Week 4 Assignment: AI in Software Engineering – “Building Intelligent Software Solutions”.

Part 1: Short Answer Questions (30 points)

1. Problem Definition (6 points)

Hypothetical AI Problem:
Predicting student dropout rates in online learning platforms.

Objectives:

1. Identify students at risk of dropping out early.
2. Provide actionable insights to academic advisors.
3. Improve student retention and engagement through early interventions.

Stakeholders:

- University administration.
- Students (end-users).

Key Performance Indicator (KPI):

- **Dropout prediction accuracy** (percentage of correctly predicted at-risk students).

2. Data Collection & Preprocessing (8 points)

Data Sources:

1. Learning Management System (LMS) activity logs.
2. Student demographic and performance databases.

Potential Bias:
Data may overrepresent students with consistent internet access, disadvantaging those from low-connectivity areas.

Preprocessing Steps:

1. Handle missing attendance or score data through mean imputation.
2. Normalize numeric features (e.g., login frequency, grades).
3. Encode categorical features (e.g., gender, course level) using one-hot encoding.

3. Model Development (8 points)

Model Choice:
Random Forest Classifier — robust, interpretable, and handles both numerical and categorical features well.

Data Split:

- 70% training, 15% validation, 15% test (stratified sampling to balance dropout/non-dropout classes).

Hyperparameters to Tune:

1. **Number of trees (n_estimators):** affects model stability and performance.
2. **Maximum depth:** controls overfitting by limiting tree complexity.

4. Evaluation & Deployment (8 points)

Evaluation Metrics:

1. **Precision:** proportion of correctly identified at-risk students among all predicted as at-risk.
2. **Recall:** ability to identify all actual at-risk students (important for retention goals).

Concept Drift:
Occurs when data patterns change over time (e.g., new online course structures).
Monitoring: retrain the model quarterly and compare live performance metrics with baseline accuracy.

Technical Challenge:
Scalability — ensuring the system handles large real-time LMS data without latency.

Part 2: Case Study Application (40 points)

Scenario: Hospital AI for Predicting Patient Readmission Risk

1. **Problem Scope (5 points)**

Problem:
Develop an AI model to predict the likelihood of a patient being readmitted within 30 days after discharge.

Objectives:

- Identify high-risk patients early.
- Support clinicians in post-discharge planning.
- Reduce hospital costs and improve patient outcomes.

Stakeholders:

- Medical staff (doctors, nurses).
- Hospital administration.
- Patients.

2. Data Strategy (10 points)

Data Sources:

- Electronic Health Records (EHRs).
- Demographics and past medical history.
- Discharge summaries and medication logs.

Ethical Concerns:

1. **Patient privacy** — sensitive health data must be encrypted.
2. **Data bias** — some groups (e.g., elderly, minorities) may be underrepresented.

Preprocessing & Feature Engineering Pipeline:

1. Handle missing data using median imputation.
2. Normalize continuous variables (e.g., age, blood pressure).
3. One-hot encode categorical features (diagnosis, insurance type).
4. Feature engineer:
    - Number of previous admissions.
    - Length of hospital stay.
    - Medication count and lab test variability.

3. Model Development (10 points)

Model Choice:
Gradient Boosting Machine (XGBoost) — excels with structured medical data, provides feature importance, and handles imbalance effectively.

Hypothetical Confusion Matrix:

	Predicted Readmit	**Predicted No Readmit**
Actual Readmit	80	20
Actual No Readmit	15	85

Precision: 80 / (80 + 15) = **0.842 (84.2%)**
Recall: 80 / (80 + 20) = **0.80 (80%)**

4. Deployment (10 points)

Integration Steps:

1. Containerize the model using Docker.
2. Deploy via API in the hospital’s patient management system.
3. Enable real-time predictions at discharge time.
4. Provide dashboards for doctors showing patient risk scores.

Regulatory Compliance:

- Ensure HIPAA compliance via encryption, anonymization, and access control.
- Conduct periodic audits for model fairness and security.

5. Optimization (5 points)

Method to Address Overfitting:
Use **cross-validation with early stopping** and **L2 regularization** to ensure the model generalizes well.

Part 3: Critical Thinking (20 points)

1. Ethics & Bias (10 points)

Impact of Biased Data:
If training data underrepresents certain demographic groups, the model may inaccurately assess their readmission risks — potentially leading to unequal care or neglect.

Mitigation Strategy:
Implement **data rebalancing and fairness-aware algorithms**, and conduct bias audits during model evaluation.

2. Trade-offs (10 points)

Interpretability vs. Accuracy:
Highly accurate deep models (e.g., neural networks) may lack transparency, making it hard for doctors to trust predictions. Simpler models (e.g., logistic regression) offer interpretability but might be less accurate.

Resource Constraints:
With limited computational resources, prefer **lightweight models (e.g., Random Forest or Logistic Regression)** and reduce feature dimensionality to maintain efficiency.

Part 4: Reflection & Workflow Diagram (10 points)

Reflection (5 points)

Most Challenging Part:
Data preprocessing — ensuring data quality, addressing missing values, and mitigating bias were complex and time-consuming.

Improvement with More Time/Resources:
I would collect more longitudinal patient data, apply explainable AI (XAI) tools like SHAP for interpretability, and automate retraining for continuous improvement.