Week 5 Assignment on AI for Software Engineering

Part 1

1. Problem Definition

Hypothetical AI Problem: *Predicting student dropout rates in higher education institutions.*

Objectives:

- Identify students at risk of dropping out early in the academic year.
- Enable targeted interventions to improve retention.
- Reduce institutional dropout rates and associated costs.

Stakeholders:

- University administrators
- Academic advisors and student support services

Key Performance Indicator (KPI):

- Dropout prediction accuracy (e.g., percentage of correctly identified at-risk students)

2. Data Collection & Preprocessing

Data Sources: Kaggle – Student Data CSV

- Student Information Systems (SIS): Academic records, attendance, course enrolment
- Learning Management Systems (LMS): Assignment submissions, login frequency, engagement metrics

Potential Bias:

- Socioeconomic bias: Students from underrepresented or low-income backgrounds may be overrepresented in dropout predictions if the model learns from historical inequalities.

Preprocessing Steps:

1. Handle missing data through imputation or removal.
2. Normalize numerical features such as grades and attendance rates.
3. Encode categorical variables like gender, program type, or major.

3. Model Development

Model Choice: *Random Forest Classifier*

- Justification: Handles non-linear relationships, robust to outliers, interpretable via feature importance, and performs well on tabular data.

Data Splitting:

- Training set: 70%
- Validation set: 15%
- Test set: 15% (Stratified sampling to maintain class balance)

Hyperparameters to tune:

1.n_estimators (number of trees): Affects model complexity and performance.
2.max_depth (maximum tree depth): Controls overfitting by limiting tree growth.

4. Evaluation & Deployment

Evaluation Metrics:

1. Precision: Measures how many predicted dropouts were correct and important for minimizing false positives.
2. Recall: Measures how many actual dropouts were correctly identified—critical for early intervention.

Concept Drift:

- ***Definition*:** When the statistical properties of input data change over time, it reduces model accuracy.

- ***Monitoring*:** Use rolling performance metrics, retrain periodically, and implement drift detection algorithms (e.g., DDM or ADWIN).

Technical Challenge:

- **Scalability:** Ensuring the model can handle large volumes of real-time student data across departments without latency or performance degradation.

Reference: Webb, G. I., Hyde, R., Cao, H., Nguyen, H. L., & Petitjean, F. (2016). *Characterizing concept drift*. **Data Mining and Knowledge Discovery**, 30(4), 964–994.

Part 2

Case Study: *Predicting 30-Day Hospital Readmission Risk Using AI*

1: Problem Scope

- **Problem**: High 30-day readmission rates increase healthcare costs and reflect potential gaps in post-discharge care.
- **Objective**: Develop an AI model to predict patients at risk of readmission within 30 days, enabling early intervention.
- **Stakeholders**: Hospital administrators, clinicians, care coordinators, patients, and data governance teams.

2: Data Strategy

- **Data Sources**:
  - Electronic Health Records (EHRs): Diagnoses, procedures, medications, lab results
  - Demographics: Age, gender, ethnicity, socioeconomic status
  - Social Determinants of Health: Housing, support systems, insurance status
  - Historical readmission data
- **Ethical Concerns**:
  - *Patient Privacy*: Ensure data is anonymized and access-controlled (HIPAA/GDPR compliant).

○ *Bias and Fairness*: Prevent discrimination against vulnerable populations through bias audits and representative training data.

- **Preprocessing Pipeline**:
  - Data cleaning: Handle missing values, remove duplicates.
  - Feature engineering:
    - Binary target: readmitted within 30 days
    - Aggregated features: prior admissions, comorbidities, medication count
    - Encoding: One-hot or label encoding for categorical variables
    - Normalization: Scale numerical features
  - Train-test split: Stratified sampling to preserve class balance.

3: Model Development

- **Model Choice**: *Random Forest Classifier*
  - Justification: Handles non-linear relationships, robust to outliers, interpretable via feature importance
- ● **Hypothetical Confusion Matrix**
- **Metrics**:
  - *Precision* = 80 / (80 + 30) = **0.727**
  - *Recall* = 80 / (80 + 20) = **0.800**

4: Deployment

- **Integration Steps**:
  - Deploy model as a RESTful API
  - Integrate with EHR system to display risk scores
  - Trigger alerts for high-risk patients
  - Monitor performance and retrain periodically ●

Regulatory Compliance:

- Ensure HIPAA compliance via encryption, audit logs, and role-based access
- Maintain consent records and explainability tools for transparency

5: Optimization

- **Overfitting Mitigation**:

- Use *k-fold cross-validation with early stopping* to prevent overfitting and improve generalization

Part 3: Critical Thinking

Ethics & Bias

Impact of Biased Training Data on Patient Outcomes.

When training data contains historical biases such as underrepresentation of certain ethnic groups, older patients, or low-income populations, the model might systematically underestimate their risk of readmission. This could lead to fewer interventions for these patients, widening health disparities, and potentially endangering lives. For example, if patients from marginalized backgrounds historically had fewer readmissions because they lacked access to follow-up care, the model may incorrectly learn that they are at lower risk.

Mitigation Strategy: Fairness-Aware Modelling.

One effective strategy to reduce bias is re-weighting or re-sampling the dataset to ensure equitable representation across key demographic groups (e.g., age, gender, race). This can be paired with fairness-aware algorithms that constrain model outputs to minimize group-level disparities. Additionally, implementing performance audits across subpopulations ensures the model performs consistently for all patients.

References

- Obermeyer, Z., Powers, B., Vogeli, C., & Mullainathan, S. (2019). *Dissecting racial bias in an algorithm used to manage the health of populations*. Science, 366(6464), 447–453.
- Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). *A Survey on Bias and Fairness in Machine Learning*. ACM Computing Surveys.

Trade-offs

Model Interpretability vs. Accuracy in Healthcare.

Highly accurate models like deep neural networks may offer better predictive performance but lack transparency. In contrast, interpretable models like logistic regression or decision trees may not be as precise but are crucial in healthcare settings where clinicians need to understand and trust AI recommendations. Interpretability is also important for auditing, regulatory compliance, and explaining decisions to patients and families—especially in life-critical scenarios.

Computational Resource Constraints and Model Choice.

Hospitals with limited computational infrastructure may not support complex models that require high memory, GPU acceleration, or real-time inference. In such cases, simpler models with low latency (e.g., logistic regression or shallow decision trees) are more practical. They're easier to deploy on edge devices or within existing IT systems.

References

- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. ACM SIGKDD.
- Holzinger, A., Biemann, C., Pattichis, C. S., & Kell, D. B. (2017). *What do we need to build explainable AI systems for the medical domain?* Artificial Intelligence in Medicine.

Part 4: Reflection & Workflow Diagram

1. Reflection

Most Challenging Stage – Preprocessing and Bias Mitigation,

The most challenging part of the AI workflow was the data preprocessing phase, particularly in addressing missing values, engineering effective features, and mitigating biases in the training dataset. Preprocessing directly shapes model performance and ethical integrity, especially in healthcare where biased predictions can impact patient safety and equity.

Improvement Opportunities with More Time or Resources.

Given additional resources, I would implement a bias auditing framework using tools like IBM AI Fairness 360 or Aequitas, allowing real-time subgroup analysis during development. Additionally, investing in a real-time data integration pipeline and engaging interdisciplinary experts (e.g., clinicians, ethicists, data scientists) would refine both feature quality and interpretability.

References

- Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). *A Survey on Bias and Fairness in Machine Learning*. ACM Computing Surveys.
- Suresh, H., & Guttag, J. V. (2019). *A Framework for Understanding Unintended Consequences of Machine Learning*. Communications of the ACM, 62(11), 62–71
- Kansagara, D., Englander, H., Salanitro, A., et al. (2011). *Risk prediction models for hospital readmission: a systematic review*. JAMA, 306(15), 1688–1698.

Diagram

A flowchart of the AI Development Workflow including all stages

AI Development Workflow – Flowchart Representation