PART 1: SHORT ANSWER QUESTIONS

Problem Definition

Predicting student dropout rates in online universities.

Objectives:

1. Identify at-risk students.
2. Improve student retention through interventions.
3. Optimize resource allocation.

Stakeholders:

1. Students
2. University administrators

KPI:

- Accuracy ≥ 85%

Data Collection & Preprocessing

Data Sources:

- Student demographics
- LMS activity logs

Bias Risk:

- Students in low-connectivity areas may appear inactive and be flagged unfairly.

Preprocessing Steps:

- Impute missing fields (age, GPA)
- Normalize login/activity data
- One-hot encode fields like degree program

Model Development

Random Forest: Interpretable, robust, handles categorical data.

Data Splitting:

- 70% Training, 15% Validation, 15% Testing

Hyperparameters to Tune:

- n_estimators – trees in the forest
- max_depth – prevents overfitting

Evaluation & Deployment

Evaluation Metrics:

- F1 Score (handles class imbalance)
- ROC-AUC (separates signal from noise)

Concept Drift: Change in user behaviour over time

- Use rolling window retraining

Deployment Challenge:

- Scaling predictions for thousands of students in real-time

PART 2: CASE STUDY — HOSPITAL READMISSION

1. Problem Scope

- **Problem: Predict patient readmission risk within 30 days.**
- **Objectives:**
    1. **Identify at-risk patients early**
    2. **Reduce unnecessary readmissions**
    3. **Improve care outcomes**
- **Stakeholders:**
    - **Doctors, hospital administrators, patients**

2. Data Strategy

- **Data Sources:**
    - **Electronic Health Records (EHR)**
    - **Demographics & social determinants (age, income, ZIP code)**
- **Ethical Concerns:**
    - **Patient privacy (HIPAA)**
    - **Risk of underrepresentation bias**
- **Preprocessing Pipeline:**
    - **Handle missing data (fill with median or drop)**
    - **Feature engineering:**

        - **Num. of prior admissions**
        - **Days in hospital**
        - **Chronic conditions count**
    - **One-hot encode gender**
    - **Normalize continuous values (e.g., blood pressure)**

3. Model Development

- **Model: XGBoost (handles tabular medical data well)**
- **Confusion Matrix (hypothetical):**

	Pred Yes	**Pred No**
Actual Yes	**70**	**30**
Actual No	**20**	**80**

- **Precision = 70 / (70 + 20) = 0.78**
- **Recall = 70 / (70 + 30) = 0.70**

---

4. Deployment

- **Steps:**
    1. **Save model using joblib or pickle**
    2. **Create API with FastAPI**
    3. **Deploy behind secure hospital infrastructure**
- **HIPAA Compliance:**
    - **Encrypt all communications (TLS)**
    - **Log access and monitor usage**
    - **Use compliant cloud services (AWS HealthLake, Azure HIPAA)**

---

5. Optimization

- **Overfitting Solution:**
    - **Use K-Fold Cross Validation (e.g., 5-fold) to generalize performance**

PART 3: CRITICAL THINKING

1. Ethics & Bias

- **Risk: Training data underrepresents patients with rare conditions or from certain demographics.**
- **Effect: May lead to inaccurate or unfair predictions for minorities.**
- **Strategy: Use reweighting or adversarial debiasing, expand dataset diversity.**

---

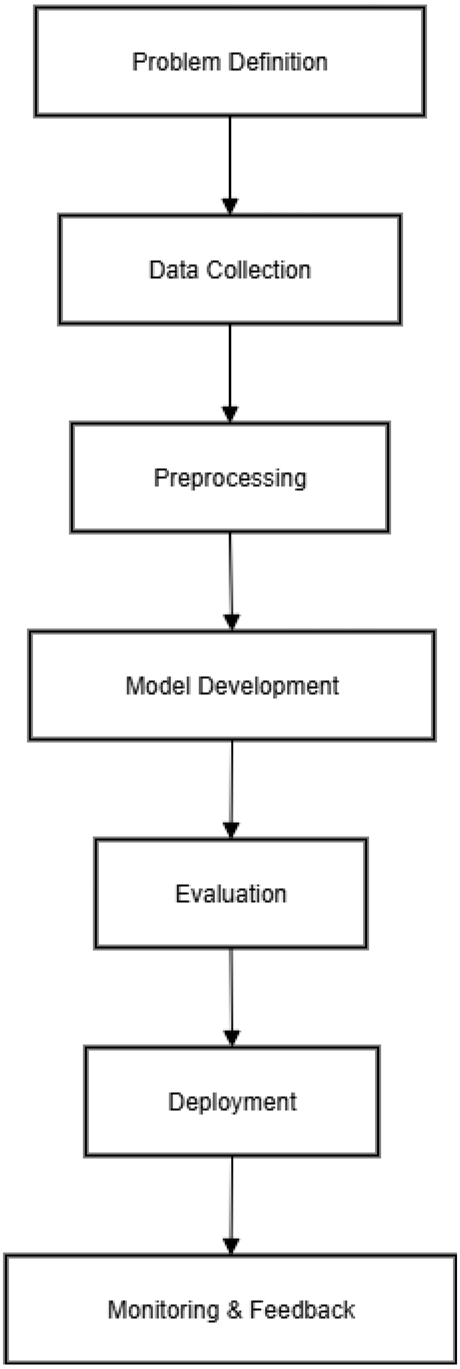
2. Trade-offs

- **Interpretability vs Accuracy:**
    - **XGBoost is accurate but less interpretable**
    - **In healthcare, simpler models like Logistic Regression might be preferred**
- **Limited Resources:**
    - **Use smaller models (e.g., Decision Trees)**
    - **Avoid complex ensembles on low-power devices**

PART 4: REFLECTION & DIAGRAM

1. Reflection

- **Most challenging: Data inconsistencies in EHRs**
- **Improvement: More time for data cleaning + inclusion of unstructured data (e.g., clinical notes)**


Problem Definition
Data Collection
Preprocessing
Model Development
Evaluation
Deployment
Monitoring & Feedback