PLP Academy Community Article: Understanding the AI Development Workflow

Course: AI for Software Engineering\ **Assignment Title**: Understanding the AI Development Workflow\ **Duration**: 7 Days\ **Author**: Grateful Juma

---

🚀Overview

This article explores the end-to-end process of applying the AI Development Workflow to a real-world case: predicting hospital patient readmission within 30 days. It includes theoretical insights, technical choices, ethical considerations, and deployment strategies, aligned with the CRISP-DM framework.

Case Study: Predicting Patient Readmission

Problem: Build an AI model that predicts whether a discharged patient will be readmitted within 30 days.

Objectives:

- Lower readmission rates.
- Assist clinicians with early warnings.
- Improve healthcare outcomes.

Stakeholders:

- Hospital administrators
- Healthcare providers

Data Strategy:

- **Sources**: EHRs, demographics
- **Ethics**: Privacy & consent
- **Preprocessing**: Imputation, encoding, feature engineering

Modeling:

- **Choice**: Gradient Boosting (XGBoost)
- **Metrics**: Precision (0.78), Recall (0.70)
- **Confusion Matrix**: [TP=70, FN=30, FP=20, TN=80]

Deployment:

- Packaged API (Flask)
- Integrated into EHR system
- Encryption + access controls

Optimization:

- Regularization + early stopping to mitigate overfitting

Short Answer Insights

1. Problem Definition:

- **Example**: Predict employee attrition
- **KPI**: Attrition rate post-model deployment

2. Data Preprocessing:

- Normalize, impute, encode
- Watch for survey bias

3. Model Development:

- Use Random Forest for structured HR data
- Hyperparameters: `max_depth`, `n_estimators`

4. Evaluation:

- AUC-ROC & F1-score
- Handle concept drift with retraining triggers

🏥Critical Thinking

Ethics:

- Biased data can worsen patient outcomes
- Mitigation: Fairness-aware algorithms (e.g., reweighing)

Trade-offs:

- High-accuracy black-box vs. interpretable models
- Limited hardware favors lightweight models

🗃️Reflection & Diagram

Challenge: Aligning technical AI solutions with clinical reality

Improvement: Add explainability (e.g., SHAP) and stakeholder interviews

Workflow Diagram:

1. Define Problem
2. Collect & Clean Data
3. Develop Model
4. Evaluate
5. Deploy + Monitor

🏠GitHub Repo

Structure:

```
ai-readmission-predictor/
├── src/
│   ├── preprocessing.py
│   ├── model_training.py
│   └── deploy_api.py
├── data/
│   └── sample_ehr.csv
├── notebooks/
│   └── exploratory_analysis.ipynb
├── README.md
├── requirements.txt
└── LICENSE
```

Includes:

- Well-commented code
- Requirements file
- Instructions to run & deploy model

Final Deliverables

- 📄PDF Report
- 💻GitHub Repo
- 📣This Article (PLP Submission)
- 🕐Slide Deck (available on request)

Let's build AI that works for people—not just the system.