MLAI Project Report 2025

Filippo Colella [03344], Camilla Marvaldi [03548], Lorenzo Licini [03884],
Riccardo Leonetti [03515], Parik Lanke [03536], Jahnavi Minocha [04497]

1 Introduction

Reducing hospital readmissions presents a persistent challenge in healthcare, particularly for diabetic patients who often require continuous and coordinated care. This project focused on addressing this challenge by developing predictive models using hospital administrative data. The primary goal was to proactively assess the likelihood of patient readmission, thereby supporting clinical decision-making by enabling early identification of high-risk cases and facilitating timely interventions.

Our efforts were concentrated on two parallel predictive tasks:

1. **Binary Classification:** Predicting early readmission (occurring in $<$30 days).
2. **Multi-level Classification:** Predicting early ($<$30 days), late ($>$30 days), or no readmission.

2 Data Overview

The dataset for this project was provided in CSV format, pre-organized into a training set and a test set (for final predictions). The data is understood to be a derivative of the "Diabetes 130-US Hospitals for Years 1999-2008" dataset by John Clore et al. [1]. The training dataset comprised over 79,183 unique hospital encounters, characterized by 49 features of various data types.

3 Methodology: Approach 1 - Traditional Machine Learning

Our initial methodological workflow for developing traditional machine learning models involved a sequence of well-defined steps:

3.1 Initial Dataset Inspection and Feature Dropping

A preliminary overview of the dataset was conducted. Analysis of missing values (NaNs) revealed:

- High missingness ($>$50% or even $>$90%) in features like `max_glu_serum`, `A1Cresult`, `medical_specialty`, and `payer_code`.
- Moderate missingness (approx. 2% to 10%) in others, e.g., `admission_type_id`.

Based on this and domain considerations, feature dropping decisions included:

- `medical_specialty`: Dropped, assuming outcome is determined by diagnosis features.
- `payer_code`: Dropped, assuming no direct influence on clinical outcomes for this task.
- `A1Cresult` and `max_glu_serum`: Removed from modeling data due to high missingness (to avoid significant data reduction), but retained temporarily for EDA.
- Low-Prevalence Drug Features: Over 10 features representing drugs given to a very small percentage ($<1\%$) of patients were dropped.

This resulted in the final feature set for analysis, with a copy including A1C and max glucose serum kept for raw EDA.

3.2 Features Encoding

Prior to EDA, features were encoded:

1. **Target Variable Creation:** An `early_readm` column (binary target) was created: `1` for <30 days readmissions, `0` for No or >30 days readmissions.
2. **General Encoding:** All features transformed into integers, preserving ordinality where relevant (e.g., binned age intervals).
3. **Diagnosis Columns (`diag_1`, `diag_2`, `diag_3`):** These had 700+ unique levels. The top 60 were encoded (0-59), others grouped as "other" (60). Diagnoses were then aggregated into 11 taxonomical macro-areas, resulting in 12 unique levels (11 areas + "Other" mapped as 60).
4. **Type Map:** Ensured each feature was treated correctly (integer, ordinal, nominal).

3.3 Exploratory Data Analysis (EDA)

EDA was structured as follows:

- **Correlation Heatmaps:** Generated twice (with/without A1C & max glucose serum), using appropriate tests (Cramer's V, Spearman, Theil's U) based on feature types. NaNs handled by the function.
- **Pairwise Mutual Information (MI):** Calculated for features against both targets (binary, multiclass) on both dataset versions; results plotted. MI was not used for feature selection at this stage.
- **Distribution Analysis:** Visual and numerical inspection of target distributions and feature distributions stratified by target levels (included A1C & max glucose serum).
- **Univariate Statistical Tests:** Comprehensive pairwise univariate analysis computing p-values for feature-target associations (using appropriate tests like Chi-squared, ANOVA, based on assumptions). P-values adjusted with Bonferroni correction. (Potential limitation: assumes statistical independence between observations).

3.4 Preprocessing Pipelines

A feature-specific preprocessing pipeline was designed:

- **Continuous Features:**
  - Imputation: k-Nearest Neighbors for features with $>10\%$ missing values; median imputation for features with lower missingness ($<10\%$).
  - Outlier Handling: Isolation Forest to identify outliers; values clipped to 1st/99th percentiles.
  - Transformation: Yeo-Johnson for skewness.
  - Scaling: Standardization.
- **Categorical Features:**
  - Ordinal: Ordinal Encoding (e.g., age groups).
  - Nominal: One-Hot Encoding (e.g., race, gender).

All preprocessing steps were encapsulated within scikit-learn model pipelines.

3.5 Model Development Process

A systematic five-phase approach:

1. **Phase 1: Base Classifier Evaluation:** Five algorithms (LightGBM, XGBoost, Random Forest, Logistic Regression, Naive Bayes) evaluated using 5-fold stratified cross-validation. Matthews Correlation Coefficient (MCC) was the primary metric.

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

2. **Phase 2: Ensemble Size Optimization:** Ensembles built using top two or three performing algorithms from Phase 1 to determine optimal ensemble size.
3. **Phase 3: Ensemble Method Selection:** Compared soft voting (combining probabilities) and stacking (Logistic Regression meta-learner).
4. **Phase 4: Class Imbalance Handling:** SMOTE-Tomek hybrid approach evaluated [Oversampling vs No Oversampling], applied only within training folds (50% augmentation for minority classes either one or both minorities).
5. **Phase 5: Final Optimization:**
   - Binary task: Decision threshold optimization to maximize MCC (CV).
   - Multiclass task: Bayesian optimization (Optuna, 25 trials, Tree-Parzen Estimator sampler) on 80% of data to tune hyperparameters for the ensemble.

3.6 Evaluation Framework, Explainability, and Final Model

All model selections used cross-validation on training data; final performance on untouched hold-out sets. Micro/macro ROC curves for multiclass. Stratified sampling ensured consistent class distributions. Explainability included SHAP Summary plots per target class (3). The best final multiclass model (with tuned hyperparameters) was saved and used for predictions on *diabetes test.csv*.

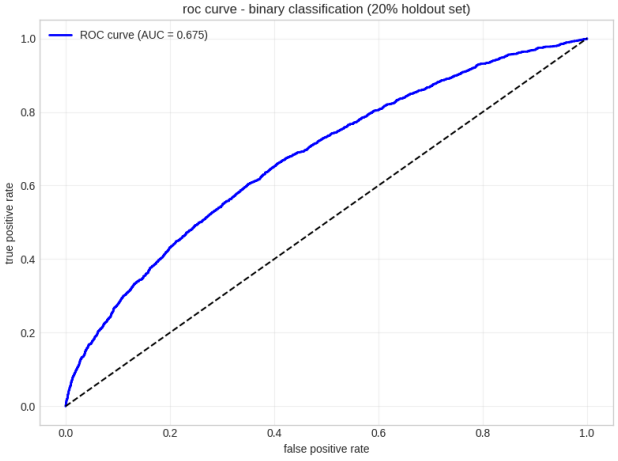

Figure 1: Binary model with tuned threshold performance

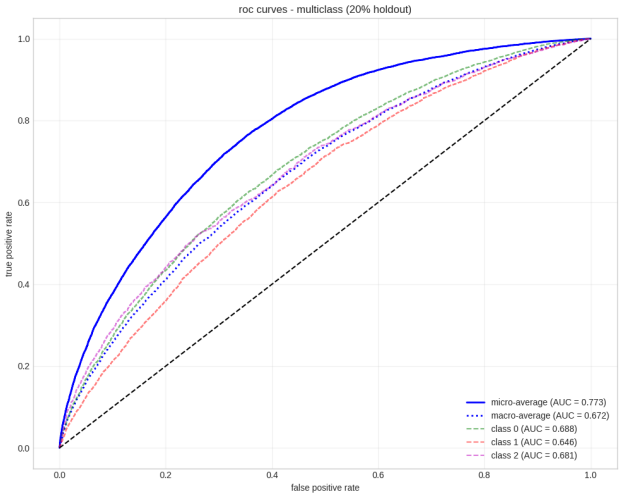

Figure 2: Multiclass tuned model performance

4 Methodology: Approach 2 - Graph Neural Networks (Multi-class Task Only)

This approach started from `df_model` (from Approach 1) and focused solely on the multi-level classification task.

4.1 I. Project Initialization and Data Preparation

Established a reproducible environment (libraries: `torch_geometric`, pandas, scikit-learn, PyTorch; parameters, seeds). Loaded data, performed preliminary feature culling. Conducted compact EDA (target distribution, missingness, encounter frequencies). A key step was engineering a new feature for the sequence of encounters per patient, known both at training and inference (and possibly real world) stages. The dataset was split at the patient level (train, validation, evaluation, hold-out) to prevent data leakage. A scikit-learn `ColumnTransformer` preprocessed data (imputation: median/KNN for numerical, most frequent for categorical; scaling: StandardScaler; encoding: OneHotEncoder), fitted only on training data. Class weights were computed from training data for imbalance.

4.2 II. Model Development and Training

Baseline models (Logistic Regression, Random Forest), accounting for class imbalance, were trained on preprocessed tabular data. **Graph Construction:** For each data split, encounters became nodes; edges linked sequential encounters for the same patient (plus self-loops). Preprocessed features served as node attributes. Graphs were encapsulated in PyTorch Geometric `Data` objects. **GNN Architecture & Training:** A GNN (e.g., SAGEConv layers) was defined. It was trained iteratively on training graph data (AdamW optimizer, class-weighted cross-entropy loss). Performance monitored on validation graph data; learning rate scheduler and early stopping (validation F1-score) used. Training history was visualized.

4.3 III. Evaluation and Final Model Selection

Rigorous evaluation on the dedicated evaluation set for all models: baselines, primary GNN, GNN variant (self-loops only), and an ensemble (baseline + GNN). Metrics included: accuracy, F1-macro, MCC, ROC AUC. Visualizations: MCC histograms, ROC curves. Best model selected (primarily MCC); its confusion matrix generated. Brief interpretability: GNN (gradient-based feature importance), Logistic Regression (coefficients). The selected best model was then tested on the untouched hold-out set.

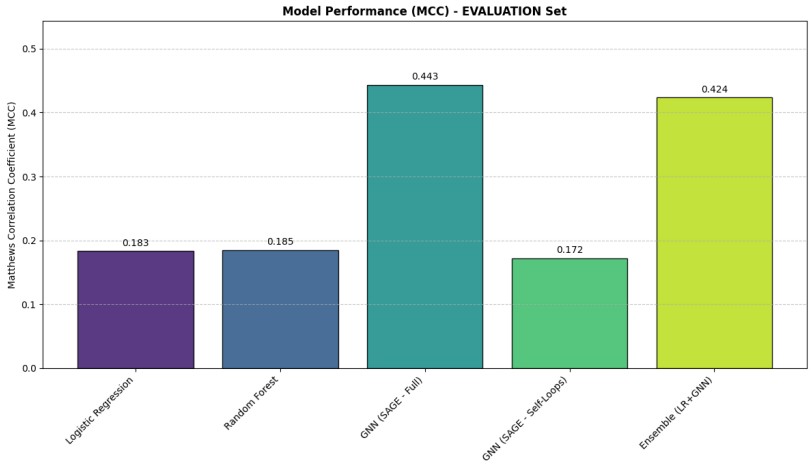

Figure 3: Model evaluation comparison on MCC

4.4 IV. Final Model and Inference Pipeline

The chosen GNN architecture was re-trained on an expanded dataset (all data except hold-out) to create the ”final GNN model,” then saved. A final sanity check was performed on the hold-out set. A clear inference pipeline was defined: load new raw data, apply saved preprocessor, construct graph, load production GNN, generate predictions, format output.

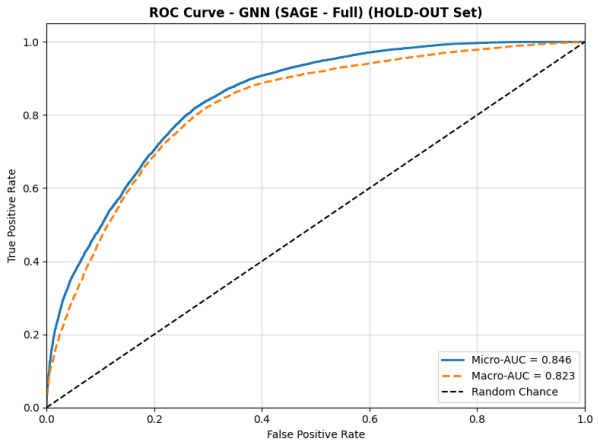

Figure 4: Micro and Macro Averaged AUC for the final best model (GNN SAGE)

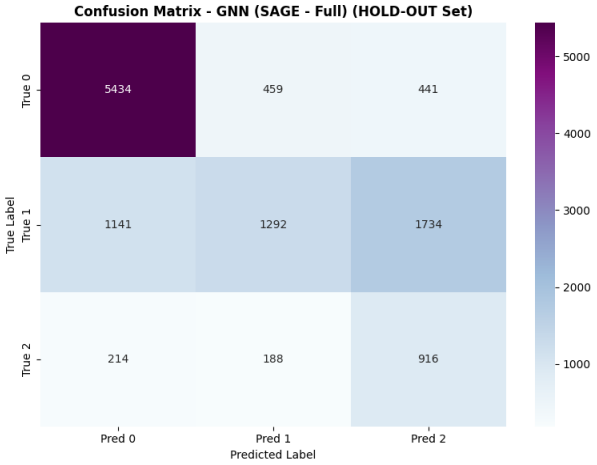

Figure 5: Confusion Matrix for the best model (GNN SAGE)

5 Conclusion (Implicit)

This project successfully developed and evaluated two distinct methodological approaches for predicting diabetic patient readmissions. Approach 1 established robust traditional machine learning pipelines, culminating in an optimized ensemble model. Approach 2 explored the advanced capabilities of Graph Neural Networks, creating a production-ready GNN model capable of leveraging sequential patient encounter data. Both approaches demonstrate viable pathways for identifying high-risk patients, with the GNN offering a novel way to incorporate the temporal dynamics of patient care.

6 References

1 Clore, J., Cios, K., DeShazo, J., & Strack, B. (2014). Diabetes 130-US Hospitals for Years 1999-2008 [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5230J.