Title: AI Development Workflow – Parts 2 to 4

Part 2: Case Study – Patient Readmission Risk Prediction

1. Problem Scope

Problem Statement: Develop an AI model that predicts the likelihood of a patient being readmitted to the hospital within 30 days of discharge.

Objectives:

- Identify high-risk patients at discharge.
- Prevent unnecessary readmissions through targeted interventions.
- Improve patient care and reduce costs for hospitals.

Stakeholders:

- Patients
- Clinicians (Doctors & Nurses)
- Hospital Administrators

2. Data Strategy

Proposed Data Sources:

- *Electronic Health Records (EHRs):* Diagnoses, lab results, medications, discharge summaries.
- *Demographic Data:* Age, gender, socioeconomic status, insurance coverage.

Ethical Concerns:

1. **Privacy Risk:** EHR data contains personally identifiable information.
2. **Bias from Historical Disparities:** Care practices may differ across demographics, skewing predictions.

Preprocessing Pipeline:

- Handle missing values in lab results and discharge documentation.
- Encode categorical features (e.g. diagnosis codes, comorbidity categories).
- Engineer features like:
  - Number of prior admissions
  - Time since last discharge
  - Risk scores from existing clinical scales
- Standardize numerical values such as lab metrics.

3. Model Development

Model Selection:

- **Gradient Boosted Trees (XGBoost):** Offers strong predictive power on medical tabular data and handles imbalanced classes well. Interpretability via SHAP values is a bonus in clinical contexts.

Hypothetical Confusion Matrix:

	Predicted Readmit	**Predicted No Readmit**
Actual Readmit	90	30
Actual No Readmit	40	140

Precision: 90 / (90 + 40) = **0.692** **Recall:** 90 / (90 + 30) = **0.75**

4. Deployment Strategy

Steps to Integrate Model:

1. Export trained model as an API (Flask/FastAPI).
2. Deploy within the hospital's EHR infrastructure.
3. Display readmission risk scores during discharge processing.

Compliance & Regulation:

- Encrypt all data in transit and at rest.
- Anonymize training data and limit feature visibility based on user roles.
- Conduct bias audits and regular security assessments.
- Follow principles from **HIPAA** or local healthcare data protection laws.

5. Optimization

How to Address Overfitting:

- Use **cross-validation** during model training.
- Apply regularization techniques (e.g., L1 or L2 penalties).
- Prune trees in XGBoost and limit maximum depth.
- Incorporate more diverse and real-time data during model updates.

Part 3: Critical Thinking

1. Ethics & Bias (10 pts)

How might biased training data affect patient outcomes? If the training data reflects historical healthcare disparities (e.g., fewer diagnoses or under-treatment of certain demographic groups), the model may underpredict readmission risk for those groups. This could lead to inequitable care—patients most in need of follow-up could be overlooked, reinforcing systemic bias.

Strategy to Mitigate Bias: Use **fairness-aware training** techniques such as reweighing underrepresented groups during model training, and regularly audit outcomes using tools like **Fairlearn** or **Aequitas**. Additionally, include diverse, representative data and consult clinicians for interpretability checks.

2. Trade-offs (10 pts)

Interpretability vs. Accuracy in Healthcare:

- High-performing models like deep neural networks may offer strong accuracy but are often "black boxes."
- In contrast, interpretable models like logistic regression or decision trees may be less accurate but allow clinicians to **understand and trust predictions**, which is essential for life-and-death decisions.

Limited Computational Resources – Model Choice Impact: If the hospital has constrained infrastructure:

- Avoid heavy models like deep neural nets.
- Prefer **lightweight, interpretable models** such as Logistic Regression, Decision Trees, or Gradient Boosted Trees with capped depth.
- Consider **on-device inference** and model quantization to reduce memory and processing needs.

Part 4: Reflection & Diagram

◆ 1. Most Challenging Aspect

Data Preprocessing was the most challenging part of the workflow. Balancing data quality, bias mitigation, and privacy—especially in sensitive domains like education and healthcare—required careful handling of missing values, normalization, and ethical considerations around fairness.

◆ 2. What Would You Do Differently?

Given more time or resources, I would:

- Experiment with **ensemble models** (e.g., stacking XGBoost with logistic regression for added interpretability).
- Integrate **longitudinal data** like behavior trends over time, not just snapshots.
- Incorporate **explainability tools** (e.g., SHAP) to better communicate model decisions to stakeholders.

◆ 3. Workflow Diagram

Here's a basic AI development workflow that reflects your process. This can be added as a flowchart in your report:

```
[Problem Definition]
        ↓
[Data Collection & Preprocessing]
        ↓
[Model Selection & Training]
        ↓
[Evaluation & Error Analysis]
        ↓
[Deployment & Monitoring]
        ↺ (Loop back to handle concept
```