

Task-Scoped Evidence Profiles for External Data Catalog Selection and Traceable Machine Learning

Ruobing Wang^{1, 2, *} and Xingzhi Wang²

¹ Khoury College of Computer Sciences, Northeastern University

² College of Engineering - Information Networking Institute, Carnegie Mellon University

* Corresponding author: ruobing2@alumni.cmu.edu

ARTICLE INFO Received January 16, 2024 Revised June 03, 2024 Accepted September 06, 2024 Available Online December 30, 2024 DOI 10.63646/ITLF6323 License Creative Commons Attribution 4.0 International Licence (CC BY 4.0) Publisher INATGI, United States of America Journal JAIAA - ISSN 3067-7386	Abstract External data catalogs improve discovery but do not by themselves establish suitability for a specified machine-learning use. This technical report defines a task-scoped evidence profile linking requirement sets, identified source versions, selection decisions and subsequent execution. The profile distinguishes provider assertions, measurements and AI suggestions; evaluates their applicability to a task and role; and propagates eligible, excluded and deferred outcomes across assets, bundles and run inputs. A pre-run manifest records alternatives and unresolved conditions, while a separate run record is required to substantiate later fitting behavior. A focused synthesis of literature published through 2023 positions the contribution as an application-level decision and record contract. The report specifies structural and semantic invariants, conservative revalidation rules, and verification protocols for Iris, monthly NYC–NOAA regression and red-wine classification. The proposed monthly case uses a 2015–2023 observation window, subject to independently verified historical source availability. Calendar dimensions, model settings and illustrative decisions are design specifications rather than observed experimental results. No verified prepublication execution archive is established here, and no predictive-performance, conformance-rate or workflow-efficiency result is claimed. The contribution is an inspectable methodology and evaluation design; implementation conformance, independent requirement adequacy and comparative benefit remain to be tested. Keywords: External data catalogs; Data selection; Evidence applicability; Data provenance; Machine learning; Reproducibility
--	--

I INTRODUCTION

External data catalogs help analysts find resources distributed across organizations and storage systems. A discovered resource, however, still needs to be connected to a particular analytical use. Its population may differ from the target population, a numerical field may measure the wrong quantity, or a current historical snapshot may contain revisions that were unavailable at the intended prediction time. Dataset search and catalog research establish important discovery and integration capabilities, but a retrieval result does not by itself resolve these conditions (Chapman et al., 2020; Ehrlinger et al., 2021).

The problem is especially visible when several individually plausible assets are combined. A feature file and a label file may each be complete while their keys produce a many-to-many join. Two partitions may contain different row identifiers but identical feature vectors. An analyst may retain a model and its training script without retaining the reasons for rejecting another source. These failures involve relationships between tasks, evidence, inputs and activities. They cannot be reduced to a single dataset quality score. Consumer-oriented quality provides the appropriate starting point: requirements depend on the intended use (Wang & Strong, 1996; Gong et al., 2023).

This article specifies a task-scoped application profile for recording those relationships. The profile identifies a requirement set, distinguishes provider assertions from measured observations and AI suggestions, evaluates evidence against its role and version, and aggregates mandatory checks at asset, bundle and run-input levels. A pre-run selection manifest records the decision, reasons, alternatives and future execution obligations. A later run record binds the inputs consumed and the observed fitting

behavior to that manifest. The profile is compatible with established metadata and provenance representations rather than a replacement for them.

The scope is a methodological technical report supported by a focused literature synthesis and explicit evaluation designs. The central questions are: RQ1, which records and constraints make a task-specific selection decision inspectable; RQ2, which invariants and tests are needed to verify an implementation under faults and source changes; and RQ3, how the contract can be instantiated for public-data tasks while preserving temporal and evidential boundaries. RQ1 is addressed by the specification; RQ2 and RQ3 define testable procedures rather than completed empirical findings. Here, AI denotes the downstream analytical setting and a possible source of discovery proposals. No new language-model retriever or automated semantic expert is introduced.

The contribution combines an explicit requirement-set identity, evidence applicability, three-level conditional decisions, retained alternatives and a selection-to-execution link. Version identifiers, checksums, joins, partition descriptions, validation and provenance histories have substantial precedents. The proposed contract makes their task-specific obligations explicit. This report does not establish that the requirements cover every scientifically relevant condition or that the profile outperforms an existing catalog, validator or workflow manager. Sections V and VI specify the evidence needed to evaluate those claims.

II RELATED WORK AND DESIGN RATIONALE

A Focused synthesis and source selection

The literature synthesis is restricted to sources published by 31 December 2023. It connects dataset discovery and catalogs, task-dependent data quality and leakage, dataset and model documentation, and metadata standards and provenance. Sources are included when they provide a relevant mechanism, representation, documented failure mode or design rationale. Foundational work has no lower date cutoff. This is a focused narrative synthesis, not an estimate of how frequently capabilities occur across all systems.

The synthesis uses published research, standards and source documentation to support the design argument. A complete original search log and fixed bibliographic export are not available, so exhaustive coverage, reproducible systematic screening and retrospective screening counts are not claimed. Table I maps the principal source families to local obligations. The comparisons concern the cited versions and mechanisms; a resolving DOI establishes bibliographic identity but does not verify every interpretation of a source.

Implemented discovery systems illustrate complementary ways to expose candidate data. Google Dataset Search indexes structured metadata published on the Web (Brickley et al., 2019); Goods organizes enterprise datasets using collected metadata and relationships (Halevy et al., 2016); and Aurum supports discovery through a graph of relationships among data elements (Castro Fernandez et al., 2018). These mechanisms motivate retaining discovery evidence while evaluating task-specific eligibility in a separate decision record.

Table I. Evidence to design mapping for the core requirements

Source family and type	Evidence used	Local requirement
Wang and Strong (1996); Gong et al. (2023) Quality research	Quality depends on intended use, not numerical completeness alone.	Identify task, role, population, measurement and requirement owner.
Breck et al. (2019); Kapoor and Narayanan (2023) Validation and leakage	Data properties and information dependencies can invalidate executable pipelines.	Retain scoped measurements of joins, partitions and fitting inputs.
Geburu et al. (2021); Mitchell et al. (2019) Documentation	Descriptions record composition, use and limitations.	Link documentation without treating it as a current measurement.

Source family and type	Evidence used	Local requirement
DCAT 2 (W3C, 2020) Catalog vocabulary	RDF descriptions identify datasets, distributions, services and relationships.	Bind each assessment to an identified distribution and its evidence.
PROV-O (W3C, 2013); ModelDB (Vartak et al., 2016); RO-Crate (Soiland-Reyes et al., 2022)	Activities, model histories and research-object context have existing representations.	Specify pre-run requirements, unresolved predicates and alternatives.

B Atomic comparison with the closest mechanisms

Table II compares mechanisms individually. Representational support is distinguished from implemented checking and from demonstrated application. A capability not reported in a cited source is a boundary of this comparison, not proof that an extensible standard or system cannot express it. No cited implementation is executed as a competitive baseline in this report.

Table II. Closest mechanisms and the precise local extension

Source and scope	Established mechanism	Local extension or constraint
DCAT 2 (W3C, 2020) Sections 5–6 and 14	Catalog descriptions of resources, distributions and services; application profiles.	Distinguish resource description from eligibility for a named task and role.
Breck et al. (2019) Sections 2–5	Schema and statistics-based validation, anomaly reporting and revalidation.	Preserve unresolved evidence and propagate decisions across levels.
ModelDB (Vartak et al., 2016) Section 3 and Figure 3	Model-management records for stages, preprocessing parameters and histories.	Link run histories to selection reasons and evaluated alternatives.
RO-Crate (Soiland-Reyes et al., 2022)	Packaging of research outputs with metadata about context and dependencies.	Specify the pre-run decision record; validate any packaging integration separately.

Foundational provenance research distinguishes the source contributions that explain an output from the source locations from which its values originate (Buneman et al., 2001). Scientific workflow research examines provenance across computational processes (Davidson & Freire, 2008), and the Open Provenance Model provides a shared representation intended to support exchange between systems (Moreau et al., 2011). The selection rationale considered here adds a task-level account of why an input was accepted; it is not identical to query-level why-provenance.

RO-Crate provides a precedent for packaging research outputs together with metadata about their context and dependencies (Soiland-Reyes et al., 2022). The profile proposed here adds explicit records of requirements, evidence applicability and selection alternatives before fitting. For a PROV-O mapping, source artifacts and manifests can be represented as entities, selection and fitting as separate activities, and responsible parties as agents (W3C, 2013). This is a proposed alignment, not a supplied export or a claim of RO-Crate conformance. An implementation must validate its chosen interchange profile separately.

The incremental record can be illustrated by a single event. A catalog describes two weather stations and a taxi outcome. Existing metadata can identify the distributions; a validator can compute coverage; a workflow system can retain the fitted model. The additional selection record states which requirement set made coverage mandatory, which source version was measured, why one eligible station was preferred, why another role remained deferred, and which obligations were still pending at selection time. This is an application-level specification of record contents and constraints. It is not a claim that general provenance vocabularies lack sufficient expressive power.

C Design tradeoffs

Explicit unknowns prevent an absent release history from silently becoming a passing temporal check, but they reduce the fraction of cases that can receive a final eligible or excluded decision. Stronger source freezing improves byte-level reproducibility while adding storage and maintenance work. Selective revalidation can avoid repeating unaffected predicates, but only when the dependency model is adequate. The conservative fallback is complete recomputation when task identity, dependency coverage or change observation is uncertain. These are costs and design choices to be evaluated, not benefits established merely by adopting a manifest.

FAIR principles and their assessment frameworks support discoverability, interoperability and reuse; they do not certify suitability for every predictive task (Wilkinson et al., 2016, 2018). Similarly, dataset documentation, model documentation and the Data Quality Vocabulary provide useful descriptions and measurements without identifying every condition a particular scientific use requires (Gebru et al., 2021; Mitchell et al., 2019; W3C, 2016). Data statements emphasize documenting population and collection context in natural language processing (Bender & Friedman, 2018), while Data Cards organize dataset facts and decisions for different stakeholders throughout the data lifecycle (Pushkarna et al., 2022). The profile therefore keeps requirement adequacy distinct from requirement execution.

III TASK AND EVIDENCE SEMANTICS

A Requirement sets define the meaning of eligibility

A selection unit is a tuple containing task version, subject version, role, information context and decision level. The task specifies the target, population, observation unit, measurement definitions, temporal horizon and evaluation design. A role explains how an asset will be used, such as an outcome, historical predictor, retrospective covariate or label. The same distribution may pass one role and remain unresolved for another. External refers to a catalog holding or exposing descriptions outside the source system’s local data-management context; it does not imply that a different legal organization must operate the catalog.

Each task points to a requirement set with an identifier, version, owner, rationale and residual-risk list. The owner is accountable for selecting the mandatory predicates and explaining their relationship to the limited intended use. Eligible means that all mandatory checks in that identified requirement set pass for the stated task, role and artifact version. It does not establish that the requirement set covers every scientifically relevant condition. A complete file with a reproducible split can still have an unsuitable label definition or an unrepresentative sampling process.

Requirement adequacy requires a separate assessment. For the NYC case, full station coverage does not establish that one station represents all citywide travel conditions. For wine, disjoint exact-feature groups do not establish disjoint production batches. These risks are retained even when the narrower computational exercise is eligible. A stronger role must introduce the corresponding requirement, evidence and decision. It cannot inherit eligibility solely because it uses the same bytes. Updating a requirement set creates a new decision context and triggers reevaluation (Table III).

Table III. Minimum task and requirement information

Component	Required content	Case implication
Task identity	Stable task ID and explicit version.	Retrospective analysis and operational forecasting have distinct roles.
Population and unit	Entities, observation unit and exclusions.	Yellow monthly averages differ from Manhattan daily pickups.
Measurement	Targets, feature definitions, native units and conversions.	Mean daily rainfall within a month differs from a monthly total.
Information context	Decision cutoff and permissible information.	Current revised history does not establish past availability.

Component	Required content	Case implication
Evaluation design	Partition rule, group definition and fitting restrictions.	Exact-feature groups are not production-batch identities.
Requirement governance	ID, version, owner, rationale and residual risks.	Rule execution differs from rule adequacy.
Selection policy	Mandatory checks followed by optional preference.	Distance ranks eligible stations without proving representativeness.

B Evidence origin and applicability

The architecture in Figure 1 distinguishes three origins. A provider assertion describes what a source claims, including variable definitions or use conditions. A measured observation reports a declared inspection of identified content. An AI suggestion proposes an interpretation or correspondence. Similar prose does not make these evidence types interchangeable. For example, a completeness statement cannot override a measured missing value, and a generated assertion of compatible keys cannot replace a key-cardinality check.

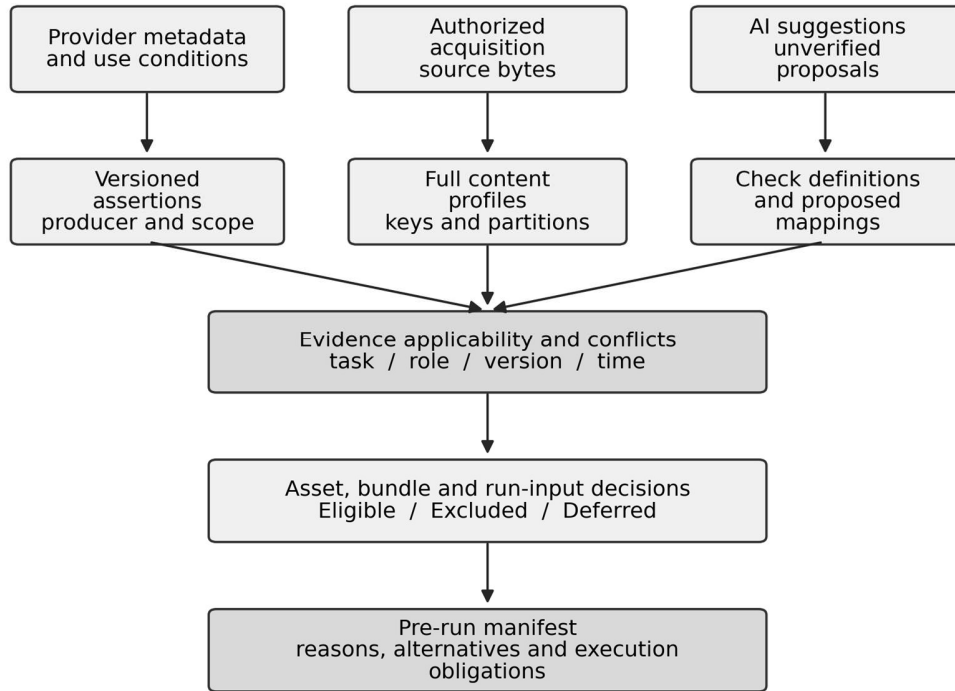


Figure 1. Architecture of task-scoped evidence assessment. Authorized acquisition supplies bytes for measurement. AI suggestions enter a verification path and cannot directly supply a passing observation.

An evidence item identifies its producer, subject version or digest, role, observation time, method, scope and substantive outcome. Optional validity boundaries use a half-open interval: the beginning is included and the expiry is excluded. A task may also set a maximum evidence age. Evidence observed after a decision cannot support that earlier decision. Immutable source bytes can remain valid without an arbitrary age limit, but a changed policy or task may still make a prior observation inapplicable. Repeating a copied assertion does not create an independent origin.

Applicability is evaluated before substantive aggregation. A wrong subject digest, wrong role, insufficient measurement scope, missing origin or expired assertion makes an item inapplicable to the current predicate. If no applicable item resolves a mandatory predicate, its result is unknown.

Conflicting applicable records are retained and require an explicit resolution; the most convenient

statement must not be chosen silently. The proposed bounded resolver assumes full coverage of the scope required by a predicate. It does not authenticate provider identities, interpret arbitrary legal terms or automatically resolve competing scientific definitions.

Source acquisition and authorization remain distinct from suitability. A source system continues to enforce access controls. Retained terms and an analyst's use review provide evidence about a declared processing scope, not universal clearance. ODRL represents policy statements without supplying the factual interpretation of every permission condition (W3C, 2018). Each proposed public-data case therefore requires an explicit review of the intended processing and redistribution scope; a local eligibility record does not replace source authorization.

C Three decision levels and failure precedence

Asset checks address an identified distribution in a role, including population, variable meaning, content scope and any required availability or use evidence. Bundle checks address relationships between members, including key uniqueness, coverage and cardinality. Data profiling supplies relevant measurements, including null-value statistics, unique-column combinations and dependencies (Abedjan et al., 2015); the requirement set determines which measurements constrain the intended use. Run-input checks inspect the actual planned inputs and partitions. Upstream results propagate to dependent levels: an otherwise valid split does not repair an excluded input asset. Lower-level eligibility also does not predetermine the outcome of a higher-level check.

For any nonempty mandatory-check set, an applicable failure yields excluded. If there is no failure but at least one unknown, the decision is deferred. Only an all-pass set is eligible. If both failed and unknown predicates occur, exclusion controls the aggregate while both kinds of evidence remain visible. Empty requirement sets and malformed records are validation errors; they are not meaningful eligibility decisions. Figure 2 distinguishes invalid input from a valid record whose scientific evidence is unresolved.

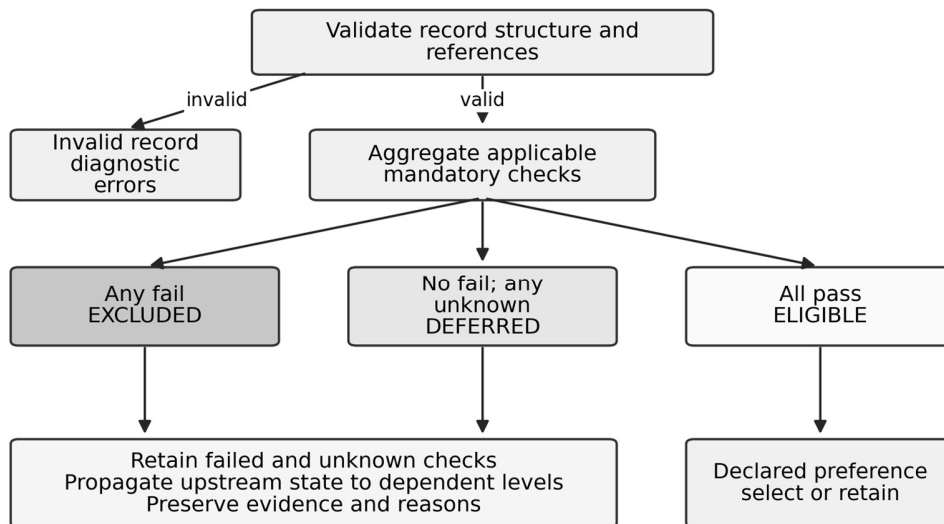


Figure 2. Decision aggregation with failure precedence. Record validation errors are separate from eligible, excluded and deferred task decisions. Preference is applied only within the eligible set.

Selection is a separate Boolean decision. An eligible resource may remain unselected because another eligible candidate better meets a declared preference, or because no acquisition or execution is planned. Excluded and deferred candidates cannot be selected by this profile. The selection record preserves each evaluated alternative, its decision and its reason. A candidate outside a stated retrieval or acquisition budget is recorded as unassessed rather than being assigned an evidence-based exclusion.

Future execution checks are obligations. Before fitting, the manifest can require that a scaler use only training rows, but it cannot report that this future fit has already been observed. After execution, the run record identifies the actual fitting inputs and retained parameters, and verifies the obligation. This separation prevents a workflow plan from being counted as successful execution evidence. Figure 3 makes deterministic conversions before partitioning distinct from learned transformations fitted after partitioning.

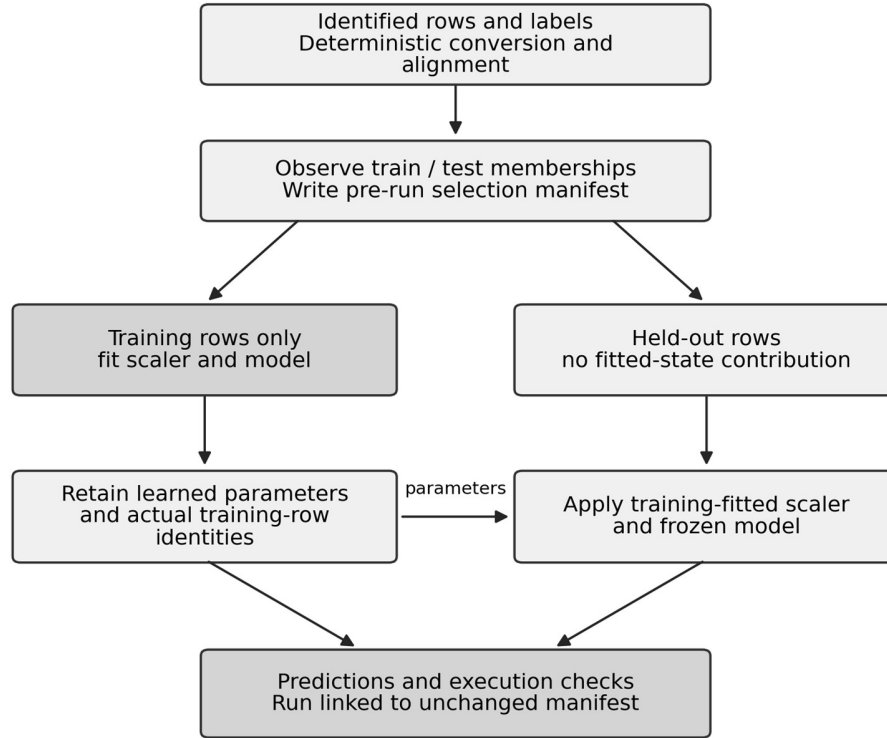


Figure 3. Required order of preparation, selection and execution. Deterministic conversion and keyed alignment precede partition verification. Learned transformations use training rows only, and a later run record must identify the inputs actually used.

IV MANIFEST CONTRACT AND IMPLEMENTATION DESIGN

A Required fields and cross-record invariants

The manifest is specified as a local application contract. Table IV defines its field groups and semantic obligations, separating structural requirements from relationships that require access to identified artifacts. An implementation should publish a versioned machine-readable schema and validate both layers. This article specifies the contract and its validation procedure; it does not claim that a validated schema, executable package or execution archive was released before publication. Field names are proposed serialization names and must remain stable within an adopted schema version.

Table IV. Manifest field groups and semantic obligations

Field group	Proposed representation	Semantic obligation
Identity and task	schema_version, manifest_id, task_id, task_version, rule_version	Nonempty identities; explicit schema version; matching task and rule context.
Requirement set	id, version, owner, rationale, residual_risks	Identify the requirements actually applied and their uncovered risks.
Subject and role	subject_id, subject_sha256, level, role	Resolve subject to the artifact registry; verify digest and role.

Field group	Proposed representation	Semantic obligation
Checks	checks, mandatory_check_ids	Nonempty unique checks; mandatory IDs resolve exactly.
Evidence	predicate, scope, evaluator, evidence_refs	Pass/fail needs applicable evidence bound to retained artifacts.
Decision	decision, selected, unknown_check_ids	Aggregation and unknown IDs agree; selected implies eligible.
Artifact registry	artifacts_sha256	Safe package-relative paths and matching SHA-256 digests.
Choice	alternatives, selection_reason, ranking_policy	Retain evaluated alternatives; distinguish no ranking from missing data.
Execution	decision_at, execution_obligations	Timezone-aware times; future actions remain obligations until observed.
Change and limits	revalidation_dependencies, predecessor, limitations	Dependencies resolve; predecessor is null or an identified digest.

Structural validation errors stop consumption and return diagnostics. They are not converted to eligible or conflated with the deferred outcome of a valid scientific assessment. Semantic validation must check duplicate and missing check IDs, subject-to-artifact binding, task/role/version agreement, permitted outcomes, unknown-list consistency and selected-state constraints. In the bounded contract, every serialized check is mandatory and the mandatory list must match the check IDs exactly. Advisory checks require an explicit extension, rather than an undocumented interpretation. A passing or failing check must resolve to applicable evidence, including the assessed scope and a recorded reason. The execution protocol requires the runner to reread the retained input table and verify its digest after writing the pre-run manifest. Only the saved training indices may contribute to scaler and model fitting. The subsequent run record must cite the manifest digest, consumed-input digest, actual fitting times and output digests; the manifest is checked again after fitting. A separate numerical reconstruction should test the relationship between saved inputs, fitted parameters and predictions. These are required verification steps, not observed results. Retention of exact inputs, software versions and analysis parameters follows guidance for reproducible computational research (Sandve et al., 2013).

B Core responsibilities and task adapters

The proposed implementation is deliberately small and file based. A common core would provide deterministic serialization, SHA-256 artifact identities, three-state aggregation, evidence-applicability checks, manifest validation and provenance mapping. Task adapters would implement source parsing, variable definitions, population rules, feature construction and partitions. Table V separates these responsibilities from limitations that require additional evidence or infrastructure. The design needs neither a deployed catalog service nor language-model inference to test its core semantics.

Table V. Proposed implementation responsibilities and boundaries

Component	Proposed responsibility	Boundary
Common core	Serialization, aggregation, applicability and manifest validation.	No provider authentication or general policy interpreter.
Iris adapter	Identify table, measure groups and verify classification inputs.	No recovered plant identities or proven biological independence.
NYC–NOAA adapter	Parse sources, check coverage, convert units, join and partition.	Historical availability and station representativeness need separate evidence.

Component	Proposed responsibility	Boundary
Wine adapter	Align labels, define the target and construct group-disjoint partitions.	No batch reconstruction or independent label adjudication.
Revalidation	Recompute affected predicates with conservative fallback.	Depends on declared dependency coverage and observed changes.
Provenance mapping	Map artifacts to entities and decisions or runs to activities.	No demonstrated export interoperability or RO-Crate conformance.
Human validation	Requirement audit and matched comparison in Section VI.	No completed suitability or workflow-effort evaluation.

A historical execution claim requires a linked archive of source snapshots, software, code revision, requirements, split membership and dated run records. Such a prepublication archive is not established for the results formerly associated with these case designs; the present report therefore makes no claim that those numerical results were obtained before publication. Acquiring old observations at a later date or selecting an older dependency version cannot reconstruct an earlier execution. Any subsequent implementation or replication must retain its actual acquisition and execution dates and be reported separately from the historical article.

C Change handling and conservative fallback

Revalidation compares current dependency fingerprints with their predecessor values and identifies the checks that must be reconsidered. A changed feature table can affect content and group separation; a changed label key can affect alignment; a changed task or policy can affect the interpretation of otherwise identical bytes. Dependent decisions are always reaggregated. Reusing an unaffected predicate does not permit an outdated aggregate decision to remain eligible. Every successor links to its predecessor digest while retaining the earlier record.

The dependency model is itself an assumption. Unknown dependency keys, a changed task version or an unreliable event stream should trigger full recomputation. Direct fingerprint comparison supplements event notification, because an unchanged notification stream does not establish unchanged content. This fallback handles uncertainty represented in the interface; it does not guarantee discovery of hidden dependencies absent from both the task model and the observation mechanism. Figure 4 shows the selective and conservative paths to be tested.

D Trust assumptions

The profile assumes a trustworthy recorder and access to the retained bytes. SHA-256, specified in the Secure Hash Standard (National Institute of Standards and Technology, 2015), binds an artifact to a digest. Comparing that digest with a trusted retained value detects mismatched content relative to the record. These digests do not prove that a provider assertion is correct, that a local clock recorded an authentic historical event, or that a malicious recorder did not recreate an entire history. External signatures or trusted time anchors would be needed for stronger adversarial claims. This study makes no such claim. Byte integrity, numerical reproducibility and scientific validity remain distinct guarantees.

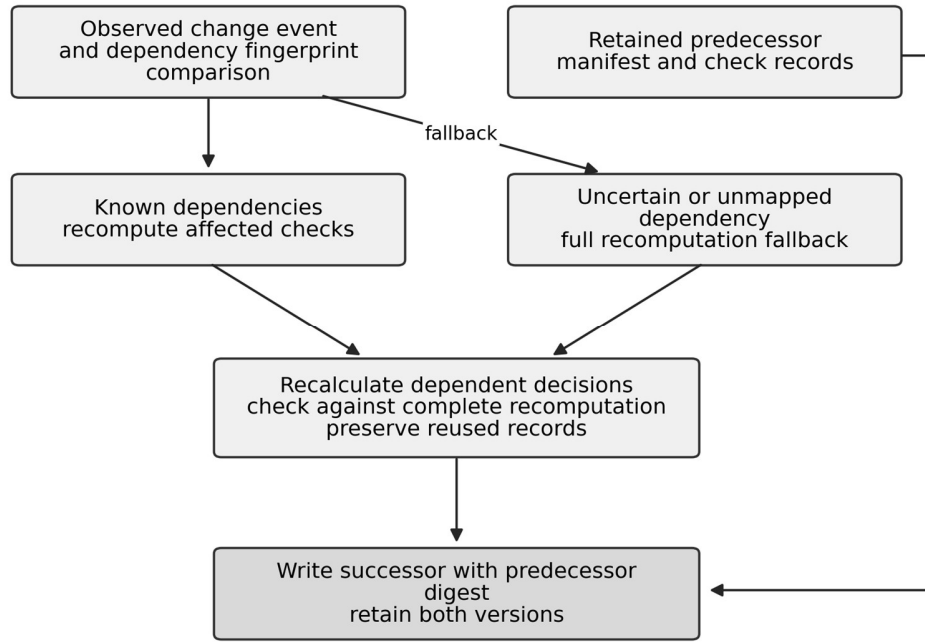


Figure 4. Proposed selective revalidation and complete-recompute fallback. Successor decisions preserve predecessor identities. Agreement with full recomputation is a verification obligation under the declared dependency model.

V VERIFICATION PROTOCOLS AND CASE DESIGNS

A Source eligibility environment and evidence scope

This section specifies prospective verification procedures and task instantiations. It reports no completed execution or newly acquired snapshot as evidence of a prepublication experiment. A historical analysis may use only source versions whose availability before the relevant publication cutoff is independently documented. Observation period, provider release identifier, acquisition time and execution time must be recorded separately. A current provider file containing older observations does not by itself establish the state of that file at the historical cutoff. Missing release evidence makes the historical-use predicate unknown.

A candidate environment using components released before 2024 is Python 3.11.7 (Python Software Foundation, 2023), NumPy 1.26.2 (NumPy developers, 2023), SciPy 1.11.4 (SciPy developers, 2023) and scikit-learn 1.3.2 (Scikit-learn developers, 2023). This is a proposed compatibility target, not a recovered original environment or a claim of successful execution. An implementation must resolve and lock transitive dependencies, operating-system details and numerical libraries, then record its actual runtime and validation results. NOAA collection citations identify the dataset family; a particular minor version or update must be supported by its own retained release evidence. No minor version is assigned here without that evidence.

Verification separates contract conformance, numerical consistency and independent scientific validity. Controlled expectations should be written before executing a test suite and kept distinct from domain-expert suitability judgments. Table VI specifies the layers of evaluation and their required outputs. Case families, roles and repeated versions share information, so their counts must not be pooled into a fictitious independent sample size. All planned counts and settings below are design choices unless explicitly identified as published dataset descriptions or calendar calculations.

Table VI. Planned verification layers and reporting requirements

Layer	Planned assessment	Required report
Decision conformance	Construct pass, fail, unknown and mixed cases at three levels.	Expected and observed outcomes with discrepancy reasons.
Evidence applicability	Vary role, subject, scope, origin, conflict and time boundaries.	Predicate-level applicability decisions and supporting records.
Manifest validation	Exercise valid and malformed records plus broken references.	Accepted cases and diagnostic rejection categories.
Sequential revalidation	Compare selective and complete recomputation after changes.	State agreement, dependency coverage, cost and predecessor integrity.
Numerical consistency	Apply identified Iris, NYC–NOAA and Wine case designs.	Actual splits, fitting inputs, parameters and independent reconstruction.
Independent validity	Audit requirements and adjudicate evidence under blinding.	Expert disagreements, suitability judgments and measured review effort.

B Controlled decisions applicability and input validation

A controlled suite can use a clearly identified Iris table as a small fixture base. Constructed conditions should include missing or incompatible units, nonfinite features, invalid labels, stale profiles, conflicting definitions, duplicate or unmatched label keys, row overlap, crossed exact-feature groups, unavailable chronology and an unbound artifact identity. Positive controls should include refreshed measurements and a declared millimeter-to-centimeter conversion. These are designed perturbations, not allegations about defects in the public distribution. Every case must state which predicates change and which remain fixed.

Table VII gives logical examples derived from the aggregation rule. They are expected decisions under explicit assumptions, not a measured confusion matrix. An implementation should record expected and observed states separately at asset, bundle and run-input levels and investigate every discrepancy. A zero observed error count in a finite constructed suite would establish only the tested conformance boundary; it would not estimate the rate of scientific suitability errors in deployed workflows.

Table VII. Illustrative aggregation cases under declared mandatory checks

Illustrative condition	Asset	Bundle	Run input
All mandatory predicates pass at every level.	Eligible	Eligible	Eligible
One asset predicate fails; all other checks pass.	Excluded	Excluded	Excluded
One asset predicate is unknown; no predicate fails.	Deferred	Deferred	Deferred
Asset checks pass; mandatory one-to-one join fails.	Eligible	Excluded	Excluded
Asset and bundle pass; row separation fails.	Eligible	Eligible	Excluded
Asset and bundle pass; required chronology is unknown.	Eligible	Eligible	Deferred
A failure and an unknown coexist at the asset level.	Excluded	Excluded	Excluded
Record structure or required references are invalid.	Diagnostic	No decision	No decision

Applicability tests should exercise another subject or role, partial measurement scope, an AI suggestion without verification, a missing producer or method, an observation after the decision, malformed times, conflicting applicable outcomes and half-open validity boundaries. A fresh applicable denial fails its predicate; after expiry, that item no longer resolves the current predicate. An immutable observation without a declared age limit is not rejected merely because time passes. Maximum-age values must be

finite and nonnegative, and the exact age boundary must be documented. Conflicts remain unknown until resolved under an explicit, recorded policy.

Manifest validation should accept coherent all-pass, deferred and failed records, including a failed record that also retains unknown predicates. Negative cases should cover missing or unexpected fields, wrong types, duplicate or unresolved IDs, context mismatches, unsupported pass/fail claims, absent artifacts, incorrect digests, unsafe paths, inconsistent unknown lists, selection of an ineligible subject, malformed times and broken predecessor or dependency references. The suite must separate input rejection from scientific decisions and report its tested coverage. No count of passed cases is asserted here.

C Sequential and combined changes

The proposed change sequence combines single and multiple perturbations, restoration, repeated delivery and incomplete change information. Table VIII states twelve event types and the response each is intended to test. An implementation should start from a documented valid fixture, retain the full state after every transition and apply the same predicate definitions to selective and complete recomputation. Expected states must be derived from the actual fixture, rather than copied from an unrelated run. Every valid successor must cite the prior manifest digest.

Table VIII. Planned sequential perturbations and verification obligations

Event	Planned perturbation	Required response to verify
1	Change a required unit.	Recheck measurement compatibility and dependent decisions.
2	Remove applicable profile evidence.	Preserve unknown predicates; retain existing failures.
3	Restore the unit without restoring the profile.	Do not infer that a missing profile has become valid.
4	Change a feature and introduce split overlap.	Reevaluate both dependencies and propagate the split failure.
5	Refresh the profile.	Resolve only checks supported by the new evidence.
6	Restore a valid partition.	Verify memberships and reaggregate all dependent states.
7	Change the task version.	Use full recomputation under the new task and requirements.
8	Introduce an applicable use denial.	Apply failure precedence for the specified use scope.
9	Change a label key.	Recheck alignment and any dependent feature-label table.
10	Deliver an unchanged event repeatedly.	Verify idempotent content decisions without losing audit history.
11	Omit a notification while changing source state.	Detect fingerprint changes; fall back when event completeness is uncertain.
12	Introduce an unmapped dependency.	Trigger full recomputation and record the unresolved dependency.

For each event, report the two paths' predicate outcomes, aggregate states, evaluated predicates, runtime and retained predecessor links. Reused predicates require unchanged dependencies and continuing evidence applicability. Count reductions alone do not establish latency, financial savings or reduced analyst effort, because hashing, storage and human interpretation have different costs. Agreement between two paths sharing predicate definitions also does not independently establish that the dependency model includes every scientifically relevant condition.

D Iris verification design

The Iris collection provides a small, historically established classification example with 150 observations and four measured features (Fisher, 1936). The proposed adapter identifies the exact packaged table and its digest before assigning local ordinal row IDs. If the corrected scikit-learn

distribution is used, that choice must be distinguished from any differently serialized UCI file. Finiteness, label domain, duplicate vectors and conflicting labels must be measured on the chosen bytes. The first fold of a five-fold StratifiedGroupKFold with shuffle enabled and random state 42 is a specified partition procedure; actual row and group counts must be reported after execution. The planned pipeline fits StandardScaler and logistic regression with $C=1$, solver=lbfgs and max_iter=1000 on training rows only. Verification should compare saved fitting memberships, scaler statistics and independently reconstructed predictions with the run outputs. Row-ID and exact-vector intersections between training and testing must be empty. No accuracy, class count or reconstruction tolerance is reported as an observed outcome here. Ordinal IDs identify retained rows, while exact-vector groups control one form of duplication; neither recovers original plant identities or proves biological independence.

E NYC and NOAA retrospective case design

The proposed task is retrospective monthly regression of citywide Yellow taxi average daily recorded trips from January 2015 through December 2023. This window excludes observations later than 2023 but remains conditional on verified source availability before the applicable publication cutoff. It is distinct from a Manhattan daily pickup forecast, a monthly trip total and unmet travel demand. The task definition requires a source field representing average recorded trips per day and a matching data dictionary. Observed contemporaneous weather and the preceding month's observed outcome define a retrospective analysis, not an operational forecast at a past decision time.

The acquisition design starts with the NYC TLC Industry Indicators catalog resource, identifier v6kb-cqej, and the NOAA GHCN-Daily collection (Menne, Durre, Korzeniewski, et al., 2012). The catalog response, metadata, data dictionary and downloaded distributions must each receive a retained identity and actual acquisition time. A proposed station filter covers latitudes 40.50–40.95 and longitudes –74.05 to –73.70. A maximum of three stations can be acquired in increasing great-circle distance from the declared reference point at 40.7831, –73.9712. These bounds and the budget are policy choices; no historical query result, matching-station count or acquired-station coverage is asserted.

Candidate assessment should distinguish transport populations, outcome meanings and station coverage. A numerically complete vehicle-count field cannot satisfy a trip-outcome requirement, and another transport sector cannot satisfy the specified Yellow population. Station completeness is a separate predicate from distance preference or spatial representativeness. Table IX records the proposed decision rules. Actual eligible stations and selection outcomes remain unresolved until the identified snapshots and all mandatory evidence have been checked. Candidates outside the acquisition budget are unassessed, not excluded.

Table IX. Candidate-role rules for the proposed monthly case

Candidate or role	Mandatory question	Decision rule
Yellow trip-rate field	Correct population, units, monthly coverage and historical availability?	Eligible only after every mandatory check passes.
Another transport sector	Does it represent the specified Yellow population?	Exclude for this role when the documented population differs.
Yellow vehicle-count fields	Do they measure average daily recorded trips?	Exclude for the trip-outcome role when the quantity is vehicles.
Station inside the geographic filter	Required variables, valid-day coverage, unique keys and source availability?	Measure each condition; defer missing evidence and exclude failures.
Multiple eligible stations	Which meets the declared distance preference?	Select by increasing distance, breaking exact ties by station ID.
Candidate outside acquisition budget	Was its content actually assessed?	Record as unassessed; do not infer an evidence-based exclusion.

Candidate or role	Mandatory question	Decision rule
Historical operational forecast	Were all predictors available at each prediction time?	Defer until release and revision evidence resolves availability.

The proposed weather requirement is at least 90% valid daily observations for each required element in every study month. The parser must verify the retained format specification, missing-value sentinel and quality-flag meaning before treating a value as valid. For the GHCN-Daily convention used in this design, −9999 denotes missing and a blank quality flag is required; measurement and source flags remain retained. The 2015–2023 calendar contains 3,287 days, giving 9,861 expected cells for three daily elements per station. This is a theoretical denominator, not a measured completeness count. Duplicate keys, valid-cell numerators and monthly coverage require actual file inspection. Distance is applied only after eligibility and does not certify citywide representativeness.

The adapter must verify native units against the retained NOAA specification (Menne, Durre, Vose, et al., 2012). For TMAX, TMIN and PRCP represented in tenths of their respective units, multiplying by 0.1 yields degrees Celsius or millimeters. Each monthly feature is the sum of valid converted daily values divided by the number of valid days. PRCP is therefore mean daily precipitation within the month, not accumulated monthly precipitation. As a mathematical illustration, thirty valid days totaling 900 source tenths of millimeters yield 3 mm as the mean daily value and 90 mm as the monthly total. No imputation is specified; a failed coverage rule remains a failed check. Table X defines the target and six features.

Table X. Variable dictionary for the proposed monthly model

Variable	Definition and unit	Information context
y	Monthly average daily recorded Yellow trips; trips/day.	Retrospective outcome; not a monthly total.
lag_y	Previous calendar month's observed y; trips/day.	First lag requires eligible December 2014 data.
TMAX	Monthly mean of valid daily maximum temperatures; degrees C.	Observed weather in the target month.
TMIN	Monthly mean of valid daily minimum temperatures; degrees C.	Observed weather in the target month.
PRCP	Monthly mean of valid daily precipitation; mm/day.	Mean daily depth, including valid zero-rain days.
month_sin	$\sin(2\pi m/12)$, for calendar month $m=1-12$.	Known calendar feature; dimensionless.
month_cos	$\cos(2\pi m/12)$, for calendar month $m=1-12$.	Known calendar feature; dimensionless.

With complete monthly inputs, the candidate window contains 108 unique month keys and 648 cells across six features. Training is specified for January 2015–December 2021, comprising 84 calendar months; January 2022–December 2023 supplies 24 holdout months. These are planned dimensions, not observed sample counts. The first lag requires an eligible December 2014 target value; if absent, the first row is removed and all dimensions and coverage reports are updated. StandardScaler and Ridge regression with $\alpha=1$ and $\text{solver}=\text{svd}$ are fitted using training rows only (Pedregosa et al., 2011). The pre-run manifest must be retained unchanged and the actual fitting record appended separately.

Table XI. Planned regression comparison on a shared temporal holdout

Path	Fitting information	Holdout and required metrics
Ridge with scaling	Training period 2015–2021; $\alpha=1$; $\text{solver}=\text{svd}$.	Same eligible 2022–2023 months; report MAE and RMSE.
Previous observed month	No fitted parameters; use the preceding observed target.	Identical months and target values; report MAE and RMSE.

Path	Fitting information	Holdout and required metrics
Comparison conditions	No test-driven tuning or alteration of inclusion rules.	Report actual n, exclusions, predictions and information-availability limits.

Both paths in Table XI must be evaluated on identical eligible holdout rows using mean absolute error and root mean squared error in trips per day. The final report must identify missing months and the denominator used by each metric; it must not substitute zeros for unperformed evaluation. No ordering of model performance is claimed. A later holdout month uses the previous observed outcome, so this is a conditional retrospective design rather than a single-origin multimonth forecast. An operational forecasting role remains deferred without historically valid availability and revision evidence. A claim about weather's predictive value would additionally require matched models with and without weather.

F Wine classification and group diagnostics design

The UCI Wine Quality collection describes 1,599 red-wine records and 4,898 white-wine records with eleven physicochemical inputs and an ordered quality score (Cortez et al., 2009). These are published dataset dimensions, not a new completeness audit. The proposed task is fixed-table red-wine classification with class 1 defined as quality at least 6. The threshold is a modeling choice, not an industry grading standard. The white-wine file provides a population-mismatch alternative. The chosen red-wine snapshot must be checked for finite features and the declared integer score domain 0–10. Native numerical scales are preserved without claiming physical-unit harmonization.

Assign local row IDs, export labels in a different order and verify a one-to-one key join before constructing the analysis table. Retained IDs identify rows, not original wines, wineries or production batches. Group rows by exact parsed feature vectors and report group sizes, repeated groups and conflicts in both original scores and thresholded labels. Conflicts must remain visible and be handled under a stated policy. No group count, conflict count or successful alignment result is presumed before the relevant snapshot is inspected.

The proposed partition uses the first fold of StratifiedGroupKFold with five splits, shuffle enabled and random state 42. Both row-ID and exact-feature-group intersections must be empty, and actual row, group and class counts must be retained for each partition. Group sizes may prevent an exact percentage split, so fixed row counts are not assumed. The planned scaler and logistic regression use $C=1$, solver=lbfgs and max_iter=1000. The majority-class baseline is chosen from training labels only, with class 0 chosen if training counts tie. Neither model uses held-out labels for fitting or tuning.

Table XII. Planned wine classification comparison

Path	Specified procedure	Outputs to report
Logistic regression	Training-only scaling; $C=1$; lbfgs; max_iter=1000.	Actual confusion counts, accuracy, balanced accuracy and class-1 F1.
Training-majority constant	Select the majority using training labels only; choose class 0 in a tie.	The same metrics on identical held-out rows and labels.
Shared design	Group-disjoint first fold, five splits, shuffle, seed 42.	Row/group/class counts, label conflicts, fitting records and exclusions.

Table XIII. Symbolic binary confusion matrix for metric reporting

Reference class	Predicted 0	Predicted 1	Total
0	TN	FP	TN + FP
1	FN	TP	FN + TP
Total	TN + FN	FP + TP	N

Table XIII defines counts to be populated only after execution. With N equal to $TN+FP+FN+TP$, accuracy is $(TN+TP)/N$. Balanced accuracy is one half of $[TN/(TN+FP)+TP/(TP+FN)]$, and class-1 F1 is $2TP/(2TP+FP+FN)$. Each denominator and the handling of undefined quantities must be reported. A zero denominator is not evidence of zero error. No observed confusion counts, predictive scores or floating-point agreement values are asserted in this methodological report.

An independent numerical checker should read the retained joined table, split memberships and model parameters without importing the case runner. It should compare aligned values and thresholded labels with the source snapshot, reconstruct probabilities and predictions, and report deviations under declared absolute and relative tolerances. This is a computational check rather than an assessment of chemical meaning or sampling validity. Exact-feature grouping cannot supply missing production-batch identities; a future-batch role remains deferred until the corresponding identifiers, temporal evidence and partition are available.

G Interpretation and reproducibility requirements

Appendix A specifies the artifacts and checks required for a reproducibility release. It is a release specification, not a claim that a verified historical package accompanies this report. A successful replication must retain the actual source bytes, code, environment, run records and independent check outputs, with a resolvable archive identifier and accessible files. A later acquisition creates a new snapshot and cannot replace the identity of an earlier observation. Availability commitments must be distinguished from successful independent retrieval and execution (Stodden et al., 2018).

The cases illustrate how different adapters can instantiate the same contract and how unresolved conditions remain visible. They do not supply measured conformance rates, predictive performance or external validation across independently sampled domains. Candidate roles may share a provider file, and all designs are chosen purposively. A future numerical result would establish only the reported calculation on the identified inputs; it would not by itself quantify the causal benefit of the selection profile. Computational reproducibility and independent scientific validation remain distinct objectives (Peng, 2011).

VI INDEPENDENT VALIDATION AND FAIR COMPARISON PROTOCOL

A Requirement adequacy before label agreement

Independent validation should first ask whether the requirement set is adequate for the declared use. Two domain-qualified reviewers, independent of rule design, should inspect the population, measurement, temporal context and proposed split before seeing the system's outcome. Each should identify missing conditions, explain why an existing condition is unnecessary or insufficient, and record residual scientific risks. This assessment is separate from applying the already-written predicates. Agreement on a deficient checklist would not establish task suitability.

After the requirement set and evidence package are fixed, the reviewers independently label each mandatory check pass, fail or unknown, cite evidence IDs and provide a reason. They then label the overall state and a separate relevance judgment. Initial submissions are locked before comparison; a third independent reviewer resolves disagreements while remaining blind to the system conclusion. If new evidence or requirements are introduced, the task and package receive new versions, and all compared systems are reevaluated against that version. Missing historical releases or batch identifiers may properly remain unknown after adjudication.

Appendix B specifies a neutral review record separating requirement-adequacy comments from predicate labels, reviewer qualifications, independence, evidence version and active review time. No independent participants, adjudicated labels, agreement statistics or review-time observations are reported. Automated or simulated labels may test software branches but cannot populate human-evaluation results. Before conducting the proposed study, the responsible authors must determine the

applicable institutional review, consent and reporting requirements; no approval or exemption is asserted here.

B Outcomes denominators and dependence

A future suitability evaluation should report an E/X/D confusion matrix separately for each decision level and task. Among system-eligible cases, the fraction labeled excluded by the reference review and the fraction labeled deferred should be reported separately. Both use the number of system-eligible cases as denominator; a zero denominator is not a zero error rate and is reported as not estimable. Nondeferred coverage is the number of eligible plus excluded decisions divided by all evaluated decisions. Selection rate is a different quantity. Reporting deferral besides acceptance errors prevents an all-deferred system from appearing operationally successful.

Task and source identifiers must remain available in analysis. Repeated roles, assets sharing a distribution and successive decisions share information and are not independent experimental units. A two-task pilot should use descriptive counts and documented disagreements. A larger study can estimate uncertainty with a design that respects clustering by task or source. It should not obtain nominal precision by counting every predicate, role or version as an independent application.

C Comparison of workflow benefit

The relevant baseline for record benefit is an existing validation-and-run-record workflow given the same candidates, source evidence, task requirements, computational budget and permissions. The compared profile adds explicit selection records and constraints, not extra privileged evidence. Reviewers can then answer predefined reconstruction questions, such as which source version was used, why an eligible alternative was not selected, which rows fitted the scaler, and which changed evidence invalidates a decision. Accuracy is checked against retained artifacts, and actual active time is measured jointly with correctness.

Discovery quality requires a distinct comparison under the same authorized catalog snapshot and candidate budget, with independently assigned relevance labels. Lexical retrieval, semantic retrieval and metadata-assisted methods may be compared, but metadata vocabulary alone is not a retrieval algorithm. Schema matching resources can support correspondence evaluation without resolving policy and historical-availability requirements (Koutras et al., 2021). Any downstream predictive comparison must fix model family, tuning budget, permitted information and partitions before attributing a result to changed input selection. Those comparative experiments are proposed, not results of the present study.

VII DISCUSSION

A Scope of the methodological contribution

RQ1 is addressed by the task, evidence and manifest specification, including explicit requirement identities, failure precedence and a separation of selection from execution. RQ2 is operationalized through controlled faults, applicability boundaries, semantic validation and sequential-revalidation protocols. RQ3 is illustrated through Iris, NYC–NOAA and Wine task designs that identify the required source, partition and fitting records. The latter two questions remain empirical obligations for a dated implementation study. A coherent specification is not evidence that an implementation has passed it, and successful software execution would not independently establish scientific suitability. The strongest practical use of the profile is explanatory discipline. A decision refers to a particular requirement and evidence version instead of an unexplained readiness score. Deferred is an informative result when available evidence does not establish historical availability or the requested group structure. It identifies a concrete next action: acquire the missing history, revise the role, obtain the relevant identifiers or narrow the use. It does not mean that the source is intrinsically poor or unusable for every purpose.

Comparison with prior work limits the novelty claim. DCAT 2 describes cataloged resources and their distributions; validation systems inspect data properties; documentation explains intended use; RO-Crate and provenance vocabularies organize research objects and their context (W3C, 2020; Soiland-Reyes et al., 2022). The proposed profile coordinates selected obligations when an input is approved for a particular task. A mature system may implement this contract through configuration or extension. The contribution is an explicit application profile and evaluation design, without a claim of replacing existing infrastructure or of demonstrated interoperability.

B Limitations and costs

The requirement sets and illustrative expectations are authored as part of the design. Cases are purposively chosen, and no executed conformance suite or independently adjudicated suitability sample is established for the historical article. The proposed core does not cover arbitrary file formats, sampling protocols, identity systems or policy interpretation. Its bounded applicability rule requires full scope; sampled evidence would need an uncertainty model and additional rules. Source and evaluator identifiers are assertions unless an implementation supplies authentication.

Historical consistency requires evidence about availability as well as observation dates. A later download can include revised values for an earlier period, and a collection-level citation cannot identify every field revision. Hashes identify bytes relative to retained records but do not prove that those bytes existed at an earlier date. This limits historical forecasting claims even when retrospective calculations can be reproduced. Older software version numbers likewise cannot establish an earlier execution. Missing historical evidence must remain visible rather than being replaced with an assumed timestamp.

Retaining files, profiles, requirements and alternatives consumes storage and analyst effort. More stringent requirements can increase deferral or make exploratory work cumbersome. Selective revalidation depends on maintained dependencies and a conservative fallback; no computational or operational saving is measured here. Practitioner research documents how neglected data work can propagate into downstream problems, making organizational effort relevant to deployment evaluation (Sambasivan et al., 2021). A deployment study must evaluate correctness and effort together, including missing requirements and evidence requests that change decisions.

AI-assisted discovery remains an optional proposal mechanism. Metadata can be incomplete, stale or misleading, and retrieved text must not override access controls or validation rules. An implementation should not execute provider prose or generated code merely because it is retrieved. If language-model proposals are integrated, model versions, prompts, retrieved evidence and acceptance checks should be retained separately. Retrieval-augmented generation and indirect prompt-injection research motivate this separation without constituting an evaluation of the proposed profile (Lewis et al., 2020; Greshake et al., 2023).

C Priorities for extension

The immediate research priority is independent assessment of requirement adequacy followed by blind, evidence-based adjudication. A subsequent study should use tasks and compatible-looking alternatives selected independently of the rule authors, including cases where population, measurement or time semantics remain difficult despite complete metadata. Only then should workflow-effort or discovery-benefit claims be tested with matched evidence and budgets. Extending the number of synthetic fixtures alone would not supply this missing evidence.

VIII CONCLUSION

This report specifies a task-scoped evidence profile connecting catalog descriptions, source identities, declared requirements and subsequent machine-learning execution. Evidence applicability, three-level decisions, retained alternatives and pending execution obligations are made explicit in a local manifest contract. The design complements established metadata, documentation and provenance mechanisms.

Its practical value is to make the evidence needed for an input decision inspectable and to preserve uncertainty when that evidence is absent.

The report provides verification procedures and case designs rather than verified prepublication experimental results. Eligibility remains conditional on the declared requirement set, source version and role. Implementation conformance, numerical reproducibility, independent requirement adequacy and comparative workflow benefit require distinct evidence. The protocols in Sections V and VI and the appendices define how those claims can be tested without confusing a later replication with a historical observation.

AUTHOR CONTRIBUTIONS

R. B. W. is for Conceptualization; Data Curation; Formal Analysis; Project Administration; Validation; Writing – Original Draft Preparation and X. Z. W. is for Supervision; Writing – Review & Editing; Methodology; Formal Analysis; Investigation; Validation.

DECLARATIONS

Conflicts of interest: No Conflicts of interest declared by authors.

Data and code availability: N/A.

Funding

N/A.

Generative AI assistance

AI tools were used to edit language, especially to sentences and native tones.

REFERENCES

- Abedjan, Z., Golab, L., & Naumann, F. (2015). Profiling relational data: A survey. *The VLDB Journal*, 24(4), 557–581. <https://doi.org/10.1007/s00778-015-0389-y>
- Bender, E. M., & Friedman, B. (2018). Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6, 587–604. https://doi.org/10.1162/tacl_a_00041
- Breck, E., Polyzotis, N., Roy, S., Whang, S. E., & Zinkevich, M. (2019). Data validation for machine learning. *Proceedings of Machine Learning and Systems*, 1. https://proceedings.mlsys.org/paper_files/paper/2019/file/928f1160e52192e3e0017fb63ab65391-Paper.pdf
- Brickley, D., Burgess, M., & Noy, N. (2019). Google Dataset Search: Building a search engine for datasets in an open Web ecosystem. *The World Wide Web Conference*, 1365–1375. <https://doi.org/10.1145/3308558.3313685>
- Buneman, P., Khanna, S., & Tan, W.-C. (2001). Why and where: A characterization of data provenance. *Database Theory—ICDT 2001, Lecture Notes in Computer Science*, 1973, 316–330. https://doi.org/10.1007/3-540-44503-X_20
- Castro Fernandez, R., Abedjan, Z., Koko, F., Yuan, G., Madden, S., & Stonebraker, M. (2018). Aurum: A data discovery system. *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, 1001–1012. <https://doi.org/10.1109/ICDE.2018.00094>
- Chapman, A., Simperl, E., Koesten, L., Konstantinidis, G., Ibáñez, L. D., Kacprzak, E., & Groth, P. (2020). Dataset search: A survey. *The VLDB Journal*, 29(1), 251–272. <https://doi.org/10.1007/s00778-019-00564-x>
- Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Wine Quality [Data set]. *UCI Machine Learning Repository*. <https://doi.org/10.24432/C56S3T>
- Davidson, S. B., & Freire, J. (2008). Provenance and scientific workflows: Challenges and opportunities. *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, 1345–1350. <https://doi.org/10.1145/1376616.1376772>
- Ehrlinger, L., Schrott, J., Melichar, M., Kirchmayr, N., & Wöb, W. (2021). Data catalogs: A systematic literature review and guidelines to implementation. *Database and Expert Systems Applications - DEXA 2021 Workshops*, 148–158. https://doi.org/10.1007/978-3-030-87101-7_15
- Fisher, R. (1936). Iris [Dataset]. *UCI Machine Learning Repository*. <https://doi.org/10.24432/C56C76>
- Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daumé, H., III, & Crawford, K. (2021). Datasheets for datasets. *Communications of the ACM*, 64(12), 86–92. <https://doi.org/10.1145/3458723>
- Gong, Y., Liu, G., Xue, Y., Li, R., & Meng, L. (2023). A survey on dataset quality in machine learning. *Information and Software Technology*, 162, 107268. <https://doi.org/10.1016/j.infsof.2023.107268>
- Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, 79–90. <https://doi.org/10.1145/3605764.3623985>
- Halevy, A., Korn, F., Noy, N. F., Olston, C., Polyzotis, N., Roy, S., & Whang, S. E. (2016). Goods: Organizing Google's datasets. *Proceedings of the 2016 International Conference on Management of Data*, 795–806. <https://doi.org/10.1145/2882903.2903730>
- Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9), 100804. <https://doi.org/10.1016/j.patter.2023.100804>
- Koutras, C., Siachamis, G., Ionescu, A., Psarakis, K., Brons, J., Fragkoulis, M., Lofi, C., Bonifati, A., & Katsifodimos, A. (2021). Valentine: Evaluating matching techniques for dataset discovery. *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, 468–479. <https://doi.org/10.1109/ICDE51399.2021.00047>
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33, 9459–9474. <https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html>
- Menne, M. J., Durre, I., Korzeniewski, B., McNeill, S., Thomas, K., Yin, X., Anthony, S., Ray, R., Vose, R. S., Gleason, B. E., & Houston, T. G. (2012). Global Historical Climatology Network - Daily (GHCN-Daily), Version 3 [Data set]. *NOAA National Climatic Data Center*. <https://doi.org/10.7289/V5D21VHZ>
- Menne, M. J., Durre, I., Vose, R. S., Gleason, B. E., & Houston, T. G. (2012). An overview of the Global Historical Climatology Network-Daily database. *Journal of Atmospheric and Oceanic Technology*, 29(7), 897–910. <https://doi.org/10.1175/JTECH-D-11-00103.1>
- Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., & Gebru, T. (2019). Model cards for model reporting. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 220–229. <https://doi.org/10.1145/3287560.3287596>

- Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., Kwasnikowska, N., Miles, S., Missier, P., Myers, J., Plale, B., Simmhan, Y., Stephan, E., & Van den Bussche, J. (2011). The Open Provenance Model core specification (v1.1). *Future Generation Computer Systems*, 27(6), 743–756. <https://doi.org/10.1016/j.future.2010.07.005>
- National Institute of Standards and Technology. (2015). Secure Hash Standard (SHS). Federal Information Processing Standards Publication 180-4. <https://doi.org/10.6028/NIST.FIPS.180-4>
- NumPy developers. (2023). NumPy 1.26.2 [Software release and release notes]. <https://github.com/numpy/numpy/releases/tag/v1.26.2>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830. <https://jmlr.org/papers/v12/pedregosa11a.html>
- Peng, R. D. (2011). Reproducible research in computational science. *Science*, 334(6060), 1226–1227. <https://doi.org/10.1126/science.1213847>
- Pushkarna, M., Zaldivar, A., & Kjartansson, O. (2022). Data Cards: Purposeful and transparent dataset documentation for responsible AI. 2022 ACM Conference on Fairness, Accountability, and Transparency, 1776–1826. <https://doi.org/10.1145/3531146.3533231>
- Python Software Foundation. (2023). Python 3.11.7 [Software release]. December 4, 2023. <https://www.python.org/downloads/release/python-3117/>
- Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P., & Aroyo, L. M. (2021). “Everyone wants to do the model work, not the data work”: Data cascades in high-stakes AI. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–15. <https://doi.org/10.1145/3411764.3445518>
- Sandve, G. K., Nekrutenko, A., Taylor, J., & Hovig, E. (2013). Ten simple rules for reproducible computational research. *PLoS Computational Biology*, 9(10), e1003285. <https://doi.org/10.1371/journal.pcbi.1003285>
- Scikit-learn developers. (2023). Version 1.3.2. Scikit-learn 1.3.2 documentation. October 2023. https://scikit-learn.org/1.3/whats_new/v1.3.html
- SciPy developers. (2023). SciPy 1.11.4 release notes. SciPy v1.11.4 Manual. <https://docs.scipy.org/doc/scipy-1.11.4/release/1.11.4-notes.html>
- Soiland-Reyes, S., Sefton, P., Castro, L. J., Coppens, F., Garijo, D., Leo, S., Portier, M., & Groth, P. (2022). Creating lightweight FAIR Digital Objects with RO-Crate. *Research Ideas and Outcomes*, 8, e93937. <https://doi.org/10.3897/rio.8.e93937>
- Stodden, V., Seiler, J., & Ma, Z. (2018). An empirical analysis of journal policy effectiveness for computational reproducibility. *Proceedings of the National Academy of Sciences*, 115(11), 2584–2589. <https://doi.org/10.1073/pnas.1708290115>
- Vartak, M., Subramanyam, H., Lee, W. E., Viswanathan, S., Husnoo, S., Madden, S., & Zaharia, M. (2016). ModelDB: A system for machine learning model management. *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, 1-3. <https://doi.org/10.1145/2939502.2939516>
- Wang, R. Y., & Strong, D. M. (1996). Beyond accuracy: What data quality means to data consumers. *Journal of Management Information Systems*, 12(4), 5-33. <https://doi.org/10.1080/07421222.1996.11518099>
- Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J. W., da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., ... Mons, B. (2016). The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, 3(1), 160018. <https://doi.org/10.1038/sdata.2016.18>
- Wilkinson, M. D., Sansone, S. A., Schultes, E., Doom, P., Bonino da Silva Santos, L. O., & Dumontier, M. (2018). A design framework and exemplar metrics for FAIRness. *Scientific Data*, 5(1), 180118. <https://doi.org/10.1038/sdata.2018.118>
- World Wide Web Consortium (W3C). (2013). PROV-O: The PROV ontology. W3C Recommendation, 30 April 2013. <https://www.w3.org/TR/2013/REC-prov-o-20130430/>
- World Wide Web Consortium (W3C). (2016). Data on the Web best practices: Data Quality Vocabulary. W3C Working Group Note, 15 December 2016. <https://www.w3.org/TR/2016/NOTE-vocab-dqv-20161215/>
- World Wide Web Consortium (W3C). (2018). ODRL Information Model 2.2. W3C Recommendation, 15 February 2018. <https://www.w3.org/TR/2018/REC-odrl-model-20180215/>
- World Wide Web Consortium (W3C). (2020). Data Catalog Vocabulary (DCAT) - Version 2. W3C Recommendation, 4 February 2020. <https://www.w3.org/TR/2020/REC-vocab-dcat-2-20200204/>

APPENDIX A REPRODUCIBILITY RELEASE SPECIFICATION

A reproducibility release must distinguish the historical source state from the actual execution environment. The candidate environment in Section V.A is a compatibility target only. Before claiming a reproducible result, archive the complete executable code, resolved dependency lock, identified source snapshots, protocol, test expectations, outputs and a verification report. Table XIV lists the minimum artifact families. The table specifies required contents and does not assert that these files were supplied or executed before publication.

Step 1. Verify artifact identities and historical source eligibility. Resolve every input and evidence reference, reject unsafe paths, compare SHA-256 digests and check documented release availability. Record missing evidence as unknown; do not replace it with a newly chosen historical date.

Step 2. Execute validation and the declared analysis. Check record structure and semantics, measure source properties, construct joins and group-aware or temporal partitions, and write the pre-run manifest. Verify input identities again; fit learned transformations using training rows only; then retain the actual run record, parameters, predictions and metrics.

Step 3. Independently verify the retained output. Read the saved inputs and parameters separately from the runner; reconstruct outputs; compare with declared tolerances; inspect selection and execution links; and report every discrepancy together with the software environment and actual verification time.

Store rerun outputs under a new identity. Fresh decision and fitting times necessarily create new records even if deterministic values agree. Compare source values, partition membership, fitted parameters, predictions and metrics separately from timestamp-bearing record digests. A changed implementation, source snapshot or requirement set requires a new version and new validation. Checksums provide integrity relative to the recorded inventory; stronger claims about historical existence require independent time evidence.

Table XIV. Minimum contents of a future reproducibility release

Artifact family	Required contents	Verification purpose
Protocol and environment	Task/rule versions, acquisition scope, code revision and dependency lock.	Identify the intended procedure and actual runtime.
Source and availability evidence	Bytes, provider metadata, release evidence, URLs and actual acquisition times.	Separate historical availability from later retrieval.
Contract and tests	Machine-readable schema, validator, fixtures and expected outcomes.	Inspect structural and semantic coverage.
Input preparation	Parsed tables, transformations, join diagnostics and coverage measurements.	Trace values from source bytes to consumed input.
Partition and fitting	Row/group memberships, class counts, scaler statistics and model parameters.	Verify separation and training-only fitting.
Decisions and outputs	Manifests, alternatives, predecessor links, run records and predictions.	Connect selection to actual execution and metrics.
Independent verification	Artifact inventory, reconstruction script, tolerances and discrepancy report.	Check integrity and numerical agreement without assuming historical existence.

APPENDIX B REQUIREMENT ADEQUACY AND REVIEW RECORDS

A neutral independent-review record should separate requirement assessment from predicate assessment. Requirement assessment asks whether the task omits relevant population, measurement, temporal, group-independence, permission or coverage conditions. Predicate assessment applies a fixed, identified requirement set to the evidence. Reviewers must state their expertise, independence and prior exposure to system outcomes. An actual person is responsible for each submission; structural form validation cannot authenticate that person or establish expertise.

For NYC, an adequacy review should address the scope of the published outcome, spatial representativeness of a selected station, the meaning of precipitation aggregation, prior-month outcome availability and the distinction between retrospective analysis and operational forecasting. For wine, it should address the binary score threshold, meaning of the native measurements, production-batch dependence, label reliability and the limits of exact-feature grouping. These questions identify what independent review should resolve; they are not completed expert findings.

The record must include task and requirement versions, subject and role, evidence IDs, proposed requirement additions or removals, per-check outcome and reason, overall state, relevance and active review time. Missing fields remain missing rather than being replaced with assumed agreement. New evidence requires a new package identity before labels are compared. A coordinator must assemble a neutral evidence subset that excludes system decisions, model-performance summaries and other outcome cues until the first independent labels are locked. This appendix specifies the record; it does not report completed reviews.