1. Problem Definition (6 Points)

Hypothetical Al Problem:

Predicting dropout rates among university students using a combination of academic, behavioural, and socioeconomic data.

Problem Background:

Student retention is a key challenge for educational institutions. Dropouts n egatively impact institutional rankings and student futures. Many universiti es lack predictive systems that can flag at-risk students early enough to int ervene.

Objectives:

1 . Early Identification: Detect potential dropouts before critical decision points.

1. Causal Analysis: Uncover the most impactful features contributing to attrition.
2. Improve Retention Strategy: Use data insights to shape personalized interventions and support systems.

Stakeholders:

* + University Administrators: Responsible for student welfare, retentio n policies, and institutional success.
  + Students: Directly impacted by targeted support, guidance. and instit utional planning.

KPI — Key Performance Indicator:

uFl Score: An ideal metric for dropout prediction since it balances the ris k of false positives (students incorrectly flagged as at-risk) and false negati ves (missed predictions). Precision and recall together offer a nuanced perf ormance measure.

2 Data Collection & Pre-processing (8 Points)

Data Sources and Types:

Student Information System (SIS): Contains historical data such a s grades, attendance. participation in extracurncular activities. and di sciplinary records.

1. Student Surveys and Psychometric Assessments: Self-reported e ngagement levels, satisfaction, mental health indicators, and financia I pressures.

Potential Data Bias:

Socioeconomic Bias: Historical records might show students from low-inc ome families as disproportionately prone to dropping out. If not properly mi tigated, this can lead to unethical profiling or unfair prioritization.

Data Pre-processing Steps:

Missing Data Handling:

* + - Use mean or median imputation for nu merical data.
    - Apply domain-specific rules or predictive models for categorical data.

2. Normalization I Scaling:

 Standardize numerical features (e.g., GPA, attendance percent age) using Min-Max or Z-score normalization for better model c onvergence.

1. Categorical Encoding:
   * Use one-hot encoding for nominal categorical features (e.g., 'Progra m of Study').
   * Apply ordinal encoding where a meaningful order exists (e. satisfa ction levels: low 'n high).

3. Model Development (8 Points)

Model Selected:

Random Forest Classifier

Justification for Model Choice:

* + Can handle large feature spaces and mixed data types (numerical + c ategorical).
  + Robust to outliers and noisy data.
  + Ensemble method reduces risk of over fitting.
  + Offers feature impotThnce scores. aiding explain ability.

Data Splitting Strategy:

* + Training Set (70%) — Used for initial learning of patterns.
  + Validation Set (15%) — Used for hyper parameter tuning and model s election.
  + Test Set (15%) — Used for evaluating generalization performance and final metrics.

Hyper parameters to Tune:

* + n\_est matc rs:
  + Number of trees.
  + More trees usually improve accuracy but increase computation al cost.

2. max\_depth:

* Limits how deep each tree can grow.
* Prevents over fitting and helps balance precision/recall.

4 Evaluation & Deployment (B Points)

Evaluation Metrics:

1. . Fl Score: Essential for balancing sensitivity (recall) and precision in p redicting dropout risk. Reduces overreaction to false positives.
2. ROC-AUC Score: Assesses the mode[s ability to separate dropout v

s. non-dropout cases rega rdless of threshold. AUC above 0. BO sugge Sts strong cla ssifi cation performance.

Concept Drift:

Definition: Changes in the statistical properties of input data over time whi ch reduce model accuracy. For example, new programs or shifting student demographics may change dropout patterns.

Monitoring Concept Drift:

* + Deploy real-time evaluation dashboards.
  + Periodically retrain using newer data
  + Use online drift detection tools (e. g.. ADWINJ Page-Hinkley) to trigger automatic alerts for retraining.

Deployment Challenge — Scalability:

Definition: The systems ability to scale and maintain performance unde r increased load, such as being used across multiple departments or camp uses.

Solutions to Scalability:

* + Containerization: Use Docker to package the model with all depende ncies.
  + Orchestration: Use Kubernetes to scale deployment across cloud ser vers.
  + APIs: Build RESTful APIs for easy integration with SIS dashboards.
  + Cloud Hosting: Utilize scalable platforms like AWS Lambda, Azure M L, or Google Cloud Run for cost-effective infrastructure.

Ethical Considerations

* + Bias Correction: Apply re-weighting techniques, fairness-aware algori thms, or data audits.
  + Interpretability: Provide transparency using feature importance plots or model explanations (e.g., SHAP or LIME).
  + Student Privacy: Protect sensitive data through encryption, anonymiz ation, and robust access controls.
  + Consent & Transparency: Inform students how their data is used and give them the option to opt\*n/out.