
Part 1: Short Answer Questions

1. Problem Definition

Hypothetical AI Problem: Predicting student dropout rates in universities.

Objectives: 1. Identify students at high risk of dropping out early. 2. Recommend targeted interventions to reduce dropout. 3. Improve overall student retention rates.

Stakeholders: - University administrators - Academic advisors

Key Performance Indicator (KPI): Dropout prediction accuracy (%) or reduction in actual dropout rate over one academic year.

---

2. Data Collection & Preprocessing

Data Sources: 1. Student academic records (grades, attendance). 2. Student demographic and engagement data (age, major, participation in clubs).

Potential Bias: Socioeconomic bias: Students from disadvantaged backgrounds may have less representation or more missing data, skewing predictions.

Preprocessing Steps: 1. Handle missing data using imputation or removal. 2. Normalize numeric features (e.g., GPA) to a common scale. 3. Encode categorical variables (e.g., major, gender) using one-hot encoding.

---

3. Model Development

Chosen Model: Random Forest – robust for tabular data, handles missing values, interpretable feature importance.

Data Split: - Training set: 70% - Validation set: 15% - Test set: 15%

Hyperparameters to Tune: 1. Number of trees – affects model accuracy and overfitting. 2. Maximum depth of trees – controls complexity and generalization.

---

4. Evaluation & Deployment

Evaluation Metrics: - Accuracy: Measures overall correctness of predictions. - F1-Score: Balances precision and recall, useful for imbalanced dropout data.

Concept Drift: Change in student behavior patterns over time can make the model less accurate.

Monitoring: Regularly track performance metrics and retrain the model with new data.

Technical Challenge: Scalability: Ensuring the system can handle increasing numbers of students and data without slowing down predictions.

Part 2: Case Study Application

Problem Scope

Problem: Predict the risk of patient readmission within 30 days of discharge.

Objectives: 1. Reduce avoidable readmissions. 2. Optimize hospital resource allocation. 3. Provide early interventions for high-risk patients.

Stakeholders: - Hospital administrators - Healthcare providers (doctors, nurses)

Data Strategy

Data Sources: 1. Electronic Health Records (EHRs) – medical history, lab results, medications. 2. Demographics – age, gender, socioeconomic factors.

Ethical Concerns: 1. Patient privacy and data security. 2. Potential bias against vulnerable populations (e.g., elderly or low-income patients).

Preprocessing Pipeline: 1. Handle missing values (e.g., median imputation for lab results). 2. Feature engineering: - Create a "comorbidity score" from medical history. - Encode categorical features like diagnosis codes. 3. Normalize numeric features (age, lab values). 4. Split data into training/validation/test sets.

Model Development

Chosen Model: Gradient Boosting Classifier – handles imbalanced data well, interpretable, strong predictive power.

Confusion Matrix (Hypothetical Data): | | Predicted Readmit | Predicted No Readmit | |---------------|-----------------|-------------------| | Actual Readmit| 80 | 20 | | Actual No Readmit | 15 | 185 |

Metrics: - Precision = 80 / (80 + 15) ≈ 0.842 - Recall = 80 / (80 + 20) = 0.8

Deployment

Integration Steps: 1. Develop API endpoint for model predictions. 2. Integrate into hospital EHR system for real-time alerts. 3. Train medical staff on interpreting AI predictions. 4. Set up monitoring dashboard for model performance.

Compliance: Encrypt all patient data, anonymize where possible, and follow HIPAA guidelines for storage and access.

Optimization

Overfitting Solution: Apply cross-validation and regularization (e.g., limit tree depth, use early stopping in gradient boosting).

References

1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
2. Chollet, F. (2017). Deep Learning with Python. Manning.
3. HIPAA guidelines: https://www.hhs.gov/hipaa/index.html