AI Development Workflow.

Part 1: Short Answer Questions

1. Problem Definition

Predicting student dropout rates in universities to enable timely intervention and reduce academic attrition.

Objectives:

- Identify at-risk students early for targeted interventions.
- Reduce dropout rates by 20% within one academic year.
- Understand key factors (e.g., attendance, grades) influencing dropout.
- Stakeholders:
- School administrators (resource allocation).
- Teachers (personalized student support).
- KPI: Recall (Sensitivity). Measures the proportion of actual dropouts correctly identified, minimizing false negatives (critical for early intervention).

Stakeholders:

- **University Administration:** For policy formulation, strategic planning, and institutional ranking.
- **Academic Advisors and Lecturers:** For targeted student support, counselling, and academic intervention.

Key Performance Indicator (KPI):

- **Dropout Prediction Accuracy:** Percentage of correctly identified students who eventually drop out.
-

2. Data Collection & Preprocessing

Data Sources:

1. **Student Academic Records:** Grades, GPA, attendance, and course load from the university student management system.
2. **Learning Management System (LMS) Logs:** Login frequency, assignment submission status, quiz participation, and forum activity.

Potential Bias:

- **Socioeconomic Bias:** If the dataset underrepresents students from lower-income backgrounds, the model may not generalize well to their specific challenges, leading to unfair predictions or overlooked interventions.

Preprocessing Steps:

1. **Handling Missing Data:** Impute missing numerical data using mean or median values; remove records with excessive missing entries to maintain data integrity.
2. **Normalization:** Scale numerical features such as grades, login counts, and attendance rates to a standard 0-1 range for balanced input to the model.
3. **Encoding Categorical Variables:** Convert categorical variables such as course codes, departments, and residence status into numerical representations using one-hot encoding or label encoding as appropriate.

3. Model Development

Model Choice:
Random Forest Classifier

Justification:

- Effectively handles datasets with both categorical and numerical variables.
- Provides feature importance, enabling explainability to stakeholders such as lecturers and administrators.
- Less prone to overfitting due to ensemble averaging across multiple decision trees.

Data Splitting Strategy:

- **Training Set:** 70% for model learning and pattern detection.
- **Validation Set:** 15% for hyper-parameter tuning and preventing overfitting.
- **Test Set:** 15% for final evaluation of model generalization performance.

Hyper-parameters to Tune:

1. **Number of Trees (n_estimators):** Determines how many decision trees are built. More trees can improve performance but increase computation time.
2. **Maximum Tree Depth (max_depth):** Controls the depth of each tree to prevent overfitting and ensure the model generalizes well to unseen data.

4. Evaluation & Deployment

Evaluation Metrics:

1. **Accuracy:** Measures the overall correctness of predictions, providing a general understanding of model performance across all classes.
2. **Recall:** Measures the proportion of actual dropouts correctly identified by the model. Critical for this problem because false negatives (missed at-risk students) have serious implications.

Concept Drift:

- **Definition:** Concept drift occurs when statistical properties of the target variable change over time, degrading model performance.
- **Monitoring Strategy:** Continuously evaluate the model on incoming recent data, track performance metrics (e.g. accuracy, recall), and schedule periodic retraining to update the model with new patterns.

Technical Challenge During Deployment:

- **Scalability:** The model must handle predictions for thousands of students simultaneously, especially during peak registration, grading, or advising periods, without delays. This requires optimized server resources, efficient model serving architecture, and load balancing strategies.