Case Study: Predicting Patient Readmission Risk

1. Problem Scope (5 points)

Problem Statement: Hospitals face high costs and patient risks due to unplanned readmissions within 30 days of discharge. An AI system can help predict which patients are at high risk, enabling proactive care.

Objectives:

- Predict 30-day readmission risk using patient data.
- Support clinicians with actionable insights for discharge planning.
- Reduce readmission rates and improve patient outcomes.

Stakeholders:

- Hospital administrators and care coordinators.
- Patients and their families.

2. Data Strategy (10 points)

Data Sources:

- Electronic Health Records (EHRs): diagnoses, procedures, medications, discharge notes.
- Demographic data: age, gender, socioeconomic status.
- Admission history: previous visits, length of stay, comorbidities.

Ethical Concerns:

1. **Patient Privacy:** Sensitive health data must be protected from unauthorized access.
2. **Bias in Predictions:** Historical data may reflect systemic biases (e.g., underdiagnosis in certain populations), leading to unfair predictions.

Pre-processing Pipeline:

- **Step 1: Data Cleaning**
  - Remove duplicates, handle missing values (e.g., imputation for lab results).
- **Step 2: Feature Engineering**
  - Create features like “number of prior admissions,” “length of stay,” “comorbidity index.”
- **Step 3: Encoding & Scaling**
  - One-hot encode categorical variables (e.g., discharge type).
  - Normalize numerical features (e.g., age, lab values).

3. Model Development (10 points)

Model Choice:

- **Gradient Boosting (e.g., XGBoost)** — handles tabular data well, robust to missing values, interpretable via feature importance.

Hypothetical Confusion Matrix:

	Predicted: Readmit	**Predicted: No Readmit**
Actual: Readmit	80	20
Actual: No Readmit	30	70

Calculations:

- **Precision** = TP / (TP + FP) = 80 / (80 + 30) = **0.727**
- **Recall** = TP / (TP + FN) = 80 / (80 + 20) = **0.800**

4. Deployment (10 points)

Integration Steps:

1. Package model as a REST API using Flask or FastAPI.
2. Connect to hospital's EHR system via secure endpoints.
3. Embed predictions into clinician dashboards with risk scores.
4. Enable feedback loop for model refinement (e.g., flag incorrect predictions).

Regulatory Compliance:

- Ensure **HIPAA** compliance by:
    - Encrypting data in transit and at rest.
    - Role-based access control for model outputs.
    - Logging and auditing all data access events.

5. Optimization (5 points)

Method to Address Overfitting:

- **Cross-validation with early stopping** — prevents the model from learning noise by halting training when validation performance plateaus.