◆ 1. Problem Definition

Problem:
Predicting student dropout rates in a university.

Objectives:

1. Identify students at risk of dropping out early in the semester.
2. Suggest targeted interventions based on predicted risk.
3. Improve overall student retention rate by 15%.

Stakeholders:

- University administration
- Academic advisors

KPI (Key Performance Indicator):

- Retention rate improvement over two semesters

◆ 2. Data Collection & Preprocessing

Data Sources:

1. Academic records (grades, attendance)
2. Student engagement data (LMS logins, participation)

Potential Bias:

- Students without consistent internet access may appear disengaged, mislabelling them as "at risk."

Preprocessing Steps:

1. Handle missing values (e.g., attendance gaps)
2. Normalize numerical features (e.g., test scores)
3. Encode categorical data (e.g., gender, course code)

◆ 3. Model Development

Model Choice:
Random Forest — handles both categorical and numerical data well, interpretable, and resistant to overfitting.

Data Splitting Strategy:

- 70% training, 15% validation, 15% test
  (Stratified to preserve dropout ratios)

Hyperparameters to Tune:

1. n_estimators – controls number of trees for better accuracy

2. max_depth – prevents overfitting by limiting tree growth

🔹 4. Evaluation & Deployment

Evaluation Metrics:

- **Precision:** Ensures we don't incorrectly classify students as "at risk"
- **Recall:** Ensures we capture most of the truly at-risk students

Concept Drift:
Changes over time in student behaviours (e.g., post-pandemic patterns).
Monitoring: Track model accuracy over semesters and retrain regularly.

Deployment Challenge:
Scalability – integrating the model with live student management systems across different faculties.

🏥 Part 2: Case Study Application (40 Points)

🔹 Problem Scope

Problem:
Predict if a patient will be readmitted within 30 days after hospital discharge.

Objectives:

- Identify high-risk patients
- Support better discharge planning
- Reduce hospital readmission costs

Stakeholders:

- Doctors and discharge coordinators
- Hospital management

🔹 Data Strategy

Data Sources:

- Electronic Health Records (EHRs)
- Demographic data (age, gender, zip code)

Ethical Concerns:

1. Patient privacy (handling sensitive data)
2. Algorithmic bias (e.g., against certain socioeconomic groups)

Preprocessing Pipeline:

1. Impute missing values (e.g., lab results)

2. Normalize numerical features (e.g., age, BMI)
3. One-hot encode categorical variables (e.g., diagnosis codes)
4. Feature engineering:
    - Create a “readmissions in past 6 months” feature
    - Extract comorbidities count from diagnoses

◆ **Model Development**

Model Choice:
Logistic Regression — interpretable and suitable for binary classification with healthcare constraints.

Hypothetical Confusion Matrix:

	Predicted Yes	**Predicted No**
Actual Yes	45	5
Actual No	10	40

Precision: 45 / (45 + 10) = 0.818
Recall: 45 / (45 + 5) = 0.9

◆ **Deployment (10 pts)**

Integration Steps:

1. Convert model into a REST API
2. Integrate into hospital's EHR dashboard
3. Provide visual risk scores for each patient at discharge
4. Schedule periodic retraining

Regulatory Compliance:

- Ensure full **HIPAA compliance**
- Encrypt patient data during processing and storage
- Limit access via authentication

◆ **Optimization**

Overfitting Fix:

- Use **cross-validation** and **dropout (if neural nets used)** to generalize the model across patient populations

---

Part 3: Critical Thinking (20 Points)

◆ **Ethics & Bias**

Impact of Bias:
If the model is trained on biased historical data (e.g., only patients from urban hospitals), it may underpredict readmissions in rural or underserved communities — leading to worse care for those patients.

Mitigation Strategy:

- Ensure the training data is **diverse and representative**
- Use **fairness metrics** during model evaluation
- Regularly audit predictions for bias

◆ **Trade-Offs**

Interpretability vs Accuracy:

- High-accuracy models (e.g., deep neural nets) are harder to interpret, which may be **unacceptable in healthcare** where explainability is critical.
- In some cases, simpler models (e.g., logistic regression or decision trees) are preferred even if accuracy is slightly lower — because they're easier for doctors to trust.

Computational Limits:

- If the hospital has limited computing resources, models with heavy training demands (e.g., deep learning) may be impractical.
- Choose **lightweight models** like logistic regression or smaller tree ensembles for real-time inference.

Part 4: Reflection & Diagram (10 Points)

◆ Reflection

Most Challenging Part:

Designing a balanced preprocessing pipeline — ensuring the data was clean while also ethical and privacy-conscious.

Improvements with More Time:

I would:

1. Test more models (like gradient boosting)
2. Collect more diverse data
3. Deploy the model in a secure sandbox for trial use

- Workflow Diagram (5 pts)

[Problem Definition]

↓

[Data Collection]

↓

[Data Preprocessing]

↓

[Model Selection]

↓

[Model Training & Tuning]

↓

[Evaluation]

↓

[Deployment]

↓

[Monitoring & Maintenance]