Part 1: Short Answer Questions (30 points)

1. Problem Definition (6 points)

Hypothetical AI Problem:
Predicting student dropout rates in universities.

- **Objectives:**
  1. Identify students at risk of dropping out early in the semester.
  2. Provide timely interventions to improve retention.
  3. Support university planning and resource allocation.
- **Stakeholders:**
  1. University administration.
  2. At-risk students.
- **KPI:**
  Dropout Prediction Accuracy (%) — the percentage of correctly predicted dropouts.

---

2. Data Collection & Preprocessing (8 points)

- **Two Data Sources:**
  1. Student academic records (grades, attendance).
  2. Learning Management System (LMS) activity logs.
- **Potential Bias:**
  Socioeconomic status may be underrepresented or misrepresented, leading to biased predictions favoring well-off students.
- **Three Preprocessing Steps:**
  1. Handle missing values in attendance or grade columns using imputation.
  2. Normalize numerical features like GPA and hours studied.
  3. Encode categorical variables like course type and student major.

---

3. Model Development (8 points)

- **Model Chosen:**
  Random Forest — handles both numerical and categorical data well, is robust to outliers, and reduces overfitting through ensemble learning.
- **Data Splitting:**
  - 70% for training
  - 15% for validation
  - 15% for testing
- **Two Hyperparameters to Tune:**
  1. `n_estimators` — Number of trees; affects accuracy and stability.

2. `max_depth` — Controls tree complexity to prevent overfitting.

---

4. Evaluation & Deployment (8 points)

- **Evaluation Metrics:**
  1. *Precision* — ensures the system doesn't flag students who aren't truly at risk.
  2. *Recall* — important to identify as many at-risk students as possible.
- **Concept Drift:**
  Changes in student behavior or curriculum over time that may reduce model accuracy.
  - *Monitoring:* Regularly retrain the model on fresh data; use performance tracking dashboards.
- **Deployment Challenge:**
  Scalability — Integrating and maintaining the system across different departments and campuses.

Part 2: Case Study Application (40 points)

Problem Scope (5 points)

- **Problem:**
  Predict if a patient will be readmitted within 30 days after discharge.
- **Objectives:**
  1. Reduce avoidable readmissions.
  2. Optimize post-discharge care plans.
- **Stakeholders:**
  1. Hospital administrators.
  2. Patients and their families.

Data Strategy (10 points)

- **Data Sources:**
  1. Electronic Health Records (EHRs): diagnoses, medications, prior admissions.
  2. Demographics: age, gender, zip code, insurance type.
- **Two Ethical Concerns:**
  1. *Patient privacy* — sensitive health data must be protected.
  2. *Bias* — certain groups (e.g., minorities) might be unfairly flagged due to historical disparities.
- **Preprocessing Pipeline:**

1. **Missing Data Handling:** Fill gaps in medication/demographic data using median/mode.
2. **Feature Engineering:**
    - Time since last admission
    - Number of chronic conditions
3. **Encoding/Scaling:** One-hot encode categorical features; normalize numeric ones like lab results.

Model Development (10 points)

- **Model Chosen:**
  Gradient Boosting Machine (e.g., XGBoost) — Excellent at handling imbalanced classification problems and tabular healthcare data.
- **Confusion Matrix (Hypothetical):**

	Predicted Yes	**Predicted No**
Actual Yes	80	20
Actual No	30	170

- **Precision:**
  = 80 / (80 + 30) = **0.727** (72.7%)
- **Recall:**
  = 80 / (80 + 20) = **0.80** (80%)

Deployment (10 points)

- **Integration Steps:**
    1. Develop an API to serve the model.
    2. Connect API with hospital's EHR system.
    3. Display prediction results in the doctor's dashboard.
- **HIPAA Compliance:**
    - Use encryption for data transmission.
    - Apply role-based access controls.
    - Maintain audit logs for model access and usage.

Optimization (5 points)

- **Method to Address Overfitting:**
  Use *cross-validation* and add *regularization* (L1/L2 penalties) to prevent the model from learning noise in training data.

Part 3: Critical Thinking (20 points)

Ethics & Bias (10 points)

- **Impact of Bias:**
  Biased training data may lead to higher readmission predictions for certain races or income groups, leading to discrimination in care.
- **Bias Mitigation Strategy:**
  Use *fairness-aware learning*, e.g., reweighing training samples or introducing constraints that enforce equal opportunity during training.

Trade-offs (10 points)

- **Interpretability vs Accuracy:**
  - Complex models (e.g., XGBoost) offer high accuracy but are harder to explain.
  - Simpler models (e.g., Decision Trees or Logistic Regression) are more interpretable but might underperform.
  - In healthcare, *interpretability* is often prioritized to maintain clinician trust and meet regulatory standards.
- **Limited Resources Impact:**
  - May need to choose lighter models (e.g., Logistic Regression) that consume less CPU/GPU.
  - Cloud deployment may be restricted; on-premise deployment becomes necessary.

Part 4: Reflection & Workflow Diagram (10 points)

Reflection (5 points)

- **Most Challenging Part:**
  Preprocessing medical data — due to complexity and irregular formats.
- **Improvement with More Time/Resources:**
  Would gather more labeled data and include domain experts (e.g., doctors) during feature selection.

Diagram (5 points)

AI Development Workflow Flowchart:

```
plaintext
CopyEdit
         +--------------------+
         |  Problem Definition |
         +--------------------+
                   ↓
         +--------------------+
         | Data Collection     |
         +--------------------+
                   ↓
         +--------------------+
         | Data Preprocessing |
         +--------------------+
                   ↓
         +--------------------+
         | Model Development  |
         +--------------------+
                   ↓
         +--------------------+
         | Model Evaluation   |
         +--------------------+
                   ↓
         +--------------------+
         | Deployment         |
         +--------------------+
                   ↓
         +--------------------+
         | Monitoring & Update|
         +--------------------+
```