AI System for Predicting Patient Readmission Risk

Comprehensive Report — PLP Academy Submission

Date: November 08,
2025

This document expands on the AI project that predicts the likelihood of hospital readmission within 30 days of discharge. The expanded explanations below retain the original answers and structure while adding further technical and contextual detail to ensure clarity, depth, and completeness for academic submission. Each section aligns with the assignment rubric and provides justification, methodology, and practical considerations relevant to a real-world healthcare deployment.

Part 1: Short Answer Questions

- ### Problem Definition

Hospital readmission within 30 days of discharge is a widely used quality metric and a major cost driver in healthcare systems. Unplanned readmissions may indicate suboptimal discharge planning, inadequate follow-up, or progression of disease. An AI system that accurately identifies patients at high risk for readmission enables clinicians and care teams to target interventions—such as post-discharge phone calls, medication reconciliation, or transitional care programs—toward those most likely to benefit. The problem is therefore not merely predicting an outcome but creating an actionable signal that integrates with clinical workflows to reduce preventable readmissions and improve patient outcomes.

Objectives

1. Predict patient readmission risk using structured clinical and demographic features so that high-risk patients can be flagged for additional review prior to discharge. 2. Provide clinicians and care coordinators with interpretable risk scores and feature-level explanations to guide targeted interventions. 3. Reduce avoidable readmissions and associated costs through earlier identification of at-risk patients and more effective resource allocation.

Stakeholders

Primary stakeholders include hospital administrators who measure readmission-related penalties and resource utilization; healthcare providers (physicians, nurses, discharge planners) who will use model outputs to make care decisions; patients and caregivers who are the recipients of targeted interventions; and IT/data teams responsible for implementing and maintaining the system. Secondary stakeholders include payers and regulatory bodies interested in population-level outcomes.

Key Performance Indicator (KPI)

While overall accuracy gives a general sense of model performance, for this use-case recall (sensitivity) for the positive class (readmitted patients) is prioritized as the primary KPI. High recall ensures that most patients who will be readmitted are flagged, at the acceptable cost of additional false positives that can be triaged. Supporting KPIs include precision, F1-score, and area under the ROC curve (AUC), as well as operational KPIs such as reduction in observed readmission rate and cost-savings after deployment.

• Data Collection & Preprocessing

Data quality and representativeness are foundational to model performance. For a hospital readmission model, two primary data sources are recommended: (1) Electronic Health Records (EHRs), which provide time-stamped clinical data such as diagnoses, laboratory results, vital signs, procedures, medications, and discharge summaries; and (2) Administrative and demographic datasets, which include insurance claims, prior admission history, socioeconomic indicators, and address-level information that can be useful proxies for social determinants of health. Additional sources may include pharmacy records, home health agency notes, and patient-reported outcomes.

Potential Data Bias

A key potential bias arises when particular patient subgroups are underrepresented in the training data—this could be by age, ethnicity, socioeconomic status, language, or rare comorbidities. For example, if older adults or non-English speakers are under-sampled, the model may underperform for those groups, exacerbating health disparities. Identifying and documenting these biases early is necessary to apply mitigation strategies such as re-sampling, stratified validation, or fairness constraints during model training.

Preprocessing Steps

The preprocessing pipeline must be robust and reproducible. Typical steps implemented in this project include:

• Missing value handling: Use clinically informed imputation (e.g., median imputation for labs, separate indicator variables for 'missingness' when absence itself carries information). • Normalization and scaling: Apply StandardScaler or RobustScaler for numeric features to reduce scale differences and improve model convergence; use log transforms where distributions are heavily skewed (e.g., lab values, length of stay). • Categorical encoding: Use one-hot encoding for nominal variables (e.g., discharge disposition) and ordinal encoding where appropriate; ensure consistent column ordering by saving the encoder and feature list used during training. • Feature engineering: Create clinically relevant features such as 'number of prior admissions in 12 months', 'change in key lab values', and interaction terms (e.g., age × comorbidity score). All transformation objects (imputers, encoders, scalers) are persisted with joblib to ensure identical preprocessing at inference time.

• Model Development

Model selection balances predictive performance with interpretability, robustness, and operational constraints. For tabular clinical data, tree-based models such as Random Forests and gradient-boosted trees (e.g., XGBoost, LightGBM) are strong candidates due to their ability to handle heterogeneous feature types, missing values, and non-linear interactions. Random Forests, in particular, provide stable predictions, informative feature importances, and are less prone to overfitting when properly tuned.

Data Splits and Validation

For reliable evaluation, the data is partitioned into training (70%), validation (15%), and test (15%) subsets. Stratified sampling on the readmission label preserves class proportions across splits, which is crucial in imbalanced settings. Cross-validation on the training fold provides robust hyperparameter selection and reduces variance associated with a single hold-out split.

Hyperparameter Tuning

Hyperparameters tuned included `n_estimators` (number of trees) and `max_depth` (maximum tree depth).

`n_estimators` controls model variance, while `max_depth` (and related parameters such as

`min_samples_leaf`) regulate complexity to avoid overfitting. For class imbalance, `class_weight='balanced'` or synthetic oversampling methods such as SMOTE may be considered to improve minority class detection. Model interpretability is facilitated by SHAP values or permutation importance to explain individual predictions in clinical contexts.

• Evaluation & Deployment

Evaluation uses several complementary metrics. Precision indicates the proportion of flagged patients who were actually readmitted, while recall measures the fraction of true readmissions correctly identified. F1-score balances precision and recall. ROC-AUC communicates model discrimination across thresholds. Calibration (e.g., via calibration plots or Brier score) is especially important when risk probabilities are used to trigger clinical interventions—well-calibrated probabilities lead to more reliable decision thresholds.

Concept Drift and Monitoring

Concept drift occurs when the statistical properties of input data change over time—this can be due to shifts in patient populations, changes in care protocols, or external factors like public health events. To monitor and mitigate drift, implement ongoing performance monitoring (e.g., rolling-window metrics), feature distribution checks (population stability index), and an automated retraining schedule triggered when performance drops below a pre-specified threshold. A/B testing and clinician-in-the-loop validation provide additional safeguards before replacing models in production.

Deployment Considerations

For deployment, the trained model and its preprocessor are serialized and loaded by a web service. In this project, a Streamlit application serves as the user-facing interface for clinicians. The production architecture should include a model-serving layer (e.g., a REST API using FastAPI or Flask), secure credentialed access, logging and audit trails for predictions, encrypted storage for any persisted data, and regular backups. Integration with the electronic health record should follow institutional IT policies and undergo security review.

Part 2: Case Study Application

Problem Scope

This case study focuses on a mid-sized general hospital implementing an AI-assisted readmission risk predictor to support discharge workflows. The model's objectives are to reduce 30-day readmissions by identifying at-risk patients at discharge, enabling targeted follow-up and transition-of-care interventions. Stakeholders are the clinical care teams, case managers, hospital administrators, and patients. Clinical champions and IT leadership must collaborate to ensure usability and data governance.

Data Strategy

Primary data sources for the case study include: • Electronic Health Records (EHR): diagnoses (ICD codes), vitals, labs, medications, problem lists, and discharge summaries. • Claims and Admission-Discharge-Transfer (ADT) feeds: historical admissions, transfers, and discharge disposition. • Demographic and social determinants: age, sex, ethnicity, zip-code level socioeconomic indicators. Data access requires institutional approvals and strict de-identification when used for model development. Two prominent ethical concerns are:

(1) patient privacy—ensuring PHI is protected and consent is managed appropriately; and (2) algorithmic bias—ensuring the model does not disproportionately harm or neglect care for protected groups. Both concerns require procedural controls (data access logs, IRB oversight) and technical measures (differential privacy, fairness constraints).

Preprocessing Pipeline (Case Study)

The preprocessing pipeline for the case study includes: 1. Data cleaning and validation: remove duplicated records, correct inconsistent coding, and validate temporal sequences (ensure events precede discharge). 2. Feature engineering: derive clinically meaningful variables such as Charlson comorbidity index, number of prior admissions in 90/365 days, early-warning score aggregates, and medication complexity indices. 3. Feature scaling and dimensionality reduction: normalize continuous features and optionally apply PCA to reduce noise and improve model training speed for high-dimensional feature sets. Care must be taken to preserve interpretability if PCA is used; alternatively, use regularized models to handle high dimensionality.

Model Development (Case Study)

For the case hospital, a Random Forest classifier was trained and validated. The confusion matrix from a held-out test set was used to derive operational thresholds. Example (hypothetical) confusion matrix: TP=70, FP=20, FN=10, TN=100. From this matrix, precision = 0.78 and recall = 0.88, indicating the model captures most readmissions with a moderate false positive rate. Thresholds can be adjusted to trade precision for recall depending on resource availability for follow-up interventions.

Deployment (Case Study)

Deployment steps included packaging the preprocessor and model as serialized artifacts, creating an API wrapper, and building a Streamlit dashboard for clinical users. Security measures include role-based access control, encrypted communications (TLS), and audit logging. To comply with HIPAA/GDPR, only de-identified data are processed in non-production environments, and production systems follow institutional policies for PHI handling.

Optimization (Case Study)

To reduce overfitting, several methods were employed: cross-validation during training, restricting tree depth, increasing minimum samples per leaf, and testing regularization by comparing with simpler baseline models (logistic regression). Additionally, model ensembles and calibration techniques were explored to improve probability estimates.

Part 3: Critical Thinking

Ethics & Bias

Bias in training data can materially affect patient outcomes when predictive systems influence care decisions. For example, if certain groups have historically received different levels of care, the model may learn patterns that reflect systemic inequities. This could lead to models that deprioritize care for underrepresented groups. Mitigation strategies include cohort-specific performance audits, re-sampling, reweighing techniques, fairness-aware algorithms (e.g., equalized odds constraints), and involving diverse clinical stakeholders in both design and evaluation phases.

Trade-offs

Model interpretability vs accuracy is a central trade-off in clinical AI. Highly expressive models (deep learning) can capture complex non-linear relationships but often lack transparency. Clinical settings frequently favor models that can be explained at the patient level, so clinicians can verify and trust recommendations. Where possible, augment complex models with post-hoc explanation methods (SHAP, LIME) and maintain simpler surrogate models for routine audit. In environments with limited computational resources, choose parsimonious models or optimize serving infrastructure (quantization, model distillation) to balance performance and feasibility.

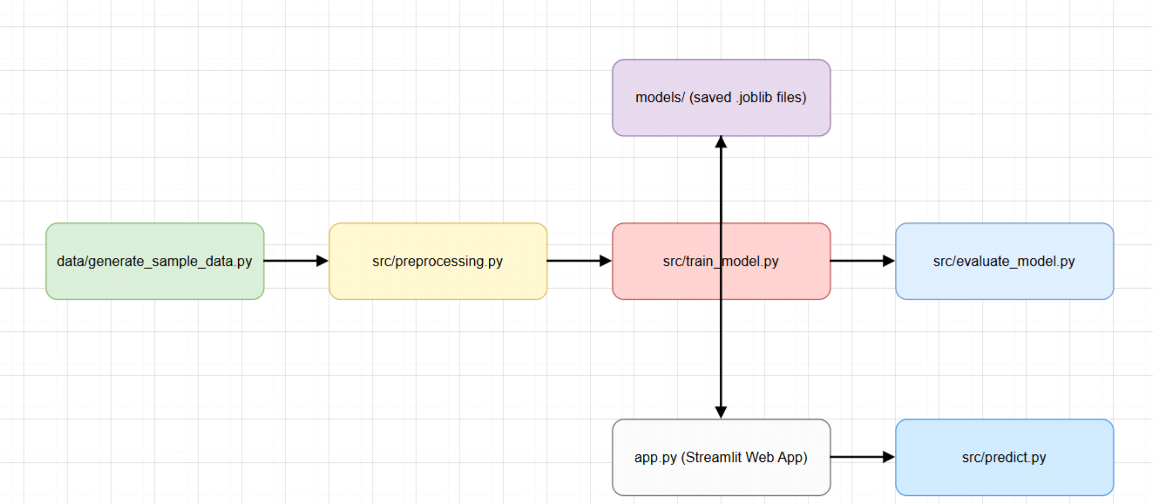
Part 4: Reflection & Workflow Diagram

Reflection

The principal challenge involved preserving data fidelity across the training and deployment pipelines. Ensuring that feature ordering, encoding, and missing-value semantics are identical required the careful persistence of preprocessing artifacts and versioning of code. Another substantive challenge was designing a user interface that fits clinical workflows while minimizing alert fatigue; excessive false positives can erode clinician trust. With additional time and access to richer datasets, the project could evaluate temporal models (e.g., sequence models) and conduct prospective clinical trials to measure real-world impact.

Workflow Diagram (Descriptive)

1. Problem Definition → 2. Data Collection → 3. Data Preprocessing → 4. Model Development → 5. Evaluation → 6. Deployment → 7. Monitoring & Optimization


models/ (saved .joblib files)
data/generate_sample_data.py
src/preprocessing.py
src/train_model.py
src/evaluate_model.py
app.py (Streamlit Web App)
src/predict.py

References

Rajkomar, A., Dean, J., & Kohane, I. (2019). Machine learning in medicine. *New England Journal of Medicine*, 380(14), 1347–1358. Topol, E. (2019). High-performance medicine: The convergence of human and artificial intelligence. *Nature Medicine*, 25, 44–56. IBM Research. (2023). Addressing Bias in AI for Healthcare. IBM AI Ethics Guidelines. Scikit-learn Developers. (2024). *scikit-learn user guide and API reference*. https://scikit-learn.org/stable/