AI Development Workflow — Assignment

Course: AI for Software Engineering | Duration: 7 days

Team: Gervásio Magul — Authors: Gervásio Magul

Date: 2025-11-05

Abstract

This document applies the AI Development Workflow to a practical problem: predicting 30-day hospital readmissions. It covers problem definition, data strategy, model development, evaluation, deployment, ethics, and a reflection. The goal is to demonstrate a complete pipeline suitable for academic submission and practical implementation.

Part 1 — Short Answer Questions

1. Problem Definition

Problem: Predict student dropout within the next academic term for a large online university (example used in Part 1).

Objectives:

- Early identification of at-risk students 4–8 weeks prior to dropout window.
- Minimize false positives to preserve outreach resources.
- Ensure equitable performance across demographic groups.

Stakeholders: Student success team; Students.

KPI: Precision@K (precision among top 5% highest-risk students).

2. Data Collection & Preprocessing

Data sources: (1) LMS logs (page views, submissions, session length). (2) Administrative records (demographics, prior GPA, financial aid).

Potential bias: Selection bias: students with poor internet connectivity may appear disengaged and be misclassified.

Preprocessing steps:

1. Missing value handling: impute and add missingness indicators.
2. Normalization/scaling: robust scaling of continuous features.
3. Temporal aggregation and feature engineering: rolling-window engagement metrics and slopes.

3. Model Development

Model choice: Gradient Boosted Trees (LightGBM/XGBoost) — suitable for tabular data, handles missing values, offers feature importance.

Train/validation/test split: Temporal split to avoid leakage (e.g., train terms 1–6, val term 7, test term 8).

Hyperparameters to tune: Learning rate and max depth / num_leaves.

4. Evaluation & Deployment

Evaluation metrics: AUROC (ranking ability) and Precision/Recall (operational relevance).

Concept drift: Changing relationships over time; monitor performance metrics, input distributions, and calibration. Retrain when drift exceeds thresholds.

Technical challenge: Scalability of low-latency inference; require model optimization, autoscaling, or batching.

Part 2 — Case Study: Hospital Readmission (30-day)

Problem Scope

Problem: Predict whether a discharged patient will be readmitted within 30 days to enable targeted follow-up.

Objectives:

- Reduce preventable readmissions by targeted interventions.
- Prioritize high-risk patients for case management.
- Maintain fairness and privacy.

Stakeholders: Care coordinators, patients/families, clinicians, hospital administration.

Data Strategy

Data sources: EHR (diagnoses, labs, vitals, meds), demographics and social determinants, utilization history, post-discharge follow-up records.

Ethical concerns: (1) Patient privacy and PHI protection. (2) Model bias and unequal performance across groups.

Preprocessing Pipeline (summary)

1. Ingestion & validation: schema checks and timestamp alignment.
2. Missing value strategy: clinically-informed imputation + missingness flags.
3. Temporal aggregation: last-value, mean, slope for vitals and labs during stay.
4. Comorbidity encoding: Charlson index and chronic condition counts.
5. Medication features: counts and high-risk medication flags.
6. Utilization history: prior admissions, ED visits in 6/12 months.
7. Social determinants: discharge disposition, follow-up scheduled (binary flags).
8. Scaling/encoding: robust scaler for numerics; target/frequency encoding for high-cardinality categoricals.

Model Development

Model: LightGBM for primary model; logistic regression as an interpretable baseline.

Hypothetical confusion matrix (N=1000):

	Pred +	**Pred -**
Actual +	TP = 80	FN = 30
Actual -	FP = 20	TN = 870

Precision = 80 / (80+20) = 0.80; Recall = 80 / (80+30) ≈ 0.727; F1 ≈ 0.761.

Deployment

1. Package model & preprocessing pipeline (ONNX/joblib + config).

2. Feature store and data pipeline near EHR (batch or stream transforms).
3. Expose private API (gRPC/REST) integrated with EHR via FHIR for clinician dashboard.
4. Shadow deployment, clinician validation, then phased rollout.
5. Monitoring: performance metrics, data drift, calibration, and audit logs.

HIPAA compliance: BAA with vendors, encryption in transit/at rest, RBAC, de-identification where possible, audit logs, and data retention policies. Clinical validation and governance sign-off required.

Optimization (Overfitting)

Use early stopping with validation, L2 regularization, and temporal cross-validation. Limit tree complexity (max_depth/num_leaves) and perform feature selection.

Part 3 — Critical Thinking

Ethics & Bias

Biased training data may reproduce historic disparities, leading to underprediction of risk for marginalized patients and reduced care. Mitigation: audit metrics by subgroup, reweight or oversample underrepresented groups, remove or adjust proxy features (e.g., zip code), and involve clinicians and community representatives.

Trade-offs

Interpretability vs accuracy: Complex models often improve accuracy but reduce explainability. In healthcare, explanations are vital—use SHAP/local explanations and keep an interpretable baseline for audits.

Limited compute: choose efficient models (logistic regression, small tree ensembles), distill larger models into smaller ones, quantize, or run heavy models offline while using a lightweight triage model online.

Part 4 — Reflection & Workflow Diagram

Reflection

Most challenging: Data acquisition and quality—EHR systems are heterogeneous, labels can be noisy, and linking records is difficult.

Improvements with more time/resources: More annotated data, richer social determinants, prospective pilots, bespoke clinical feature engineering, and fairness audits.

Workflow Diagram (ASCII)

```
Problem Definition
  |
  v
Data Collection & Ingestion
  - EHR, demographics, labs, claims
  |
  v
Data Cleaning & Preprocessing
  - missing handling, normalization, temporal aggregations, feature
  |
  v
Exploratory Data Analysis (EDA)
  - distributions, correlations, bias checks
  |
  v
Model Development
  - baselines, feature selection, hyperparameter tuning
  |
  v
Evaluation & Validation
  - metrics (AUROC, precision@K), cross-val, subgroup fairness check
  |
  v
Deployment Preparation
  - model packaging, API, EHR integration, BAAs, security
  |
  v
Deployment (Shadow -> Pilot -> Production)
  |
  v
Monitoring & Maintenance
  - performance, drift detection, retraining, audit logs
  |
  v
```

```
Stakeholder Feedback & Iteration
```

References

- CRISP-DM framework and standard ML deployment practices.
- Fairness in Machine Learning literature (audits, reweighting).
- HIPAA guidance for healthcare data protection.

Appendix — Code & Repo Structure (provided in GitHub skeleton)

Repository includes commented scripts for preprocessing, model training, evaluation, and a minimal FastAPI serve stub. See README in repository for run instructions and placeholders for sensitive data.

Note: Edit the cover author names and team name before final submission. Do not include PHI in any public repository.