CHECKLIST

Checklist: Operational Readiness for AI and Machine Learning Programming

A practical checklist for technical teams to assess whether an AI or machine learning use case is suitable, validated, deployable, and monitorable in production.

Checklist: Operational Readiness for AI and Machine Learning Programming

Use this checklist to decide whether an AI or machine learning project is ready for production use and ongoing operations.

1) Use-case fit

- [ ] The business problem is clearly defined.
- [ ] The expected outcome is measurable.
- [ ] A simple rules-based or deterministic solution was considered.
- [ ] Machine learning is justified by noise, variability, scale, or the need for prediction/ranking/classification.
- [ ] Failure impact has been assessed for the intended use case.
- [ ] The model will support a decision, not replace required human judgment where risk is high.

2) Data readiness

- [ ] Required data sources have been identified.
- [ ] Data ownership and access are approved.
- [ ] Input data is representative of the production environment.
- [ ] Label quality has been reviewed, where supervised learning is used.
- [ ] Missing values, duplicates, and outliers have been addressed.
- [ ] Schema, feature definitions, and data contracts are documented.
- [ ] Known data limitations are recorded.
- [ ] Sensitive data handling requirements are defined.

3) Baseline and model selection

- [ ] A simple baseline has been established for comparison.
- [ ] Model selection is based on the problem type and operational constraints.
- [ ] The rationale for choosing machine learning over rules is documented.
- [ ] Interpretability requirements have been considered.
- [ ] Resource constraints such as latency, memory, and compute cost are known.

4) Training pipeline controls

- [ ] Training steps are reproducible.
- [ ] Data preprocessing is versioned.
- [ ] Feature engineering logic is documented and repeatable.
- [ ] Training and validation splits are appropriate for the problem.
- [ ] Leakage checks have been performed.
- [ ] Random seeds and dependencies are controlled where applicable.
- [ ] The training environment can be recreated.

5) Evaluation and validation

- [ ] Evaluation metrics are chosen for the real business goal.
- [ ] Accuracy alone is not the only success measure.
- [ ] Precision, recall, false positive rate, calibration, latency, or cost metrics are included as relevant.
- [ ] The model was tested on held-out or otherwise unseen data.
- [ ] Performance is compared against the baseline.
- [ ] Edge cases and failure modes have been reviewed.
- [ ] The model meets minimum thresholds for acceptance.
- [ ] Validation results are reviewed by the appropriate stakeholders.

6) Deployment readiness

- [ ] The deployment pattern is defined: batch, online service, or embedded use.
- [ ] Inference dependencies are known and controlled.
- [ ] Rollout can be limited, staged, or reversed.
- [ ] Rollback steps are documented and tested.
- [ ] Versioning exists for the model, code, and features.
- [ ] Runtime configuration is tracked.
- [ ] The model is deployed behind a control or approval gate if risk is meaningful.

7) Monitoring and drift management

- [ ] Input data quality is monitored in production.
- [ ] Model performance is monitored after deployment.
- [ ] Drift detection is in place for data, concept, or behavior changes.
- [ ] Alert thresholds are defined and actionable.
- [ ] Latency and resource usage are monitored.
- [ ] Business impact is measured, not just technical uptime.
- [ ] Retraining triggers are defined.

- [ ] Manual review or fallback procedures are available if quality drops.

8) Security and governance

- [ ] Access to training data and model artifacts is restricted appropriately.
- [ ] Sensitive features are reviewed for policy or compliance impact.
- [ ] The use case has been assessed for bias or unfair outcomes.
- [ ] Human review requirements are defined for high-impact decisions.
- [ ] Audit logs capture key model actions and changes.
- [ ] Model behavior is explainable enough for the audience that must approve or operate it.
- [ ] The system has an owner responsible for lifecycle management.

9) Operational decision check

Answer these before production approval:

- [ ] Can we define what good looks like in measurable terms?
- [ ] Can we detect when quality is degrading?
- [ ] Can we roll back safely if the model behaves unexpectedly?
- [ ] Are the data inputs stable enough to support continued use?
- [ ] Is the model?s output being used within its intended decision scope?
- [ ] Is the operational cost justified by expected value?

10) Go-live criteria

Proceed to production only if all of the following are true:

- [ ] The use case is appropriate for machine learning.
- [ ] Data quality and labels are sufficient.
- [ ] Evaluation meets agreed thresholds.

- [ ] Deployment is versioned and reversible.
- [ ] Monitoring and drift response are in place.
- [ ] Security and governance requirements are satisfied.
- [ ] Stakeholders understand the model?s limits.

Quick decision summary

Use machine learning when:

- The task involves prediction, ranking, classification, anomaly detection, or forecasting.
- The data is noisy or too variable for fixed rules.
- Ongoing learning or adaptation is valuable.

Use deterministic software when:

- The logic is stable and fully understood.
- Errors would create unacceptable risk.
- You cannot validate the model?s behavior reliably.

Final note

A machine learning system is not just a model. It is a production pipeline with data dependencies, evaluation criteria, operational controls, and lifecycle obligations. Treat it like any other critical system: define success, validate it, monitor it, and be ready to change it safely.