Part 1: Short Answer Questions (30 points)

1. Problem Definition (6 points)

- Define a hypothetical AI problem (e.g., "Predicting student dropout rates").

Title: Predicting University Student Dropout Rates

Objectives

1. Accurately identify students at risk of dropping out before the end of the semester.
2. Provide actionable insights for academic advisors to intervene early.
3. Minimize false positives to avoid unnecessary interventions.

Stakeholders

1. **University Administration** – for strategic planning and resource allocation.
2. **Students** – to ensure academic success and support.

Key Performance Indicator (KPI)

- **Recall** (True Positive Rate): Measures how many actual dropouts were correctly identified by the model.

2. Data Collection & Preprocessing (8 points)

Two Data Sources

1. **Student Information Systems (SIS)**: Includes academic records, course enrollment, attendance, grades, and registration data.
2. **Learning Management Systems (LMS)**: Provides behavioral data such as login frequency, assignment submissions, and discussion participation.

One Potential Bias in the Data

- **Socioeconomic Bias**: Students from low-income backgrounds may have limited access to stable internet or devices, skewing LMS activity data and unfairly labeling them as at-risk.

Three Preprocessing Steps

1. **Handling Missing Data**: Use imputation (e.g., median for numerical features, mode for categorical ones) or remove records with excessive missing values.

2. **Normalization**: Apply Min-Max scaling or Z-score standardization to features like attendance rates and GPA for consistent model input.
3. **Encoding Categorical Variables**: Convert fields such as course types or program names using one-hot encoding or label encoding.

3. Model Development (8 points)

Model Choice & Justification

- **Model**: **Random Forest Classifier**
- **Justification**: It handles both numerical and categorical data well, is robust to overfitting due to ensemble learning, and provides feature importance—which helps in explaining predictions to stakeholders.

Data Splitting Strategy

- **Training Set**: 70% – Used to train the model.
- **Validation Set**: 15% – Used to tune hyperparameters.
- **Test Set**: 15% – Used to evaluate final model performance on unseen data.

Two Hyperparameters to Tune

1. **Number of Trees (n_estimators)**: Controls how many decision trees are used; too few may underfit, too many may slow down the model.
2. **Maximum Tree Depth (max_depth)**: Limits the complexity of each tree; prevents overfitting and reduces computational cost.

4. Evaluation & Deployment (8 points)

Two Evaluation Metrics

1. **Recall**: Critical in identifying as many true dropouts as possible, minimizing the risk of missing students in need of help.
2. **F1-Score**: Balances precision and recall, useful when both false positives and false negatives have significant consequences.

What is Concept Drift?

- **Concept Drift** occurs when the statistical properties of the target variable change over time—e.g., dropout patterns may shift due to new online learning policies.

- **Monitoring Approach**: Track performance metrics (like F1-score) over time and implement periodic model retraining using the latest data.

One Technical Challenge During Deployment

- **Scalability**: Ensuring the model can handle predictions for thousands of students across multiple campuses in real-time without lag, especially during peak periods (e.g., registration season).

Part 2: Case Study Application (40 points)

Scenario: A hospital wants an AI system to predict patient readmission risk within 30 days of discharge.

Problem Definition

Develop an AI system that predicts the likelihood of a patient being readmitted to the hospital within 30 days of discharge.

Objectives

1. Reduce avoidable readmissions to improve patient outcomes.
2. Assist healthcare providers in targeting post-discharge care.
3. Optimize resource allocation and reduce costs associated with penalties for high readmission rates.

Stakeholders

1. **Hospital Management & Care Teams** – for improving service delivery and cost-efficiency.
2. **Patients** – to receive timely follow-up care and avoid repeat hospitalization.

Data Strategy (10 points)

Proposed Data Sources

1. **Electronic Health Records (EHRs)**: Include diagnosis history, treatment plans, medications, length of stay, discharge notes.
2. **Demographic & Socioeconomic Data**: Age, gender, race, income level, insurance status, and housing stability—all of which impact readmission risk.

Two Ethical Concerns

1. **Patient Privacy**: Misuse or unauthorized access to personal health data could violate confidentiality (e.g., HIPAA in the US or Data Protection Act in Kenya).
2. **Discrimination and Fairness**: The model might learn biased patterns (e.g., underestimating risk for certain racial or socioeconomic groups), leading to unequal care.

Preprocessing Pipeline

1. **Missing Data Handling**:
    - Impute missing values using mean (numerical features) or mode (categorical features).
    - For critical missing values (e.g., diagnosis), consider record exclusion.
2. **Feature Engineering**:
    - Derive features like "number of hospital visits in last year", “length of stay”, “number of chronic conditions”, or “polypharmacy count”.
    - Convert discharge notes into sentiment or topic scores using NLP.
3. **Normalization and Encoding**:
    - Normalize continuous variables like age or lab test results.
    - One-hot encode categorical variables such as insurance type and discharge disposition.

Model Development (10 points)

Model Selection

- **Model**: Gradient Boosting Machine (GBM), such as XGBoost or LightGBM.
- **Justification**:
    - Performs well on tabular healthcare data with mixed feature types.
    - Handles class imbalance better than simpler models.
    - Offers interpretability via SHAP values to explain predictions to medical staff.

Hypothetical Confusion Matrix (Out of 100 predictions)

	Predicted: Readmit	Predicted: No Readmit
Actual: Readmit	25 (TP)	5 (FN)
Actual: No Readmit	10 (FP)	60 (TN)

Precision and Recall Calculations

- **Precision** = TP / (TP + FP) = 25 / (25 + 10) = **0.714** (71.4%)
- **Recall** = TP / (TP + FN) = 25 / (25 + 5) = **0.833** (83.3%)

Interpretation: The model correctly identifies most patients who will be readmitted (high recall), and when it predicts a readmission, it is correct 71% of the time (precision).

Deployment (10 points)

Integration Steps into Hospital Systems

1. **Model Packaging**: Export the trained model using a framework like joblib (Python) or ONNX for cross-platform compatibility.
2. **Backend API Service**: Deploy the model via REST API using Flask/FastAPI to allow hospital software to send patient data and receive risk predictions in real time.
3. **EMR Integration**: Connect the API to the hospital's Electronic Medical Records (EMR) system so predictions are shown in the patient dashboard at discharge.
4. **Alerts & Recommendations**: If a patient is flagged as high-risk, the system can trigger a notification for care teams with suggested follow-up actions.
5. **Logging & Feedback**: Store prediction outcomes and actual readmission data for model monitoring and retraining.

Ensuring Compliance with Healthcare Regulations

- **Data Encryption**: Encrypt all patient data at rest and in transit (e.g., using HTTPS and AES-256).
- **Access Controls**: Use role-based access to restrict model use and data viewing to authorized personnel.
- **Audit Trails**: Maintain logs of who accessed the model and when for accountability.
- **Policy Alignment**: Regular audits to ensure compliance with relevant laws such as:
    - **HIPAA (USA)**: For protecting identifiable health info.

    - **Kenya Data Protection Act (2019)**: Ensures lawful processing of sensitive health data.

Optimization (5 points)

Proposed Method to Address Overfitting

- **Regularization with Early Stopping**

Explanation:
During model training (e.g., with XGBoost), implement **early stopping** by monitoring the validation loss. If the loss doesn't improve after a set number of rounds (e.g., 10–20), training halts automatically. This prevents the model from memorizing noise in the training data.

Additional Tip: Combine early stopping with **cross-validation** and regularization hyperparameters like max depth, min child weight, and lambda to further reduce overfitting.

Ethics & Bias (10 points)

How Might Biased Training Data Affect Patient Outcomes?

If the training data contains historical biases—such as underrepresentation of certain ethnic or socioeconomic groups—the model might:

- Underestimate readmission risk for minority or low-income patients.
- Over-prioritize care for majority or higher-income groups.
- Result in **inequitable access** to preventive care, worsening health disparities.

Strategy to Mitigate This Bias

- **Bias Auditing and Rebalancing**
    - Analyze model predictions across sensitive groups (e.g., race, age, gender).
    - Use **reweighting or resampling techniques** to balance the dataset.
    - Apply **fairness constraints** (like equal opportunity or demographic parity) during model training to ensure fair outcomes.

Trade-offs (10 points)

Interpretability vs Accuracy in Healthcare

- **High Accuracy Models** (e.g., deep neural networks) may make more correct predictions but are often "black boxes"—hard to explain.
- **Interpretable Models** (e.g., logistic regression or decision trees) allow doctors to understand **why** a patient is flagged as high risk.
- In healthcare, **interpretability is crucial**: Physicians need to trust and validate the model's logic, especially when patient care is influenced.
- **Trade-off**: You may sacrifice some accuracy to gain transparency and build clinical trust.

Impact of Limited Computational Resources

- May prevent use of complex models like deep learning that require high memory/processing power.
- Could lead to choosing lightweight models (e.g., Logistic Regression, Decision Trees) that are easier to deploy and maintain.
- Impacts may include:
    - Reduced model complexity and feature space.
    - Longer inference times if not optimized.
    - Constraints on real-time deployment or retraining frequency.

Reflection (5 points)

Most Challenging Part of the Workflow

- **Data Collection & Preprocessing** was the most challenging due to:
    - Handling missing/incomplete clinical records.
    - Ensuring data privacy while aggregating sensitive patient information.
    - Identifying and mitigating hidden biases in EHR data.

Improvements with More Time/Resources

- Conduct stakeholder interviews (clinicians, data officers) to better understand operational constraints.
- Implement a **feedback loop** post-deployment to monitor model decisions and outcomes in real-world scenarios.

- Use more advanced techniques like **SHAP values** for explainability and **federated learning** for secure training across multiple hospitals.

Workflow Diagram (5 points)