

## Part 1: Short Answer Questions (30 points)

### 1. Problem Definition

*Hypothetical AI problem: Predicting Student Dropout Rates*

#### Description

Develop an AI system that predicts the likelihood of students dropping out based on academic performance, attendance records, and socio-economic factors. This enables early interventions to improve retention.

#### Objectives

- ❖ Identify students at high risk of dropping out early in the academic year.
- ❖ Provide actionable insights to tutors and administrators for targeted interventions.
- ❖ Reduce overall dropout rates by at least 20% via data driven decision making.

#### Stakeholders.

- a) Teachers/Tutors/Counselors – Receive alerts about at-risk students and provide timely academic or emotional support.
- b) School Administrators – Use predictions to implement retention strategies and allocate resources effectively.

#### Key Performance Indicator (KPI)

Model Accuracy / F1 Score – Measures how effectively the AI system correctly identifies at risk students while minimizing false alarms.

### 2. Data Collection & Preprocessing

#### Data Sources

School Information System (SIS): Contains student attendance, grades, disciplinary records, and demographic details.

Socio – Economic Surveys: Data from government bodies like the Economic Survey, KDHS etc. or student background forms showing family income, home environment, and parental education.

#### Potential Bias

Socio-economic bias: Students from low-income households might be overrepresented in the “at-risk” category, leading the model to unfairly associate poverty with dropout likelihood rather than other underlying causes like health issues or school quality.

#### Preprocessing Steps

- a) Normalization: Scale numerical features e.g. attendance rates, grades to a common range (0-1) for consistent model training.
- b) Handling Missing Data: Fill missing attendance or grade values using mean/median imputation or remove records with excessive gaps.
- c) Encoding Categorical Variables: Convert non-numeric features like parental education, region, gender into numeric form using one-hot label encoding.

### 3. Model Development

#### Model Choice – Random Forest

I'd choose Random Forest classifier for this dropout-prediction task because:

- ❖ It performs very well on tabular data with mixed feature types (categorical + numerical)
- ❖ It gives feature importance measures that can help show which factors drive dropout risk – useful for stakeholders (teachers, admins).
- ❖ It naturally handles non-linear interactions and is robust to noisy features.
- ❖ It requires relatively little feature scaling and preprocessing compared with neural nets for example.
- ❖ It's less likely to overfit than single decision trees and is straightforward to train and deploy.

#### Data split strategy

- Training set: 60% - used to fit the model.
- Validation set: 20% - used for hyperparameter tuning and early stopping decisions.
- Test set: 20% - held out and used only once for the final performance estimate.

*Implementation details/best practices:*

- Use stratified splitting on the dropout label to keep class proportions consistent across splits - important if dropouts are a minority.
- If the data has a time component (e.g., sequential school years), use a temporal split: train on earlier years and validate/test on later years to avoid leakage.
- During tuning, prefer k-fold cross-validation (stratified) on the training validation portion (or use nested CV) to get reliable hyperparameter selection if dataset size allows.

#### Hyperparameters to tune (and why)

- ❖  $n\_estimators$  (number of trees): Controls ensemble size. More trees usually improve performance and stability but increase computing time. Tune to find the sweet spot where additional trees give diminishing returns.

- ❖ *max\_depth* (maximum depth of each tree): Controls model complexity. Shallow trees may underfit (miss complex patterns); very deep trees may overfit. Tuning *max\_depth* helps balance bias vs variance and improves generalization.

#### 4. Evaluation & Deployment

##### Evaluation metrics

- a) F1 Score - Balances precision and recall, making it suitable when dropout cases are rare. It ensures the model performs well at identifying actual at-risk students without over-alerting teachers with false positives.
- b) Area under the Receiver Operating Characteristic Curve (ROC -AUC) - Measures the model's ability to distinguish between "dropout" and "non-dropout" cases across thresholds. A higher AUC means better discrimination between classes.

##### Concept Drift

*Definition:* Concept drift happens when the statistical relationships between input features and the target (dropout risk) change over time. For example, grading systems, new policies, or economic shifts may alter dropout patterns.

##### Monitoring

- ❖ Continuously track model performance metrics (e.g., F1 score) on new incoming data.
- ❖ Use data drift detection tools (like Kolmogorov–Smirnov tests) to compare new data distributions against the training data.
- ❖ Periodically retrain the model using the latest labeled data to maintain accuracy.

##### Technical Challenge – Scalability

Deploying models across multiple education systems or schools can be challenging because:

- ❖ Large data volumes require efficient batch processing or distributed inference.
- ❖ Different schools may have varying data formats, so integrating heterogeneous data sources at scale needs strong data standardization pipelines before model inference.

## Part 2: Case Study Application (40 points)

### 1. Problem Scope: Problem Definition

The hospital wants to develop an AI-based predictive system that estimates the likelihood of a patient being readmitted within 30 days after discharge. This system aims to support healthcare

staff in identifying high-risk patients early and implementing preventive care plans to reduce avoidable readmissions.

## Objectives

1. Predict readmission risk for each discharged patient using clinical and demographic data.
2. Assist medical staff in making informed decisions about post-discharge follow-up and patient monitoring.
3. Reduce hospital readmission rates and associated healthcare costs while improving patient outcomes.

## Stakeholders

- a) Hospital Management: Uses predictions to improve efficiency, allocate resources, and meet quality benchmarks.
- b) Doctors and Nurses: Receive alerts for high-risk patients, enabling early intervention and better discharge planning.
- c) Patients: Benefit from targeted care plans and reduced likelihood of preventable readmissions.

## 2. Data Strategy: Data Sources (Proposed)

- a) EHRs (Electronic Health Records) – diagnoses, problem list, procedures, medications, vital signs, length of stay, lab results, prior admissions, discharge disposition.
- b) Outpatient / Follow-up Data - scheduled follow-ups, primary care visits, post-discharge appointments, home health referrals.
- c) Demographics & Social Determinants - age, sex, ethnicity, ZIP code, living situation, employment status, family support, socio-economic indicators (can be linked via ZIP-level census data).
- d) Clinical Notes (unstructured) - discharge summaries, progress notes, nursing notes, reason for admission (for NLP-derived signals).
- e) Administrative / Billing Data - admission/discharge timestamps, insurance type, codes (ICD-10), readmission flags.
- f) Device / Remote Monitoring (if available) - remote vitals or device alerts after discharge.

## Ethical Concerns

- ❖ *Bias & fairness*: The model could learn spurious correlations (e.g., linking insurance type, race, or ZIP code to higher risk) and thus propagate inequities (e.g., undertreating certain groups). Mitigation requires bias audits, fairness-aware metrics, and stakeholder review before using predictions to influence care.

- ❖ *Patient privacy & data security*: EHRs and clinical notes contain highly sensitive personal health information. The project must ensure de-identification where possible, strong access controls, encryption at-rest/in-transit, audit logs, and compliance with applicable laws/regulations (e.g., HIPAA, GDPR-like rules, local laws).

### **Preprocessing pipelines (end to end including feature engineering)**

- a) Data ingestion & governance
- b) Label creation
- c) De-identification & privacy-preserving transforms
- d) Data cleaning
- e) Temporal aggregation / windowing
- f) Feature engineering (key examples)
- g) Encoding & scaling
- h) Imbalance handling
- i) Feature selection & dimensionality reduction
- j) Data splits (to avoid leakage)
- k) Validation & sanity checks
- l) Pipeline automation & reproducibility

## **3. Model Development**

### **Selected Model**

Gradient Boosting Machine (GBM) – specifically XGBoost or LightGBM

#### *Justification*

- ❖ *High accuracy*: Often outperforms traditional models like logistic regression for structured healthcare data.
- ❖ *Robust to missing data*: Can handle nulls internally, reducing preprocessing complexity.
- ❖ *Feature importance & interpretability*: Produces interpretable rankings of clinical risk factors (useful for doctors and hospital administrators).
- ❖ *Handles mixed data types well*: Works with both numerical (e.g., lab results) and categorical (e.g., discharge type) features.
- ❖ *Captures complex relationships*: Can model nonlinear interactions among medical, demographic, and behavioral variables.

### **Confusion Matrix**

	Predicted: Readmit (Yes)	Predicted: No Readmit
Actual: Readmit (Yes)	80 (True Positives – TP)	20 (False Negatives – FN)

Actual: No Readmit	30 (False Positives – FP)	170 (True Negatives – TN)
--------------------	---------------------------	---------------------------

## Calculations

Precision (Positive Predictive Value):

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) = 80 / (80 + 30) = 80 / 110 = 0.727 \approx 72.7\%$$

Recall (Sensitivity):

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) = 80 / (80 + 20) = 80 / 100 = 0.80 \approx 80\%$$

## Interpretation

- The precision (72.7%) means that when the model predicts a patient will be readmitted, it's correct about 73% of the time.
- The recall (80%) indicates that the model successfully identifies 8 out of every 10 patients who are truly at risk of readmission.
- In healthcare, recall is often prioritized because missing high-risk patients (false negatives) can lead to worse outcomes and higher costs.

## 4. Deployment: Steps to integrate the Model into the Hospital System

a). Model Packaging & API Development:

- ❖ Convert the trained AI model into a deployable format (e.g., .pkl file or containerized in Docker).
- ❖ Expose it via a RESTful API or microservice that can receive patient data and return readmission risk scores.

b). Integration with Hospital Information Systems (HIS/EHR):

- ❖ Connect the API to the Electronic Health Record (EHR) system.
- ❖ Automate daily scoring of discharged patients using real-time data feeds and store model outputs (e.g., readmission probability, risk level) in each patient's record for clinician review.

c). Clinical Dashboard & Alerts

- ❖ Develop a secure dashboard for doctors and nurses to view patient risk levels and implement alert notifications for high-risk patients before discharge to trigger follow-up planning or counseling.

#### d). Monitoring & Maintenance

- ❖ Track key metrics (e.g., F1 score, recall) in production to detect concept drift or data quality issues.
- ❖ Set up automatic retraining every 3–6 months using the latest hospital data and collect clinician feedback on prediction accuracy and usefulness to continuously refine the system.

#### e). Security & Access Control

- ❖ Deploy within the hospital's secure cloud or on-premises environment.
- ❖ Enforce role-based access—only authorized staff (e.g., clinicians, data scientists) can view or modify model outputs.

#### Compliance Assurance with Healthcare Regulations (e.g., HIPAA)

- Data Privacy & Protection
- Access Control & Authentication
- Governance & Documentation
- Model Transparency & Explainability
- Regular Security & Compliance Reviews

### 5. Optimization: Method to address overfitting

Regularization – Adds a penalty term to the model's loss function to discourage overly complex models that fit noise in the training data.

Example:

In XGBoost/LightGBM, apply L2 (Ridge) regularization using parameter `lambda` using `alpha`. This forces the model to keep weights smaller, improving its ability to generalize to unseen patient data.

### Part 3: Critical Thinking (20 points)

#### **Ethics & Bias: How Biased training data might affect patient outcomes**

If the training data used to build the readmission prediction model is biased, it could lead to unfair or harmful outcomes for certain patient groups. For example:

- ❖ The model might underpredict the risk of readmission if the historical data underrepresents certain populations (e.g., rural patients, low-income groups, women, or elderly patients)

- ❖ This could result in unequal care, where some high-risk patients fail to receive necessary follow-up, increasing preventable readmissions and worsening health outcomes.
- ❖ Conversely, overpredicting risk for certain demographics could cause unnecessary interventions, strain hospital resources and reducing trust in the AI system.

### **Mitigating Bias Strategy**

Perform Fairness Audits and Rebalancing

Assessing model performance regularly across demographic subgroups.

If disparities are found, apply techniques like re-sampling, re-weighting, or fairness constraints during training to ensure equitable predictions.

### **Trade Off: Interpretability vs. Accuracy in Healthcare:**

- High-accuracy models (like deep neural networks or gradient boosting) often act as “black boxes,” making it difficult for clinicians to understand *why* a prediction was made.
- Interpretable models (like logistic regression or decision trees) are easier to explain but may have lower predictive accuracy on complex datasets.
- **Trade off:** In healthcare, interpretability is often prioritized over marginal gains in accuracy because doctors must understand and trust model outputs to make safe, ethical decisions. A slightly less accurate but explainable model is often preferred if it ensures transparency and accountability.

### **Impact of Limited Computational Resources on Model Choice**

- ❖ If the hospital has limited computational power or storage, complex models (e.g., deep neural networks or ensemble methods with hundreds of trees) may be impractical.
- ❖ The hospital might opt for simpler, lightweight models such as logistic regression or smaller decision trees, which:
  - Train faster and require less hardware.
  - Are easier to deploy on existing IT systems.
  - Still provide interpretable and actionable insights for clinical staff.

## **Part 4: Reflection & Workflow Diagram (10 points)**

### **Reflection: Most Challenging Part**

The most challenging segment was data preprocessing and bias management, here is why:

- Ensuring data privacy while preserving model utility is difficult and addressing bias in demographic or socioeconomic variables requires careful analysis and validation.
- Hospital data (EHRs) are often incomplete, inconsistent, and highly sensitive, making cleaning and standardization complex.

### Improvement with Resources/Time

With additional resources and time, I would:

- ❖ Expand data sources — include outpatient visits, home care data, and patient-reported outcomes for richer modeling.
- ❖ Collaborate with clinicians — for domain-informed feature engineering and validation to ensure the model's recommendations are clinically sound.
- ❖ Implement automated MLOps pipelines — for continuous data integration, retraining, and performance monitoring.
- ❖ Use explainable AI (XAI) tools — such as SHAP or LIME to improve transparency and trust among healthcare professionals.

### Diagram: AI Development Workflow - Patient Readmission prediction

