

Part 2: Case Study Application

Hospital Patient Readmission Risk Prediction System

Problem Scope

Problem Definition

Predicting 30-Day Patient Readmission Risk

Develop an AI system to predict which patients are at high risk of being readmitted to the hospital within 30 days of discharge, enabling proactive care management and resource allocation.

Objectives

- Achieve 80%+ accuracy in identifying high-risk patients before discharge
- Reduce overall 30-day readmission rates by 15% through targeted interventions
- Optimize resource allocation for post-discharge care management programs

Stakeholders

- Hospital Administration: Reduces costs and improves quality metrics
- Clinical Care Teams: Enables targeted follow-up care and intervention planning
- Patients & Families: Benefits from improved continuity of care and health outcomes

Data Strategy

Proposed Data Sources

- Electronic Health Records (EHRs): Medical history, diagnoses, medications, lab results, vital signs, length of stay
- Demographics & Social Determinants: Age, gender, insurance status, zip code, socioeconomic indicators
- Previous Admission History: Number of prior admissions, readmission patterns, emergency department visits

Ethical Concerns

1. Patient Privacy & Data Security

Healthcare data contains highly sensitive personal information protected by regulations. Unauthorized access or data breaches could expose patient identities, medical conditions, and treatment histories. Must implement strict access controls, encryption, and anonymization techniques while maintaining data utility for accurate predictions.

2. Algorithmic Bias & Health Equity

Models trained on historical data may perpetuate existing healthcare disparities. Certain demographic groups (minorities, low-income patients) may be over-predicted as high-risk due to systemic access barriers rather than true medical risk, leading to stigmatization or denial of care. Must ensure fair representation and regularly audit for discriminatory patterns.

Preprocessing Pipeline

- Data Cleaning: Remove duplicate records, handle missing values using domain-appropriate imputation (clinical defaults for vitals, mode for categorical variables)
- Outlier Detection: Identify and handle anomalous values in lab results and vital signs using clinical reference ranges
- Feature Engineering:
 - Create derived features: number of comorbidities, medication count, previous readmission rate
 - Time-based features: days since last admission, seasonal patterns
 - Risk scores: calculate Charlson Comorbidity Index, LACE score components
- Normalization: Standardize continuous variables (lab values, vital signs) to ensure balanced feature contribution
- Encoding: Convert categorical variables (diagnosis codes, medication names) using appropriate encoding techniques

Model Development

Model Selection & Justification

Gradient Boosting Classifier (XGBoost)

Justification: XGBoost is optimal for this healthcare application because:

- Excels with tabular healthcare data containing mixed feature types
- Handles imbalanced datasets well (readmissions are typically 10-20% of cases)
- Provides feature importance for clinical interpretability
- Robust to missing values and outliers common in EHR data
- High predictive accuracy while maintaining computational efficiency

Confusion Matrix & Metrics (Hypothetical Data)

Confusion Matrix (n=1000 test patients):

Predicted: No Readmit

Predicted: Readmit

Actual: No Readmit

720 (TN)

80 (FP)

Actual: Readmit

30 (FN)

170 (TP)

Precision: $TP / (TP + FP) = 170 / (170 + 80) = 170 / 250 = 0.68$ or 68%

→ Of all patients predicted to be readmitted, 68% actually were readmitted

Recall (Sensitivity): $TP / (TP + FN) = 170 / (170 + 30) = 170 / 200 = 0.85$ or 85%

→ Of all patients who were actually readmitted, we correctly identified 85%

High recall is prioritized in this healthcare context to minimize missing at-risk patients (false negatives), even if it means some false alarms (lower precision).

Deployment

Integration Steps

- Step 1 - API Development: Create RESTful API endpoint that accepts patient data and returns readmission risk score (0-100) and risk category (Low/Medium/High)
- Step 2 - EHR Integration: Develop HL7/FHIR-compliant interface to automatically pull required patient data from hospital's EHR system at discharge
- Step 3 - Clinical Workflow Integration: Embed risk scores into discharge workflow dashboard, triggering alerts for high-risk patients to care coordination team
- Step 4 - Pilot Testing: Deploy to single department for 3-month validation period with parallel manual review before hospital-wide rollout
- Step 5 - Monitoring Dashboard: Implement real-time performance tracking showing daily predictions, actual outcomes, and model accuracy metrics

HIPAA Compliance Strategy

Key Compliance Measures:

- Data Encryption: Implement end-to-end encryption for data in transit (TLS 1.3) and at rest (AES-256), ensuring all PHI is protected during transmission and storage
- Access Controls: Role-based access with multi-factor authentication, audit logging of all data access, automatic session timeouts
- Data Minimization: Only collect and process minimum necessary PHI for prediction; implement automatic de-identification where possible
- Business Associate Agreements: Establish BAAs with all third-party vendors, cloud service providers involved in data processing
- Regular Security Audits: Conduct quarterly risk assessments, penetration testing, and HIPAA compliance reviews with documentation
- Breach Response Plan: Maintain documented incident response procedures with notification protocols meeting HIPAA's 60-day requirement

Optimization

Addressing Overfitting

Method: Cross-Validation with Early Stopping

Approach: Implement k-fold cross-validation ($k=5$) combined with early stopping during model training:

- Cross-Validation: Split training data into 5 folds, train on 4 folds and validate on the 5th, rotating through all combinations. This ensures model generalizes across different data subsets rather than memorizing specific training examples.
- Early Stopping: Monitor validation loss during training and stop when it stops improving for 10 consecutive iterations, preventing the model from continuing to optimize on training data after validation performance plateaus.

- Additional Regularization: Apply L2 regularization to XGBoost parameters (lambda, alpha) to penalize complex models and reduce feature weight magnitudes.

This combination prevents overfitting while maintaining strong predictive performance on unseen patient data, crucial for reliable clinical deployment.