AI Development Workflow Report

Course: AI for Software Engineering
Assignment: Understanding the AI Development Workflow

---

1. Introduction

Artificial Intelligence (AI) has become a cornerstone in modern software engineering, providing predictive insights, automation, and decision support across industries. The AI development workflow ensures a systematic approach, from defining problems to deploying models. This report demonstrates the workflow through a hypothetical student dropout prediction problem and a hospital readmission risk case study. The report also analyzes ethical considerations and deployment challenges in real-world settings.

---

2. Problem Definition

2.1 Hypothetical AI Problem

Predicting student dropout rates.

Objectives:

1. Identify students at high risk of dropping out.
2. Provide actionable interventions to reduce dropouts.
3. Optimize allocation of academic support resources.

Stakeholders:

- University administration
- Academic advisors and faculty

KPI: Percentage of correctly predicted at-risk students (prediction accuracy).

2.2 Hospital Case Study Problem

Predict 30-day patient readmission risk.

Objectives:

1. Identify high-risk patients before discharge.

2. Reduce avoidable readmissions.
3. Optimize hospital resource allocation.

Stakeholders:

- Hospital management
- Physicians and care coordinators

---

3. Data Collection & Preprocessing

3.1 Hypothetical Student Dataset

Sources:

- Student academic records (grades, attendance)
- Engagement surveys and socio-economic data

Potential Bias: Students from underrepresented backgrounds may have incomplete records, affecting model fairness.

Preprocessing Steps:

1. Impute missing numeric values using median.
2. Normalize numeric features (GPA, attendance).
3. Encode categorical variables (program type) using one-hot encoding.

3.2 Hospital Dataset

Sources:

- Electronic Health Records (EHRs) – diagnoses, lab results, medications
- Patient demographics – age, gender, socio-economic status

Ethical Concerns:

- Patient privacy and HIPAA compliance
- Algorithmic bias affecting disadvantaged groups

Preprocessing Steps:

1. Impute missing lab values or prior admission counts.
2. Normalize numeric features.
3. Encode categorical variables (e.g., discharge type).
4. Feature engineering: calculate comorbidity scores, prior admissions count.

---

4. Model Development

4.1 Student Dropout Prediction

- **Model:** Random Forest – robust to missing data, interpretable via feature importance.
- **Data Split:** Training 70%, Validation 15%, Test 15%
- **Hyperparameters Tuned:**
    - Number of trees (n_estimators)
    - Maximum tree depth (max_depth)

4.2 Hospital Readmission Prediction

- **Model:** XGBoost – handles tabular, imbalanced data efficiently.
- **Train/Test Split:** 70/30
- **Hyperparameters Tuned:**
    - Learning rate (controls convergence speed)
    - Maximum tree depth (prevents overfitting)

Hypothetical Confusion Matrix for Hospital Case Study:

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

Metrics:

- Precision = 80 / (80 + 15) ≈ 0.842
- Recall = 80 / (80 + 20) = 0.80

---

5. Evaluation & Deployment

Evaluation Metrics:

- **F1-score:** balances precision and recall for imbalanced classes
- **ROC-AUC:** measures model discrimination ability

Concept Drift: Change in student behavior or patient population over time reducing model accuracy.

Deployment Challenges:

- Scalability: Models must handle predictions for thousands of students or patients in real time.
- Compliance: Hospital models must comply with HIPAA and patient privacy regulations.

Deployment Steps (Hospital Case Study):

1. Export trained model as REST API (Flask/FastAPI)
2. Integrate API with EHR system for real-time scoring
3. Display results via clinician dashboard
4. Implement access controls, encryption, and logging

---

6. Critical Thinking

6.1 Ethics & Bias

- Biased training data may underrepresent certain student demographics or patient groups, leading to inequitable interventions or treatment recommendations.
- **Mitigation Strategy:** Oversample underrepresented groups or generate synthetic samples (SMOTE), monitor performance across demographics.

6.2 Trade-offs

- **Interpretability vs Accuracy:** Complex models (XGBoost) are accurate but less interpretable; simpler models (logistic regression) are transparent but less precise.
- **Resource Constraints:** Hospitals with limited computational resources may prefer simpler models to reduce inference latency and memory usage.

7. Reflection & Workflow Diagram

7.1 Reflection

- **Challenge:** Preprocessing real-world datasets — missing values, inconsistent records, sensitive information.
- **Improvement:** With more time/resources, collect larger datasets, automate data cleaning, and perform bias audits.

7.2 Workflow Diagram

Problem Definition → Data Collection → Data Preprocessing → Model Development → Evaluation → Deployment → Monitoring & Maintenance

(Insert diagram image in figures/workflow_diagram.png here in your PDF)

---

8. References

1. Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow.*
2. Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling.*
3. HIPAA Guidelines: https://www.hhs.gov/hipaa/index.html