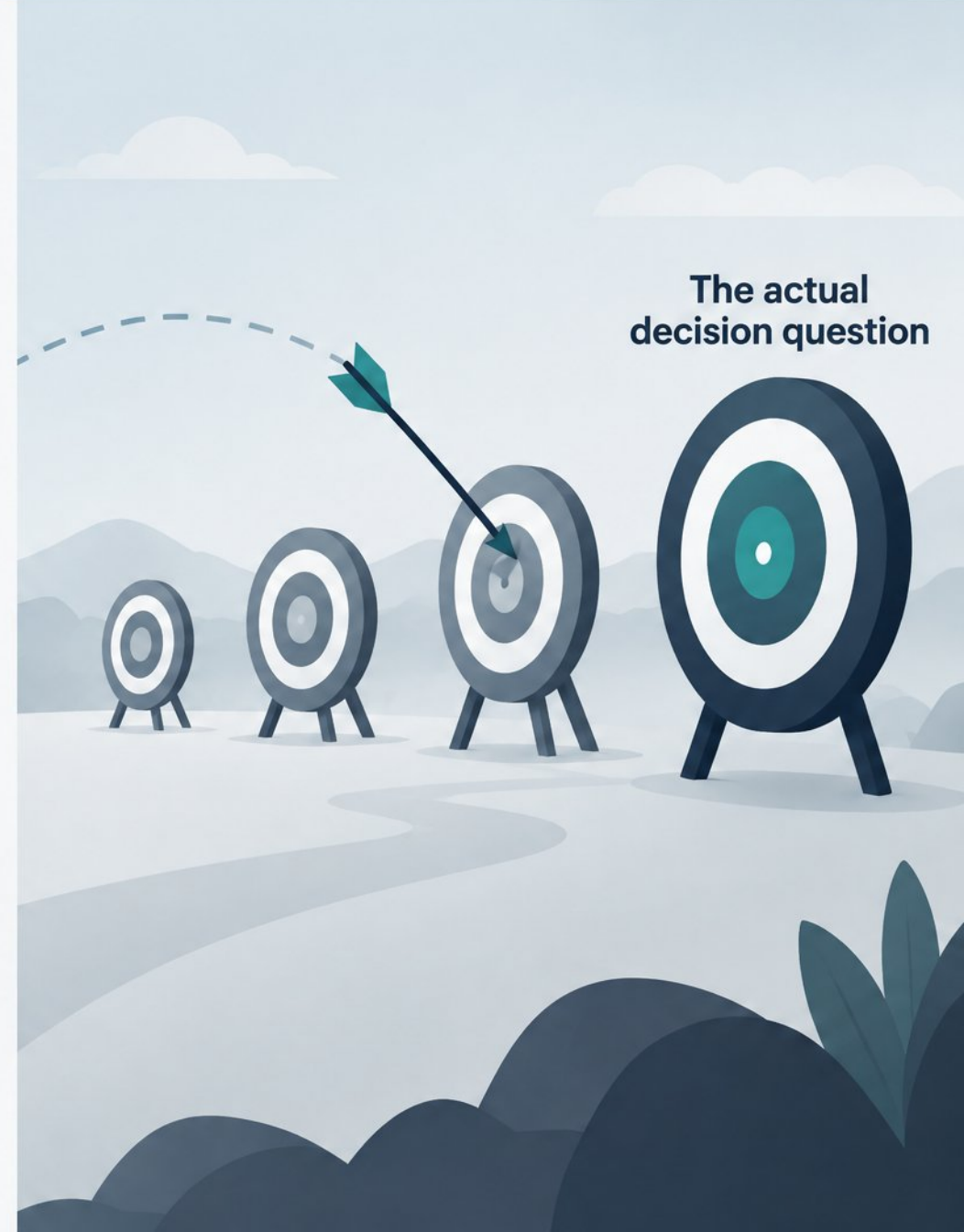
Data Contracts and Expectation Testing

- A data contract: versioned, machine-readable producer-consumer agreement
- Covers schema, semantics, quality rules, SLAs, ownership, change policy
- Prevention at the boundary; observability catches what slips through
- Enforce with Pydantic, Pandera, dbt, Great Expectations, schema registries
- Standardize on Open Data Contract Standard (ODCS) v3.1.0
- Operational test: does a violation trigger something automatically?
- Start with the 3-5 datasets that cause the most incidents

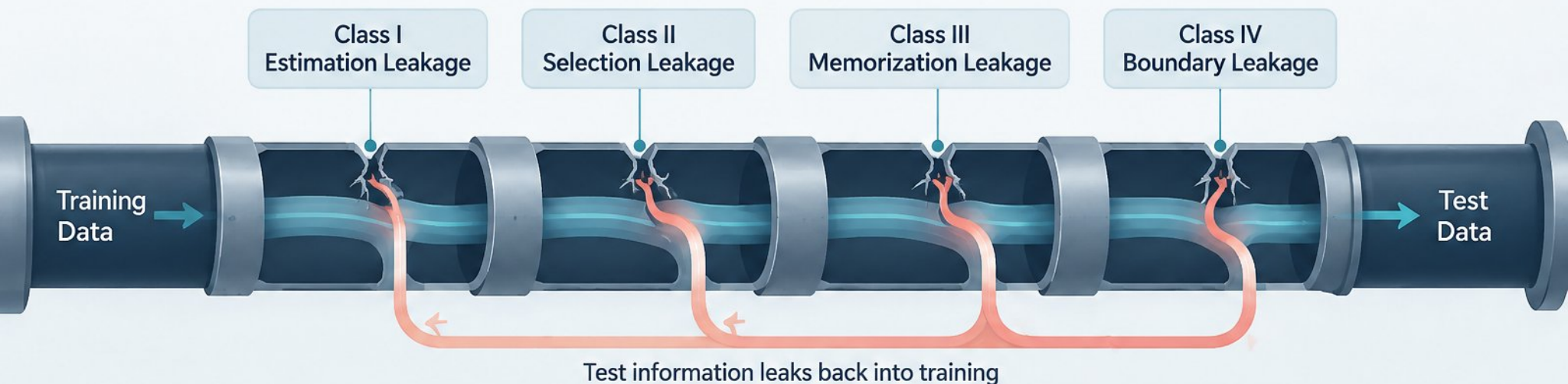


Methodological Soundness and Statistical Validity

- **Fitness for purpose:** answer the real decision question, not a convenient proxy.
- **Bias sources:** sampling, survivorship, confounding, feature selection, and target construction.
- **Check assumptions** and run **sensitivity** analysis before trusting any result.
- **Guard against** multiple comparisons, p-hacking, and overfitting to the analysis set.
- **Claim causality only** when the design supports it; otherwise stay correlational.
- **Reviewer heuristic:** trace every headline claim back to its evidence.



Leakage Deep Dive: The Silent Killer



- Class I estimation leakage — negligible: all nine conditions show ΔAUC below 0.005
- Class II selection leakage is largest at real sizes: peeking +0.040, seed cherry-picking +0.045 AUC
- Class III memorization scales with capacity: dz 0.37 (NB) to 1.11 (decision tree) at 10% duplication
- Class IV boundary leakage hides under random CV: averages +0.023 AUC on true temporal data
- Naive z-based CV intervals cover only ~55% at nominal 95%; conservative-z reaches ~87%
- Guards: split before transform, grouped and temporal splits, importance audits, one untouched holdout



**BETTER
PIPELINES
BUILD
TRUSTWORTHY
MODELS**

Model Evaluation: Metrics, Validation, and Robustness

Stronger models come from better questions, fairer tests, and deeper checks.



Pick metrics by business objective and error cost, not convenience



Splits are structural: temporal for drift, grouped for repeated entities



Discrimination $\neq$ reliability: calibrate scores and justify thresholds



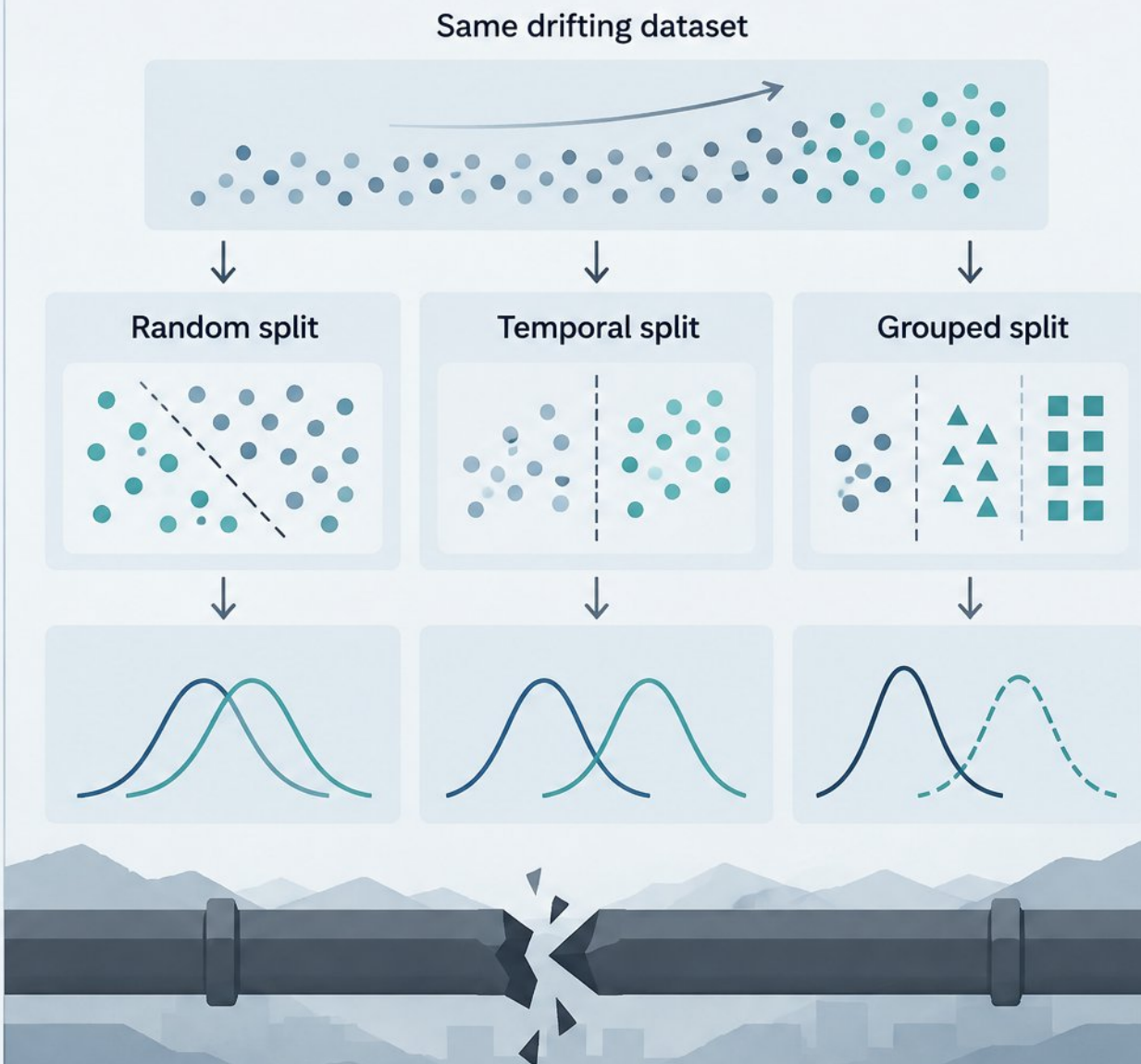
Stress-test under shift, selective labels, adversarial and edge inputs



Baselines and ablations prove added value over rules or last model

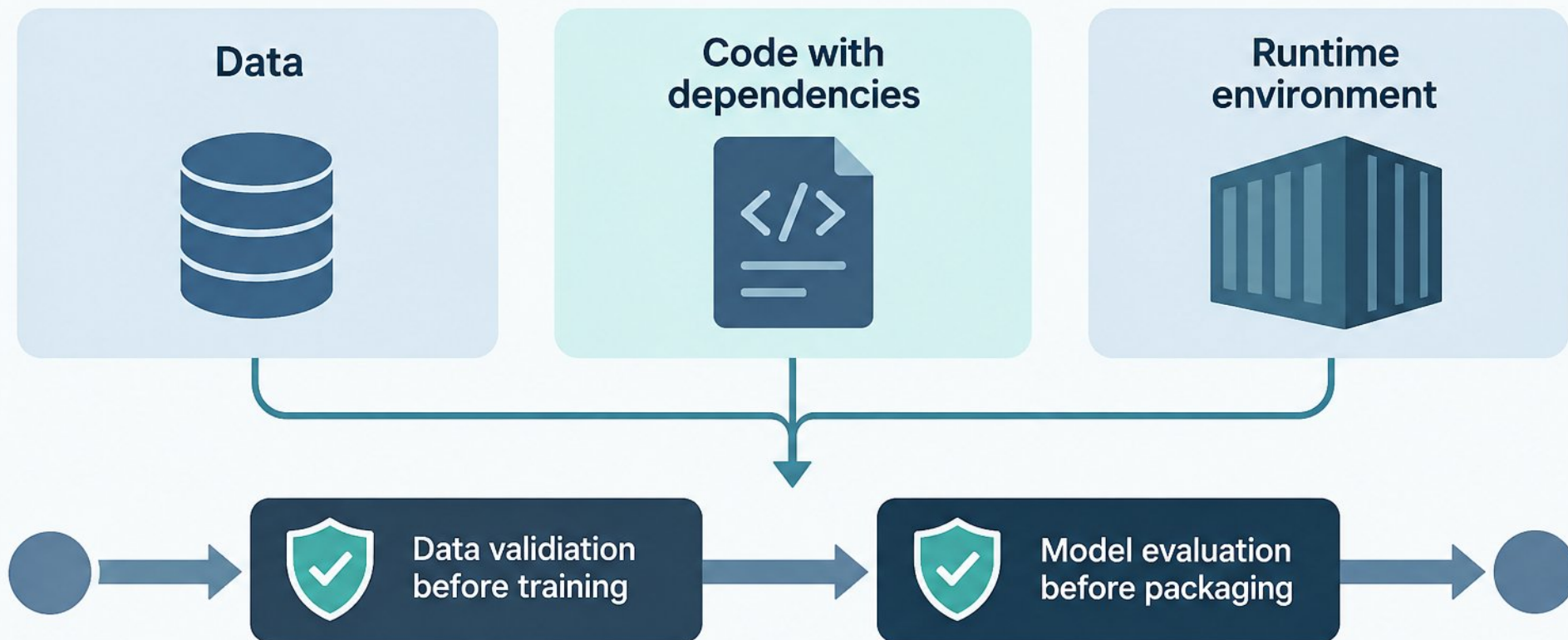


Disaggregated results need causal assumptions to be interpretable



Code, Pipelines, and Reproducibility

- Three ingredients: data, code with dependencies, runtime environment — none sufficient alone
- Version code, data, model, config together; tag datasets by content hash
- Pin container images by digest; pin seeds in Python, NumPy, framework
- If two identical-data runs differ, your eval gate is measuring noise
- CI gates: data validation before training, model evaluation before packaging



Review Process and Technical Peer Critique



Review checklist spans data, method, model, code, claims, and governance



Severity classification separates blocking issues from suggestions



Ask clarifying questions before judgment; structure the conversation



Red-team high-stakes work: feature blackout and prompt injection tests



Analyst-inspector test: can a reviewer regenerate the numbers independently?



Failure messages name the broken field, owner, and approval path



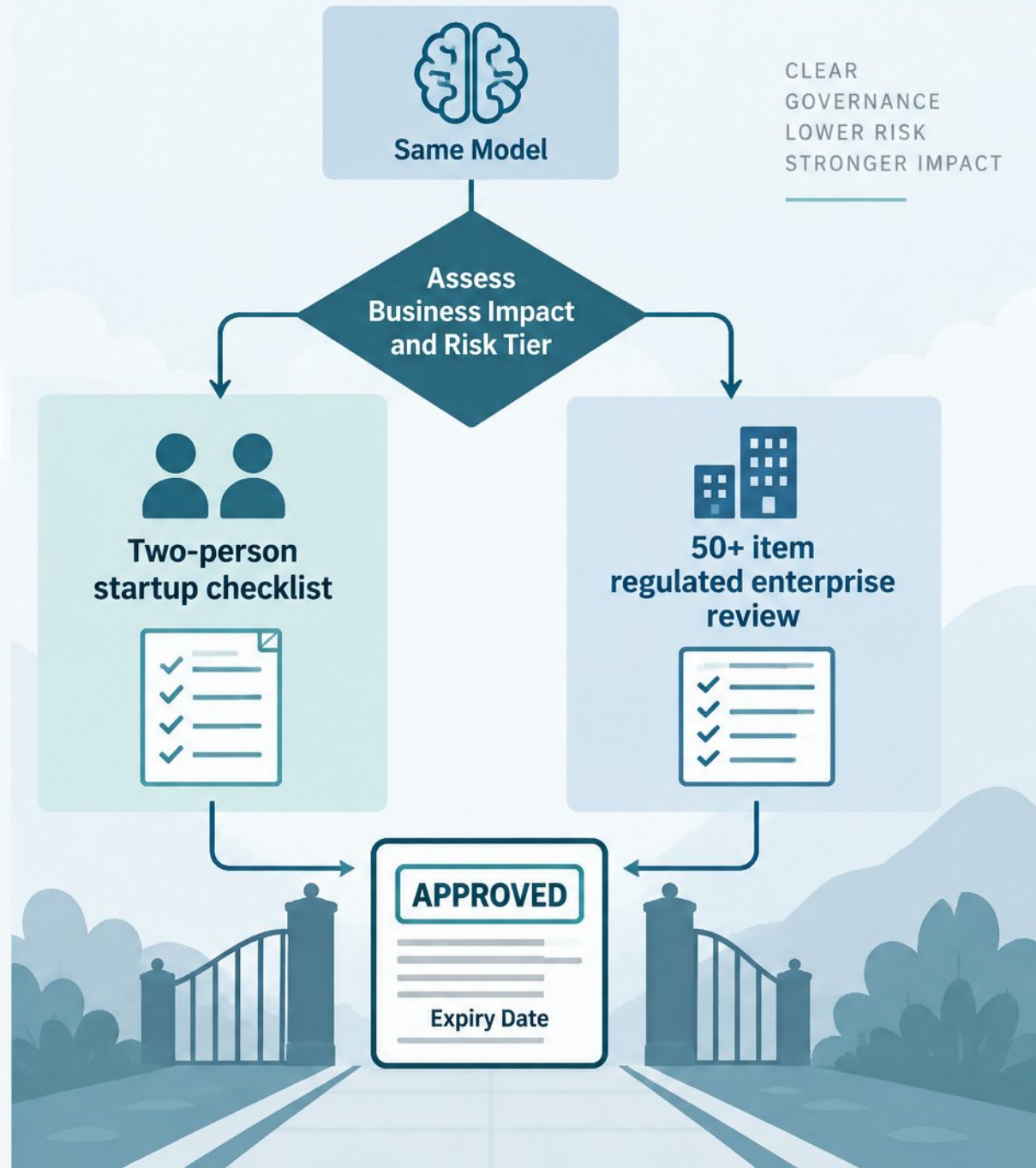
Keep review records for audit and knowledge reuse



Business Impact, Risk Tiering, and Governance

RESPONSIBLE AI
FOR A SAFER
TOMORROW

- 1 Validation is a bounded permission: stated use, stated period, named approver
- 2 Approval memos need expiry dates, compensating controls, and rollback plans
- 3 EU AI Act Article 72 applies since 2 August 2026: post-market monitoring is a rolling conformity audit
- 4 Monitor Art.9–15: risk (9), data (10), docs (11), logs (12), transparency (13), oversight (14), robustness (15)
- 5 Risk-tiered intensity: 10–15 item startup checklist vs. 50+ item regulated enterprise review
- 6 State residual risk honestly; EU non-compliance fines reach €35M or 7% of global turnover



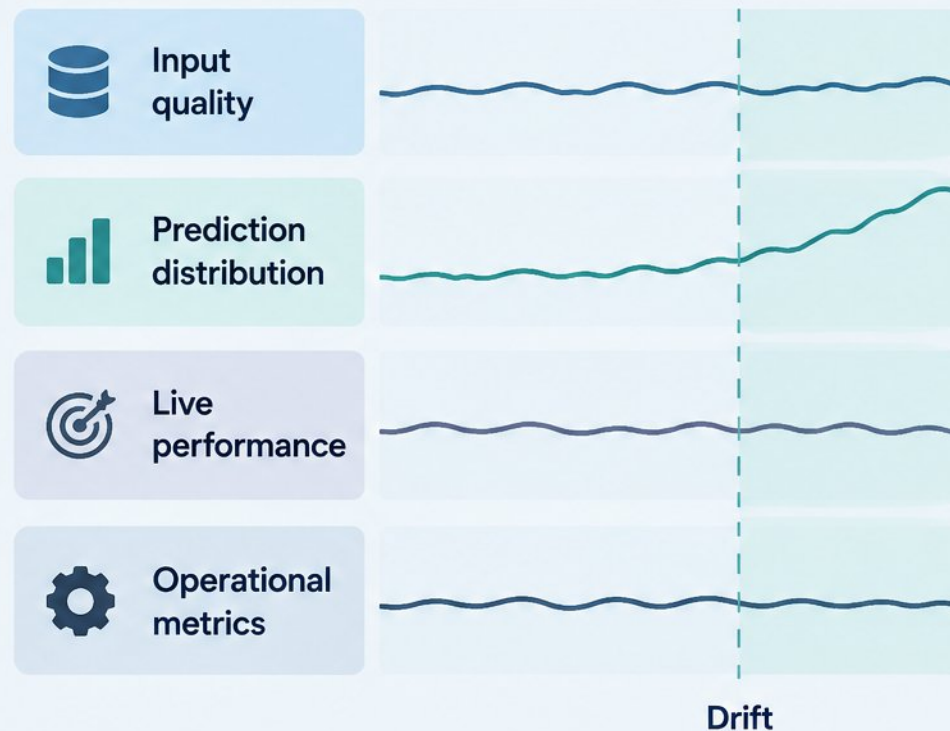
Monitoring, Drift Detection, and Retraining Triggers

STABLE SYSTEMS • RELIABLE INSIGHTS • CONTINUOUS IMPROVEMENT



- 1 Models degrade silently: normal latency, zero errors, wrong predictions
- 2 Monitor four layers: input quality, prediction distribution, live performance, operational metrics
- 3 Drift triggers lead; performance triggers lag; time-based retraining is a failsafe
- 4 PSI: below 0.1 stable, 0.1–0.25 watch, at or above 0.25 alert
- 5 A drift alert starts an investigation, not an automatic retrain
- 6 Every promotion needs champion-challenger gating, canary rollout, tested rollback

Monitoring Across Four Layers



Safe Model Promotion



Monitor.
Detect drift.
Retrain with discipline.
Deliver lasting value.

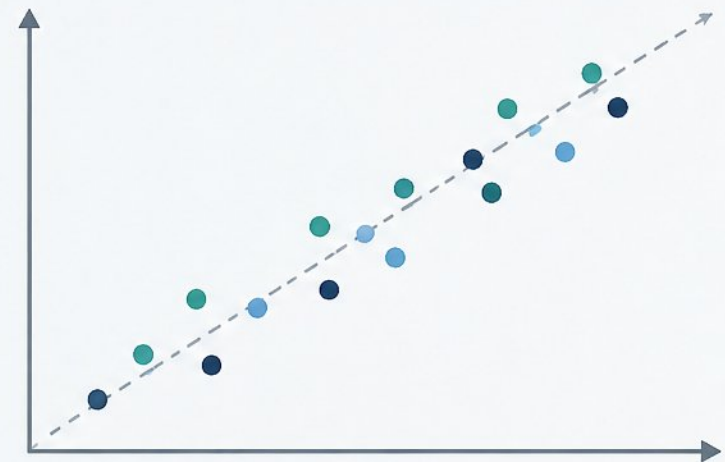
Teaching and Assessing Quality Evaluation Skills

- Align learning objectives with the evaluation tasks that measure them
- Rubrics separate implementation, solution correctness, and interpretation points
- Calibrated peer review: score low/medium/high samples, compare, discuss gaps
- Practice: case critiques, seeded defects, code review, I like / I wish / What if
- Feedback separates technical defects from reasoning gaps; specific, meta, respectful
- Assess evaluators via inter-rater agreement and reviewer calibration over time

Evaluation Rubric

	Implementation	Solution Correctness	Interpretation

Calibrating Peer Review



Tiered Evaluation Workflow: Lightweight Review to Full Audit



One integrated checklist: data, method, model, code, impact, monitoring, governance



Match intensity to stakes: PR review, release review, independent structural audit



Define roles: who decides, who produces evidence, who can block



Artifacts: model cards, data docs, control-to-evidence matrices, approval records



Manual checks

-  Who produces evidence
-  Who decides
-  Who can block



Pipeline automation

-  Who produces evidence
-  Who decides
-  Who can block



Full CI/CD with continuous training

-  Who produces evidence
-  Who decides
-  Who can block



Maturity ladder: manual checks → pipeline automation → full CI/CD with continuous training

Action Plan: What You Change in Your Own Workflow



Data

- Data: put a first contract on the dataset whose changes burned you most



Owner: _____



Target date: _____



Method

- Method: audit your validation split for temporal, grouped, or duplicate leakage



Owner: _____



Target date: _____



Process

- Process: add an evaluation gate with an absolute floor and regression guard



Owner: _____



Target date: _____



Review

- Review: classify severity so blockers and suggestions are treated differently



Owner: _____



Target date: _____



People

- People: set a review cadence, name an owner, schedule reviewer calibration



Owner: _____



Target date: _____



PersonWise.

PersonWise • AI digital-human course platform



Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/10a8560c/public