AI Development Workflow Assignment

Part 1: Short Answer Questions

1. Problem Definition

Problem: Predicting student dropout rates in Kenyan high schools.
Objectives:
1. Identify students at risk of dropping out early.
2. Recommend personalized intervention strategies.
3. Reduce national dropout rates through data-driven insights.

Stakeholders:
- Ministry of Education
- School Administrators

KPI: Reduction in dropout rate (%) within 1 year after system implementation.

2. Data Collection & Preprocessing

Data Sources:
1. National Education Management Information System (NEMIS)
2. School attendance and performance records

Potential Bias: Urban schools may be overrepresented compared to rural schools, skewing predictions.

Preprocessing Steps:
1. Handle missing attendance or income data.
2. Normalize numeric features like scores, income.
3. Encode categorical values (e.g., gender, school type).

3. Model Development

Model: Random Forest — handles non-linearity, robust against overfitting.

Data Split: 70% training, 15% validation, 15% testing using train_test_split() in sklearn.

Hyperparameters:
1. n_estimators: Number of trees in forest
2. max_depth: To prevent overfitting

4. Evaluation & Deployment

Evaluation Metrics:
- Accuracy
- ROC AUC (for probability thresholds)

Concept Drift: It's when data patterns change over time (e.g., due to curriculum changes). Monitor using periodic re-evaluation and re-training.

Deployment Challenge: Scaling the system to rural schools with poor internet.

Part 2: Case Study Application

Problem Scope

Problem: Predicting patient readmission within 30 days after hospital discharge.

Objectives:
1. Identify at-risk patients
2. Recommend post-discharge interventions

Stakeholders:
- Doctors
- Hospital Administrators

Data Strategy

Data Sources:
- Electronic Health Records (EHRs)
- Patient demographics, prescriptions, discharge notes

Ethical Concerns:
1. Patient data privacy (HIPAA)
2. Bias against older or low-income patients

Preprocessing Pipeline:
1. De-identify personal information
2. Handle missing values
3. Feature engineer: average stay, number of visits, diagnosis types

Model Development

Model: Logistic Regression or Random Forest

Confusion Matrix (hypothetical):

Predicted Yes | Predicted No

Actual Yes	70	30
Actual No	20	80

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

Deployment

Steps:
1. Export model (.pkl) using joblib
2. Integrate with hospital dashboard (Flask/Django)
3. Deploy via cloud (e.g., Heroku or local server)

HIPAA Compliance: Encrypt data in transit & at rest, access logs, and secure user roles.

Optimization

Overfitting Fix: Use dropout layers or L2 regularization in Neural Networks; or limit tree depth in Random Forests.

Part 3: Critical Thinking

Ethics & Bias

Issue: Biased training data (more data from urban hospitals) may ignore rural patients, reducing care quality.

Strategy: Use balanced sampling, fairness metrics, and include diverse training data.

Trade-offs

Interpretability vs Accuracy: Logistic Regression is interpretable but less accurate. Neural Nets are accurate but hard to explain.

Resource Constraints: Use lightweight models like Logistic Regression on edge devices with limited RAM.

Part 4: Reflection & Workflow Diagram

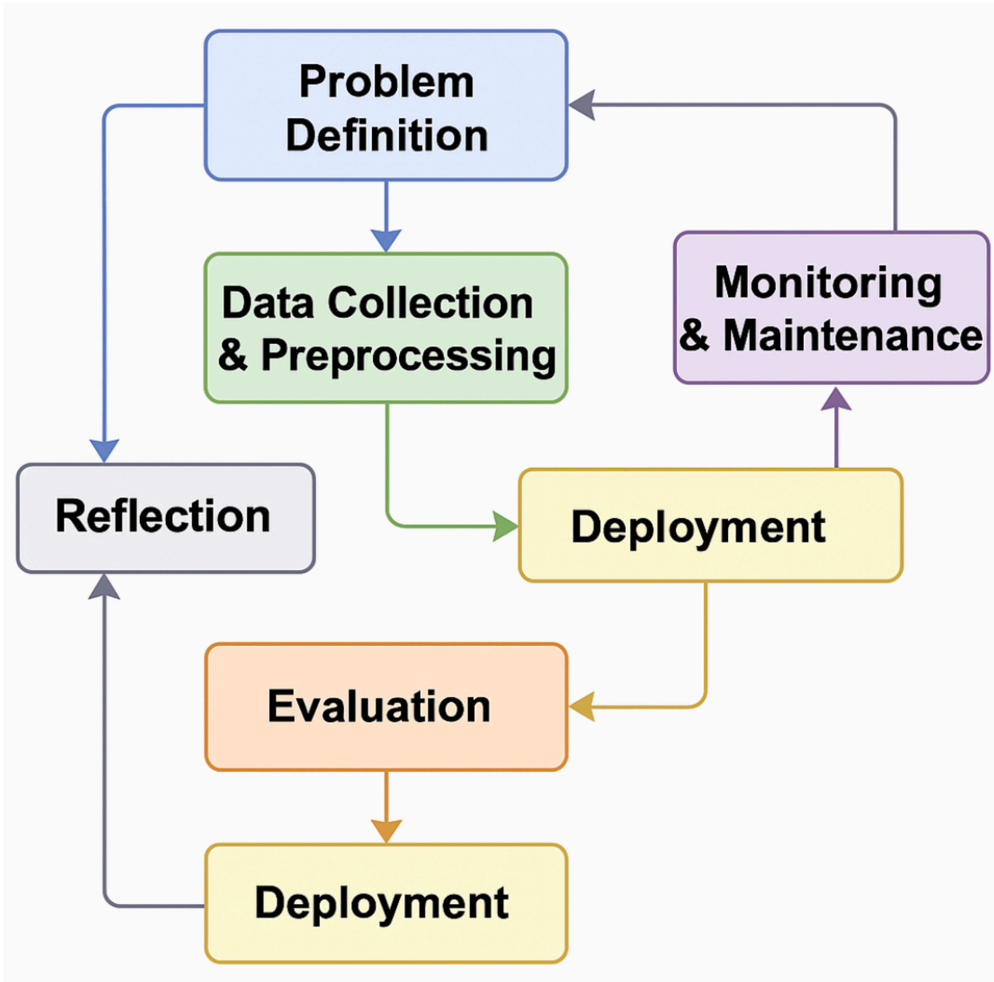
Reflection

Most challenging: Identifying and mitigating bias

What to improve: Collect more representative data and use fairness-aware models

Workflow Diagram

Refer to attached 'AI_Workflow_Diagram.png'


Problem Definition
Data Collection & Preprocessing
Monitoring & Maintenance
Reflection
Deployment
Evaluation
Deployment