Part 1: Predicting Graduate Employment Rates

1. Problem Definition

The core problem is predicting the likelihood of employment within 6 months after graduation for university students.

Objectives

1. Identify at-risk students early to provide targeted career support.

2. Help universities improve placement rates and refine curriculum design.

3. Assist students in making informed decisions about majors and career paths.

Stakeholders

1. University Career Services

2. Students and Parents

Key Performance Indicator (KPI)

Accuracy of employment prediction within ±10% of the actual placement rate per cohort.

2. Data Collection and Preprocessing

Data Sources

1. University Records (GPA, major, attendance, internships, extracurricular activities)

2. Alumni Survey and LinkedIn API (employment status, job title, company)

Potential Bias

Selection bias occurs when only alumni who respond to surveys or maintain active LinkedIn profiles are included. This overestimates employment rates for tech-savvy majors.

Preprocessing Steps

1. Handle missing data by imputing GPA with the median per major and dropping rows with missing employment status.

2. Encode categorical features using one-hot encoding for major and gender, and label encoding for job sector.

3. Normalize numerical features by scaling GPA, internship count, and attendance rate to [0,1] using MinMaxScaler

3. Model Development

Chosen Model: XGBoost Classifier

Justification: XGBoost handles mixed data types, captures non-linear relationships, remains robust to outliers, and provides feature importance insights.

Data Split

70% Training – to learn patterns

15% Validation – for hyperparameter tuning

15% Test – for final unbiased evaluation

Hyperparameters to Tune

1. max_depth – controls tree complexity and prevents overfitting

2. learning_rate – smaller values improve generalization with more trees

4. Evaluation and Deployment

Evaluation Metrics

1. F1-Score – balances precision and recall, critical when employed class is imbalanced

2. AUC-ROC – measures ranking ability across thresholds, robust to class imbalance

Concept Drift

Concept drift occurs when the statistical properties of the target variable or feature-target relationship change over time, reducing model accuracy on new data.

Post-Deployment Monitoring

Track model accuracy on new monthly cohorts.

Apply Kolmogorov-Smirnov test to detect feature distribution shifts.

Retrain if performance drops more than 5% for two consecutive months.

Technical Challenge in Deployment

Scalability – predicting for 10,000+ graduates annually requires batch processing and low-latency inference.

Solution: Deploy via FastAPI + Docker on AWS SageMaker with daily batch predictions.

Part 2: Case Study – Hospital Readmission Risk Prediction

Problem Scope

Problem Definition: Predict the probability of patient readmission within 30 days of discharge to enable early intervention and reduce costs.

Objectives

1. Reduce 30-day readmission rate by 50%

2. Prioritize high-risk patients for post-discharge follow-up

3. Optimize hospital resource allocation

Stakeholders

1. Hospital Administrators – reduce CMS HRRP penalties

2. Clinical Care Teams – improve patient outcomes

Data Strategy

Proposed Data Sources

1. Electronic Health Records (EHRs) – diagnosis codes, medications, labs, length of stay, discharge summary

2. Social Determinants – age, insurance type, zip code

3. Historical Admission Logs

Ethical Concerns

1. Patient Privacy (HIPAA) – risk of re-identification from rare diagnoses

2. Manual Health Records – handwritten summaries are hard to process

Preprocessing Pipeline

1. Impute missing lab results with median per diagnosis group

2. One-hot encode insurance type and discharge disposition

3. Feature engineering: num_comorbidities, readmission_history, social_risk_score (PCA on zip-code data)

4. StandardScaler on age, length of stay and lab values

Model Development

Selected Model: XGBoost Classifier

Justification: Handles mixed data, robust to missing values, outperforms logistic regression on tabular healthcare data.

Hypothetical Confusion Matrix (1,000 patients)

	Predicted: No	Predicted: Yes
Actual: No	780 (TN)	40 (FP)
Actual: Yes	60 (FN)	120 (TP)

Precision = 120/(120+40) = 0.75 (75%)

Recall = 120/(120+60) = 0.67 (67%)

F1-Score ≈ 0.71

Deployment

Integration Steps

Patient Discharged → Trigger API → Fetch EHR + Demographics → Preprocess → XGBoost Inference → Output Risk + SHAP → Alert EMR (Epic/Cerner) → Notify Case Manager

HIPAA Compliance

Data Encryption: TLS in transit, AES-256 at rest

Access Control: RBAC with audit logs

De-identification: Remove 18 HIPAA identifiers

BAA: Signed with cloud provider

Optimization

Method: Early Stopping + Cross-Validation

Early stopping halts training when validation loss stops improving, preventing overfitting.

Part 3: Critical Thinking

Ethics and Bias

Impact of Biased Data: Underrepresenting groups leads to underestimated readmission risk, fewer interventions, higher readmissions, and worsened health disparities.

Mitigation Strategy: Enforce fairness metrics (equal representation by race/income) during training.

Trade-offs

Interpretability vs. Accuracy: Complex models (neural networks) are accurate but black-box; interpretable models (logistic regression, trees) support clinical trust and compliance but may sacrifice performance.

Limited Resources: Use lightweight models (logistic regression) for real-time inference on edge devices, prioritizing speed and deplorability over marginal accuracy gains.

Part 4: Reflection

Most Challenging Part: Data Preprocessing and Client Privacy

Healthcare data has missing values, inconsistent codes, and handwritten notes. Strict privacy laws add complexity.

Improvements with More Resources

Collaborate with clinicians in workshops to validate features.

Build real-time data quality dashboards.

Implement federated learning across hospitals without sharing raw data.