1. Problem Definition

Hypothetical AI Problem: *Predicting student dropout rates.*

Three objectives:

1. Identify at-risk students early.
2. Improve student retention through targeted interventions.
3. Allocate institutional resources efficiently.

Two stakeholders:

1. University administration.
2. Students and academic advisors.

One KPI:

- **Dropout prediction accuracy** (or reduction in actual dropout rates after interventions).

2. Data Collection & Preprocessing (8 points)

Two data sources:

1. Student academic records (grades, attendance).
2. Learning management system (LMS) activity logs.

One potential bias:

- **Socioeconomic bias:** Students from disadvantaged backgrounds may have less online activity, leading the model to unfairly classify them as high-risk.

Three preprocessing steps:

1. Handle missing values using imputation.
2. Normalize numerical features (e.g., attendance rate).
3. Encode categorical variables (e.g., program type) using one-hot encoding.

3. Model Development (8 points)

Chosen model and justification:

- **Random Forest** because it handles mixed data types well, is robust to noise, and provides interpretable feature importance.

Data splitting:

- **70% training, 15% validation, 15% testing** to ensure the model generalizes and tuning does not leak into final evaluation.

Two hyperparameters to tune:

1. **Number of trees (n_estimators):** Controls model complexity and performance.
2. **Maximum tree depth (max_depth):** Prevents overfitting by limiting how deep each tree can grow.

4. Evaluation & Deployment (8 points)

Two evaluation metrics and relevance:

1. **Accuracy:** Measures overall correctness of dropout predictions.
2. **Recall (for dropout class):** Important because failing to detect actual at-risk students can harm outcomes.

Concept drift:

- **Definition:** When the statistical properties of input data or target labels change over time, causing the model's performance to degrade.
- **Monitoring:** Track prediction accuracy weekly, use drift-detection tools (e.g., population stability index), and compare feature distributions to historical baselines.

One technical deployment challenge:

- **Scalability:** Serving predictions in real time for thousands of students may require load balancing or autoscaling infrastructure.