



CHECKLIST

Learning Checklist for AI and Machine Learning Projects

A practical, vendor-neutral checklist for validating AI and machine learning work before production use, with evidence, ownership, acceptance criteria, and readiness scoring.

Learning Checklist for AI and Machine Learning Projects

Use this checklist to review an AI or machine learning project before production use. It is designed to help technical teams verify evidence, assign ownership, and make a clear go/no-go decision.

How to use

- Work through the checklist in order.
- For each item, record objective evidence.
- Assign one accountable owner.
- Mark the acceptance decision: Pass, Fail, or Risk Accepted.
- Do not average out failures with stronger areas.
- If a check fails, fix it or document the risk and rationale.

Review record

Project name: _____ Model or system: _____
Review date: _____ Reviewer: _____ Decision
owner: _____ Target release date: _____



Readiness scoring

Score each section:

- 2 = Pass
- 1 = Partial / needs follow-up
- 0 = Fail

Scoring guide

- 10-12 per section: Strong readiness
- 7-9 per section: Ready with minor gaps
- 4-6 per section: Not ready; remediation needed
- 0-3 per section: High risk; do not proceed

Overall decision:

- Go if all critical items pass and total readiness is acceptable.
- No-Go if any critical item fails.
- Conditional Go only if risks are explicitly documented, owned, and time-bound.

1. Scope and learning objective

Purpose: Confirm the problem is well defined, measurable, and operationally relevant.

Item	Evidence to capture	Owner	Status
	Confirm the target outcome is stated in measurable terms, such as classification, ranking, forecasting, or anomaly detection.		Problem statement
	Review the business or operational decision the model will influence and document what action changes when the prediction changes.		Decision
	Validate that the target label, ground truth source, or proxy metric is available and stable enough for review.		Label source or metric definition
	Document the assumed environment, including data sources, latency constraints, and known edge cases.		Environment assumptions and constraints



Assign a single accountable owner for the learning objective and acceptance decision. | Named owner and approver | |

Acceptance criteria

- The objective is specific, measurable, and bounded.
- The owner can explain the expected production decision in one sentence.
- There is a clear way to tell whether the learning outcome is usable.

Section score: ___ / 10

2. Data readiness and representativeness

Purpose: Verify the data reflects production conditions and is suitable for learning.

Item	Evidence to capture	Owner	Status
Confirm the training, validation, and test splits are defined and protected from leakage.	Split methodology and leakage check		
Review whether the dataset is representative of the intended production population, time window, and usage patterns.	Population comp		
Validate that missing values, duplicates, outliers, and schema drift are measured, not assumed.	Data profile summary		
Document all feature sources, transformation steps, and manual filtering applied before training.	Lineage and preprocessing log		
Test whether the same input record could appear in more than one split, especially for time series, user-level, or asset-level data.	Dup		
Assign data ownership for each source, including update frequency and quality expectations.	Data owner registry		

Acceptance criteria

- No confirmed leakage.
- Split strategy matches the problem type.
- Dataset is representative enough for the intended decision domain.
- Data quality issues are within tolerance or explicitly accepted.

Section score: ___ / 12



3. Baseline and benchmark definition

Purpose: Establish a fair comparison against a simple alternative.

Item	Evidence to capture	Owner	Status
Confirm a simple baseline exists, such as a rule-based method, historical average, majority class, or prior model.	Baseline definition		
Review the baseline against the same data split and evaluation criteria used for the candidate model.	Baseline evaluation results		
Validate that the chosen benchmark reflects business cost, not just statistical convenience.	Cost rationale or benchmark note		
Document the threshold for "good enough" performance, including minimum improvement or acceptable error bands.	Acceptance threshold		
Assign a reviewer to sign off on whether the benchmark is fair and reproducible.	Reviewer sign-off		

Acceptance criteria

- The candidate model is compared against a reproducible baseline using the same test conditions.
- The improvement is enough to justify operational complexity, monitoring, and maintenance.

Section score: ___ / 10

4. Evaluation design and validation integrity

Purpose: Verify the evaluation is trustworthy and resistant to leakage, overfitting, or metric bias.

Item	Evidence to capture	Owner	Status
Confirm the evaluation metric matches the intended operational tradeoff, such as precision, recall, calibration, latency, or cost-weighted error.			
Review whether cross-validation, temporal validation, or holdout testing is appropriate for the problem structure.	Validation design not		



Validate that hyperparameter tuning did not use the final test set. | Tuning protocol or experiment log | |

Test that results are stable across multiple runs, folds, or time windows where randomness is expected. | Variance or repeatability result

Document confidence intervals, error slices, or subgroup results where they materially affect the decision. | Confidence and slice analysis

Assign a reviewer to verify that no evaluation shortcuts were used. | Review sign-off | |

Acceptance criteria

- The evaluation method matches the problem type.
- The final test set remains untouched until the end.
- The result is stable enough to support a production decision.

Section score: __ / 12

5. Operational constraints and integration fit

Purpose: Confirm the model can operate within the real system without creating hidden technical debt.

Item	Evidence to capture	Owner	Status
------	---------------------	-------	--------

Confirm the intended serving mode is defined, such as batch, online, near-real-time, or embedded in a workflow.	Serving mode decision
---	-----------------------

Review latency, throughput, memory, and compute assumptions against the target environment.	Capacity and performance estimate
---	-----------------------------------

Validate that runtime dependencies, feature availability, and deployment prerequisites are known.	Dependency and deployment checklist
---	-------------------------------------

Document fallback behavior if the model fails, degrades, or becomes unavailable.	Failure handling plan	
--	-----------------------	--

Verify integration points, interfaces, and data contracts with downstream systems.	Interface or contract documentation	
--	-------------------------------------	--

Assign ownership for runtime support, monitoring, and rollback.	Support owner and rollback owner	
---	----------------------------------	--

Acceptance criteria

- Serving mode is clear and feasible.



- Runtime constraints fit the target environment.
- Failure behavior and rollback are defined.
- Ownership for support and monitoring is assigned.

Section score: __ / 12

6. Monitoring, drift, and retraining plan

Purpose: Confirm the project can be observed and maintained after release.

Item	Evidence to capture	Owner	Status
Define the post-release metrics that will be monitored, including model quality and system performance.	Monitoring plan		
Specify the drift signals or trigger conditions that require review.	Drift thresholds		
Document the retraining cadence or retraining trigger logic.	Retraining policy		
Confirm that production data needed for monitoring is actually captured.	Logging and telemetry plan		
Assign an owner for ongoing review of alerts, drift, and retraining decisions.	Operations owner		

Acceptance criteria

- Monitoring covers both model behavior and system health.
- Drift and retraining triggers are explicit.
- The data needed to monitor the model will be available.
- An owner is responsible for ongoing review.

Section score: __ / 10

7. Risk, governance, and approval



Purpose: Ensure the work can be approved, audited, and defended.

Item	Evidence to capture	Owner	Status
Confirm any legal, privacy, security, or policy constraints have been reviewed.	Governance review note		
Document known limitations, including populations, conditions, or edge cases where the model should not be used.	Limitations statement		
Record whether human review is required for certain decisions.	Human-in-the-loop policy		
Capture the go/no-go decision, rationale, and any conditional approvals.	Decision record		
Assign named owners for all remediation items and due dates.	Action list		

Acceptance criteria

- All major risks are identified and owned.
- Approval conditions are explicit.
- Limitations are documented in language that operational teams can use.

Section score: ___ / 10

Final readiness summary

Critical failures:

- _____
- _____
- _____

Verified gaps:

- _____
- _____
- _____

Required follow-ups:



Item	Owner	Due date	Status

Overall readiness score: ___ / 66

Decision: ? Go ? Conditional Go ? No-Go

Decision rationale:

Approvals

- Reviewer: _____ Date: _____
- Decision owner: _____ Date: _____
- ML lead: _____ Date: _____

Common mistakes to avoid

- Starting with a dataset before defining the actual problem.
- Using a metric that does not map to the real decision.
- Randomly splitting data when the problem requires time-based separation.
- Comparing against a weak or undocumented baseline.
- Reporting only the best run and hiding variance.
- Releasing without a rollback or fallback plan.
- Treating monitoring as optional after deployment.



Quick pre-release check

Before release, confirm:

- The objective is measurable and owned.
- The data is representative and leakage-free.
- The baseline comparison is fair.
- The evaluation is valid and repeatable.
- Operational constraints are understood.
- Monitoring and retraining are defined.
- Governance and approvals are complete.

If any answer is no, the project is not ready for production use.