AI Development Workflow: From Theory to Healthcare Application

Part 1: Short Answer Questions

1. Problem Definition (6 points)

Hypothetical AI Problem: "Predicting Student Dropout Rates in Online Learning Platforms"

Objectives:

1. Identify at-risk students with 85% accuracy 4 weeks before potential dropout
2. Provide personalized intervention recommendations to academic advisors
3. Reduce overall student dropout rate by 25% within one academic year

Stakeholders:

1. Students and their families
2. Educational institution administration and faculty
3. Online learning platform developers

Key Performance Indicator (KPI): Early Intervention Success Rate - Percentage of flagged at-risk students who complete the semester after receiving targeted interventions.

2. Data Collection & Preprocessing (8 points)

Data Sources:

1. Learning Management System (LMS) logs: login frequency, assignment submission times, video watching duration
2. Student information system: demographic data, prior academic performance, course enrollment history

Potential Bias: Geographic bias - students from regions with poor internet infrastructure may show similar engagement patterns to at-risk students, leading to false positives.

Preprocessing Steps:

1. **Missing Data Handling:** Use multiple imputation for continuous variables (attendance rates) and mode imputation for categorical variables (course preferences)
2. **Normalization:** Apply min-max scaling for engagement metrics (0-1 range) to ensure equal feature weighting

3. **Feature Engineering:** Create temporal features like "weekly engagement trend" and "submission delay patterns"

3. Model Development (8 points)

Chosen Model: Gradient Boosting Machine (XGBoost)

Justification:

- Handles mixed data types well (numerical engagement metrics + categorical demographic data)
- Provides feature importance for interpretability
- Robust to outliers in student behavior data
- Better performance than Random Forests for tabular data with complex interactions

Data Splitting Strategy:

- Training set: 70% (historical data from previous semesters)
- Validation set: 15% (for hyperparameter tuning)
- Test set: 15% (most recent semester data for final evaluation)

Hyperparameters to Tune:

1. **Learning rate** (0.01-0.3): Controls contribution of each tree, crucial for balancing bias-variance tradeoff
2. **Max depth** (3-10): Prevents overfitting while capturing complex student behavior patterns

4. Evaluation & Deployment (8 points)

Evaluation Metrics:

1. **Precision-Recall AUC:** More informative than ROC for imbalanced datasets (few dropouts vs many completions)
2. **F2-Score:** Emphasizes recall over precision to minimize false negatives (missing at-risk students)

Concept Drift: The phenomenon where the statistical properties of the target variable change over time, making the model less accurate. In education, this could occur due to curriculum changes, new teaching methods, or external factors like pandemic disruptions.

Monitoring Strategy: Implement periodic model performance checks (monthly) using recent student data and statistical tests like Kolmogorov-Smirnov to detect feature distribution shifts.

Technical Deployment Challenge: Real-time inference latency - The system must process new student data and generate predictions quickly enough for weekly interventions. Solution: Implement model serving with TensorFlow Serving or ONNX Runtime for efficient inference.

Part 2: Case Study Application (40 points)

Problem Scope (5 points)

Problem: Develop an AI system to predict 30-day patient readmission risk post-discharge.

Objectives:

1. Identify high-risk patients with 80% precision before discharge
2. Enable targeted care transition programs for high-risk patients
3. Reduce 30-day readmission rates by 15% within implementation year

Stakeholders:

- Patients and families
- Healthcare providers (doctors, nurses, care coordinators)
- Hospital administration and insurance payers

Data Strategy (10 points)

Data Sources:

1. Electronic Health Records (EHRs): Diagnosis codes, medication history, lab results
2. Patient demographics: Age, socioeconomic status, insurance type
3. Clinical notes: Discharge summaries, physician narratives (via NLP)
4. Previous admission history: Number of prior readmissions, length of stay

Ethical Concerns:

1. **Patient Privacy:** PHI protection requiring strict data anonymization and access controls
2. **Algorithmic Fairness:** Potential bias against elderly patients or those with complex chronic conditions who may be frequently readmitted for legitimate medical reasons

Preprocessing Pipeline:

python

```
# Pseudocode for preprocessing pipeline
1. Data Extraction: SQL queries from EHR database
2. Handling Missing Values:
   - Median imputation for lab values
   - Forward fill for time-series vital signs
   - Special category for missing categorical data
3. Feature Engineering:
   - Comorbidity scores (Charlson Index)
   - Medication complexity index
   - Social determinants of health proxies
   - Temporal features: seasonal admission patterns
4. Normalization: StandardScaler for continuous variables
5. Encoding: One-hot for categorical, label encoding for ordinal
```

Model Development (10 points)

Selected Model: Logistic Regression with L2 regularization

Justification:

- High interpretability crucial for clinical adoption
- Provides probability scores aligning with risk assessment needs
- Handles collinear features common in medical data
- Regulatory compliance easier with transparent models

Hypothetical Confusion Matrix:

text

```
Actual/Predicted | High Risk | Low Risk
High Risk        |    45     |    15
```

Low Risk | 20 | 120

Calculations:

- Precision = TP/(TP+FP) = 45/(45+20) = 0.692
- Recall = TP/(TP+FN) = 45/(45+15) = 0.75
- F1-Score = 2*(Precision*Recall)/(Precision+Recall) = 0.72

Deployment (10 points)

Integration Steps:

1. **API Development:** RESTful API wrapping the model for EHR integration
2. **Database Connectivity:** Secure connection to hospital data warehouse
3. **User Interface:** Integration with clinical decision support system
4. **Alert System:** Automated flags in patient discharge workflow
5. **Monitoring Dashboard:** Real-time model performance tracking

HIPAA Compliance Measures:

- Data encryption in transit (TLS 1.3) and at rest (AES-256)
- Role-based access control with audit trails
- Regular security assessments and penetration testing
- Business Associate Agreements with all vendors
- Data minimization - only necessary features extracted

Optimization (5 points)

Overfitting Mitigation Method: Stratified K-Fold Cross-Validation with early stopping

Implementation:

- Use 5-fold cross-validation ensuring each fold maintains readmission rate distribution
- Monitor validation loss plateau for early stopping
- Combine with L2 regularization (λ=0.1) to penalize large coefficients
- Feature selection using recursive feature elimination to remove redundant clinical variables

---

Part 3: Critical Thinking (20 points)

Ethics & Bias (10 points)

Impact of Biased Training Data:
Biased data could systematically underpredict readmission risk for minority populations or patients with rare conditions. For example, if training data primarily comes from urban academic medical centers, the model may perform poorly for rural patients with different healthcare access patterns. This could lead to:

- Disparities in care coordination resources
- Worsening health outcomes for underserved populations
- Reinforcement of existing healthcare inequalities

Bias Mitigation Strategy: Adversarial Debiasing

- Train the model to predict readmission while simultaneously minimizing the model's ability to predict protected attributes (race, insurance type)
- Use fairness constraints during optimization to ensure equal precision across demographic groups
- Implement regular bias audits using disparate impact analysis

Trade-offs (10 points)

Interpretability vs. Accuracy Trade-off:
In healthcare, interpretability often outweighs marginal accuracy gains. A logistic regression model with 82% accuracy that clinicians understand and trust is more valuable than a black-box neural network with 85% accuracy. The trade-off considerations:

- **Regulatory Requirements:** FDA approvals often require model explainability
- **Clinical Adoption:** Physicians need to understand prediction rationale for treatment decisions
- **Liability:** Clear reasoning trails are essential for medical-legal protection

Limited Computational Resources Impact:
With constrained resources, the hospital should prioritize:

1. **Model Selection:** Simpler models (logistic regression) over deep learning

2. **Feature Engineering:** Manual feature selection over automated methods
3. **Training Frequency:** Weekly retraining instead of real-time updates
4. **Ensemble Methods:** Avoid complex stacking in favor of single well-tuned models

---

Part 4: Reflection & Workflow Diagram (10 points)

Reflection (5 points)

Most Challenging Workflow Component: Data Quality and Integration

The healthcare case study revealed that real-world medical data is fragmented across multiple systems (EHRs, lab systems, billing databases) with inconsistent formatting and missing documentation. The challenge wasn't model development but ensuring reliable, clean data inputs.

Improvements with Additional Resources:

1. **Data Infrastructure:** Implement unified data lake with standardized ETL pipelines
2. **Domain Expertise:** Engage clinical informaticians earlier in feature engineering
3. **Longitudinal Validation:** Extend testing across multiple hospitals for generalizability
4. **Continuous Monitoring:** Deploy automated data quality checks and drift detection

```
AI DEVELOPMENT WORKFLOW
┌────────────────────┐
│ 1. PROBLEM │
│ DEFINITION │
└──────────┬─────────┘
↓
┌────────────────────┐
│ 2. DATA │
│ COLLECTION │
└──────────┬─────────┘
↓
┌────────────────────┐
│ 3. DATA │
│ PREPROCESSING │
└──────────┬─────────┘
↓
┌────────────────────┐
│ 4. MODEL │
│ DEVELOPMENT │
└──────────┬─────────┘
↓
┌────────────────────┐
│ 5. MODEL │
│ EVALUATION │
└──────────┬─────────┘
↓
┌────────────────────┐
│ 6. DEPLOYMENT │
│ & MONITORING │
└────────────────────┘

FEEDBACK LOOPS:
• Model performance → Retraining
• Production monitoring → Feature engineering
• User feedback → Problem refinement
```