Beyond Tabular Data: A Comparative Framework for Graph Learning in Disease Prediction

Coursework Project

Submitted in partial fulfillment of the requirements of
BMI/CS 775 (Computational Network Biology)

By

Avyakt Garg
ID No. 908-876-9924

Under the supervision of:

Prof. Anthony Gitter and Prof. Sushmita Roy

UNIVERSITY OF WISCONSIN-MADISON
December 2025

Contents

List of Figures

Chapter 1

Introduction

Heart disease is known to be the leading cause of death across the United States [5]. The CDC reports that one person dies from cardiovascular disease every 34 seconds; these statistics underscore the urgent need for quick and accessible methods to screen individuals based on medical data. Machine learning has proven to be a key tool in addressing this challenge. However, a major pitfall of traditional machine learning models is the assumption that data points are independent and identically distributed (i.i.d.). This assumption treats each entry in a tabular dataset as an isolated instance. In reality, these entries represent patients who often share latent relationships and similarities, such as shared risk factors, which traditional models might fail to capture.

Network biology offers a robust alternative by modeling these clinical relationships as graphs, enabling algorithms to learn from both individual attributes and the network topology. This project applies concepts from CS 775 to transform tabular clinical data into graph structures. A central question in this domain is determining which graph construction method and learning algorithm yields the best performance for disease prediction. To answer this, we implemented two distinct graph construction methods: a Patient Similarity Network (PSN) and a Bipartite Patient-Attribute Graph. Furthermore, we evaluated two learning algorithms across these structures: Node2Vec [3] and GraphSAGE [4].

We evaluated these four resulting pipelines using the UCI Heart Disease dataset [6]. Our evaluation extended beyond standard performance metrics; recognizing the sensitivity of medical applications, we prioritized interpretability to "look under the hood" of our models and establish clinical trust by analyzing feature importance using GNNExplainer[11]. Finally, to bridge the

gap between theoretical modeling and real-world utility, we deployed one of our best-performing model as a full-stack web application. This application features an intuitive user interface and serves as a proof of concept for the practical application of network biology in disease prediction.

Chapter 2

Related Work

Various studies have successfully applied traditional machine learning algorithms to disease prediction. For the specific dataset used in this project, traditional models are currently more prevalent in the literature compared to graph-based methods. However, a recent benchmark study [8] cautions against prior works that report highly inflated metrics, often resulting from oversampling or overfitting. This study provides a rigorous baseline for the UCI dataset, allowing us to accurately benchmark our model and motivating the need for robust validation strategies to prevent overfitting.

Regarding graph-based approaches, Wajgi et al. [10] recently applied Graph Neural Networks to the heart disease dataset. Their work focused on testing different optimization algorithms, reporting a high test accuracy of 92%. While these results are promising, the authors identified the comparison of different GNN models as a key area for future work, which directly aligns with our approach of evaluating different graph structures. Furthermore, while high accuracy is essential, we emphasize that clinical acceptance relies on model interpretability. This gap in existing studies motivated our use of GNNExplainer to validate the medical relevance of our model's predictions.

Similarly, a recent study by Boll et al. [1] utilized data from Electronic Health Records to construct a Patient Similarity Network (PSN), creating edges between patients with similar clinical features. While they compared three GNN architectures, they highlighted the exploration of heterogeneous graphs as a direction for future research. This supports our specific interest in creating a bipartite patient-attribute graph to compare its performance against standard homogeneous patient networks.

Chapter 3

Approach

Note : AI tool Gemini 3 was used to assist in coding but the approach and decisions were not taken by the AI tool it was used just for the purpose of understanding syntax and debugging errors

3.1 Dataset

The dataset utilized in this study was the publicly available UCI Heart Disease dataset [6]. It originally consists of 303 patient entries, characterized by 13 clinical features and 1 target variable (see Table 3.1). While the dataset was largely pre-cleaned, it contained a small number of records with missing values, which were removed to yield a final sample size of 297 patients. The scope of this project was to build a binary classifier; consequently, the original multi-class target attribute (values 0–4) was converted into a binary format, where 0 indicates the absence of disease and 1–4 indicates its presence. To facilitate stable model training, Z-score normalization was applied to standardize the input features. Crucially, this scaling was performed strictly within the cross-validation loops to prevent data leakage between training and validation sets.

3.2 Graph Construction Frameworks

To systematically evaluate the impact of topology on predictive performance, we employed two distinct approaches for graph construction: a standard homogeneous Patient Similarity Network(PSN) and a heterogeneous Bipartite Patient-Attribute Graph.

TABLE 3.1: UCI Heart Disease Dataset Features and Data Types

Attribute	Description	Data Type
age	Patient's age	Integer
sex	Gender (1 = male, 0 = female)	Categorical
cp	Chest pain type	Categorical
trestbps	Resting blood pressure (mm Hg)	Integer
chol	Serum cholesterol (mg/dl)	Integer
fbs	Fasting blood sugar (> 120mg/dL)	Categorical
restecg	Resting ECG results	Categorical
thalach	Max heart rate achieved	Integer
exang	Exercise induced angina	Categorical
oldpeak	ST depression induced by exercise	Integer
slope	Slope of peak exercise ST	Categorical
ca	Major vessels colored by fluoroscopy (0–3)	Integer
thal	Thalassemia disorder	Categorical
target	Diagnosis of patient	Integer

3.2.1 Patient Similarity Network (PSN)

In the PSN, each node represents an individual patient from the dataset. A key challenge in constructing this network was the mixed nature of the data, which contains both continuous and categorical variables (as shown in Table 3.1). Standard distance metrics like Euclidean distance are ill-suited for such mixed data types. To address this, we utilized Gower's Distance [2], a similarity coefficient specifically designed for mixed-type data. The resulting fully connected graph was sparsified for computational efficiency using a k-Nearest Neighbors (k-NN) approach, where patients were connected only to their k most similar neighbors. This sparsification reduces noise in the graph structure. The value of k was treated as a structural hyperparameter and optimized during the validation phase.

3.2.2 Bipartite Patient-Attribute Graph

In the second approach, we constructed a heterogeneous graph consisting of two distinct node sets: Patients and Attributes. Edges were created strictly between a patient and an attribute node if the patient possessed that specific feature value. While mapping categorical features to attribute nodes was straightforward, continuous features presented a challenge. To resolve this, we implemented dynamic binning, where continuous variables (e.g., Age) were discretized into ranges (e.g., "Age 60–67"). The number of bins was not static but treated as a tunable hyperparameter. This structure allows the model to explicitly preserve patient-feature relationships and minimizes information loss compared to simple similarity scores.

3.3 Learning Algorithms

To evaluate the impact of different learning paradigms, we compared two distinct approaches: unsupervised feature extraction via Node2Vec [3] coupled with a traditional classifier, and end-to-end supervised learning using the inductive GraphSAGE framework [4].

3.3.1 Unsupervised Representation Learning (Node2Vec)

First, we implemented Node2Vec as a baseline for unsupervised representation learning. This algorithm generates low-dimensional node embeddings by simulating biased random walks across the graph structure . These embeddings were subsequently used as input features for an XGBoost classifier, we selected it as it as has shown great performance on our tabular dataset as mentioned by the dataset's website. The random walk behavior is governed by two hyperparameters: the return parameter p, which encourages local exploration (returning to the source), and the in-out parameter q, which facilitates structural exploration further away from the start node.

3.3.2 Inductive Graph Neural Networks (GraphSAGE)

Secondly, we employed GraphSAGE, an inductive variant of Graph Convolutional Networks (GCNs). Unlike transductive methods that require the entire graph during training, GraphSAGE learns to generate embeddings for previously unseen nodes by sampling and aggregating features from their local neighborhoods . This inductive capability allows the model to generalize to new data points without reconstructing the entire graph. This inductive capability also allows us to build a web application on top of it. Our specific architecture utilized two `SAGEConv` layers to capture neighborhood information. The final node embeddings were passed through a fully connected linear layer with a Log-Softmax activation function to output the probability of disease. Key model hyperparameters, including the hidden dimension size and dropout rate, were optimized via grid search, as detailed in the following subsection.

3.4 Experiment Design

As discussed in previous sections, our proposed pipelines involved multiple hyperparameters requiring optimization. Furthermore, prior literature highlighted the prevalence of inflated

metrics in this domain, often stemming from small dataset sizes and the consequent risk of overfitting. To address these challenges, we employed a **Strict Nested Cross-Validation** protocol. This method ensures a rigorous separation between model selection (inner loop) and performance evaluation (outer loop).

The validation structure consisted of an outer loop with $k = 10$ folds and an inner loop with $k = 5$ folds. In the outer loop, the data was partitioned into 10 subsets; in each iteration, nine were allocated for training/tuning and one was strictly held out for final testing. Within the training allocation, the inner loop further partitioned the data into five folds (4 for training, 1 for validation). This inner mechanism facilitated a comprehensive grid search across all defined hyperparameter combinations.

Crucially, this protocol enforced a strict separation of datasets, preventing data leakage and mitigating the risk of "lucky splits." By constructing the graph structure independently within each fold, we ensured that the test set did not influence the training topology. Following the final evaluation, we reported a comprehensive suite of metrics for all four models, including Area Under the Precision-Recall Curve (AUC-PR), Recall, Specificity, F1-Score, Accuracy, and Area Under the ROC Curve (AUC-ROC).

3.5 Interpretability and Deployment

Beyond quantitative metrics, clinical adoption requires establishing trust by interpreting the model's decision-making process ("opening the black box"). To address this, we applied **GNNExplainer** [11], a model-agnostic explanation technique. GNNExplainer employs an edge-masking strategy to identify the most dominant subgraphs and features responsible for a specific high-risk prediction, allowing for the validation of the model against established medical knowledge.

Finally, to bridge the gap between theoretical modeling and real-world utility, our optimal model was deployed as a full-stack web application. The system features an intuitive, user-friendly frontend built with **Streamlit** [9], which communicates with a high-performance backend API developed using **FastAPI** [7]. This architecture allows for the secure input of new patient data—processed in real-time without permanent storage—and returns a probabilistic risk assessment, demonstrating the feasibility of inductive graph learning in clinical settings.

Chapter 4

Results

4.1 Performance Comparison of Graph Pipelines

We evaluated the four proposed pipelines using our experiment methodology to ensure robust performance estimation. Table 4.1 summarizes the average metrics across all folds.

The Patient Similarity Network (PSN) with GraphSAGE achieved the highest overall Recall (0.860), F1 Score (0.827), Area Under the ROC Curve (0.913) and Accuracy (0.835), demonstrating the efficacy of direct patient-to-patient similarity in capturing homogeneous population structures. However, the Bipartite GraphSAGE remained highly competitive, achieving the highest Area Under the Precision-Recall Curve (AUC-PR: 0.908). Though there is a marginal gap between both of them still we consider it as a highly competitive model due to its high interpretability nature. While with PSN we can get due to which other patients did we get a particular prediction but with bipartite graphs our interpretation is with respect to the attributes that contributed towards it.

We also observe that the across both graphs GraphSAGE performed better than the node2vec learning algorithms. They reported higher on all 6 metrics in Bipartite graphs while in PSN they reported higher on all but one metric - specificity which only differed by a margin of 0.007.

Standard decision thresholds (0.5) yielded sub-optimal recall for a medical screening task. And therefore to improve it we reduced the threshold to 0.35.

We also generated graphs relevant from a medical perspective to compare our models refer figure 4.1. We observe that the blue(PSN+GCN(SAGE)) and the green(BIPARTITE+GCN(SAGE))

TABLE 4.1: Comparative performance metrics of the four graph-based pipelines (Threshold = 0.35). Best values are highlighted in bold.

Model	AUC-PR	Recall	Specif.	F1	Acc.	AUC-ROC
PSN_GSAGE	0.907	**0.860**	0.812	**0.827**	**0.835**	**0.913**
PSN_N2V	0.877	0.802	**0.819**	0.797	0.811	0.880
BIPARTITE_GSAGE	**0.908**	0.824	0.781	0.793	0.801	0.910
BIPARTITE_N2V	0.849	0.802	0.781	0.777	0.791	0.871

lines were mostly performing better than others, and that within themselves they had very close competition. The curves also showed that our model wasn't just guessing. The staircase style graph in precision-recall curve was also observed which we will discuss further in the discussion section.

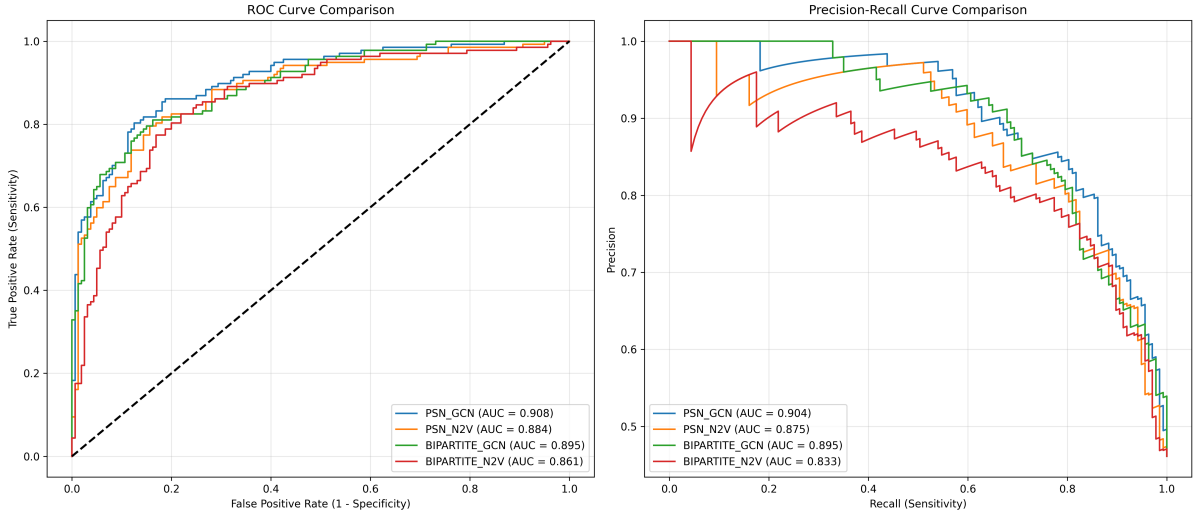

FIGURE 4.1: ROC Curves (Left) and Precision-Recall Curves (Right) for all four models. The GCN-based models (Blue, Green) consistently enclose the Node2Vec baselines (Orange, Red), demonstrating superior ranking capability.

4.2 Feature Importance and Clinical Validation

To validate the clinical utility of the Bipartite model, we applied GNNExplainer to extract the subgraphs driving high-risk predictions. As shown in Figure 4.2, the model identified 5 major contributors of the prediction. Specifically, for a representative high-risk patient, the model assigned the highest importance scores to fasting blood sugar and ST depression induced by exercise. This interpretability, enabled by the Bipartite structure, provides a level of transparency that the opaque similarity scores of a PSN cannot match.

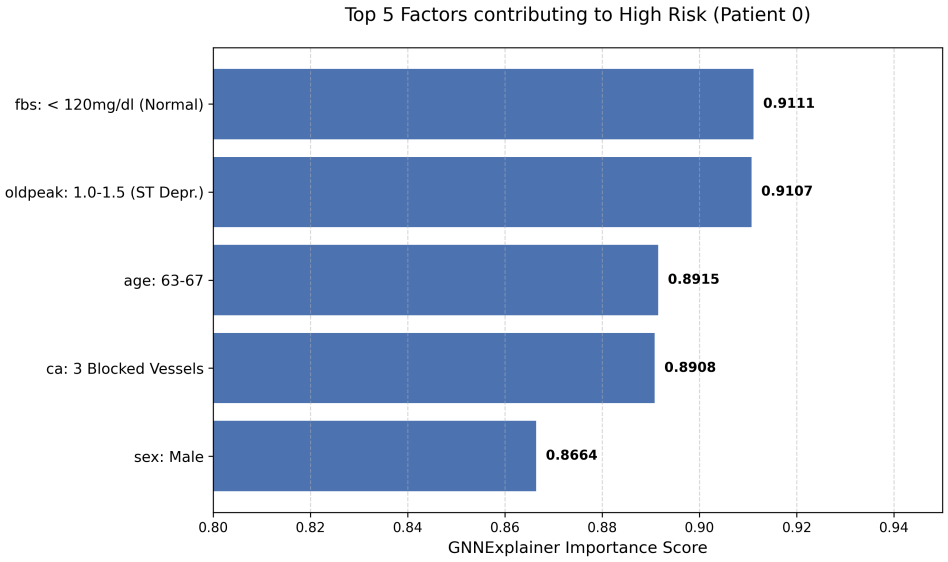

FIGURE 4.2: Top 5 clinical factors contributing to a high-risk prediction for Patient 0, identified via GNNExplainer.

Chapter 5

Discussion

5.1 Graph Topology: The Trade-off Between Accuracy and Explainability

Our results highlight a fundamental trade-off in network medicine: the balance between predictive power and clinical transparency. The Patient Similarity Network (PSN) achieved marginally higher raw performance (Recall: 0.860), likely because direct patient-to-patient connections capture latent population structures—such as shared phenotypic clusters—that are difficult to model explicitly. However, the PSN operates as a "black box"; a prediction based on similarity to "Patient 35" offers little actionable insight to a clinician. In contrast, the Bipartite model (Recall: 0.824) offers superior clinical utility through feature-level explainability. As detailed in the GNNExplainer analysis, the model successfully distinguished between structural graph properties and clinical pathology. While 'Fasting Blood Sugar < 120' (Normal) was identified as a top factor likely due to its role as a high-degree "hub" facilitating message passing, the model simultaneously prioritized highly specific risk factors. 'ST Depression' (Oldpeak) and 'Major Vessels Colored' (ca) were correctly identified as primary drivers of risk, aligning with medical literature that establishes fluoroscopy and ECG abnormalities as definitive indicators of coronary artery disease. Furthermore, the identification of demographic factors such as 'Age: 63-67' and 'Sex: Male' confirms that the model correctly learned the epidemiological baseline risks associated with older male populations in this dataset.

5.2 Benchmarking Against State-of-the-Art

Our best-performing model (PSN-GCN) achieved an accuracy of **83.5%**, strictly surpassing the rigorous benchmark of **83.3%** established by Padilla Rodriguez et al. (2024) using centralized Support Vector Machines [8]. This result validates that modeling patient relationships via Graph Neural Networks can capture predictive signals that traditional independent classifiers miss. We note that recent work by Wajgi et al. (2024) reported higher accuracies (up to 92%) using GNNs [10]. However, this discrepancy is attributable to experimental design. Wajgi et al. utilized an aggregated dataset combining four distinct cohorts (Cleveland, Hungary, Switzerland, Long Beach), whereas our study strictly adhered to the Cleveland benchmark (n=297) to ensure consistency with historical baselines. Furthermore, our use of strict nested cross-validation provides a more conservative, but generalizable, estimate of performance compared to standard train-test splits which are prone to optimization bias.

5.3 ROC and Precision-Recall Analysis

A critical reading of the model performance curves (Figure 4.1) reveals a clear hierarchy in predictive capability.

ROC Analysis: The Receiver Operating Characteristic (ROC) curves arch steeply toward the top-left corner, approaching the ideal "elbow" point of perfect classification. With AUC scores ranging from 0.861 to 0.913, all models demonstrate strong separability between high-risk and low-risk patients. Notably, both GCN-based models achieved AUCs > 0.90, classifying their performance as outstanding for this domain.

Precision-Recall Analysis: The Precision-Recall (PR) curves further validate this robustness. A distinct architectural hierarchy emerged: the Graph Neural Networks (GCNs) (Blue and Green lines) consistently enclosed the area of the Node2Vec baselines (Orange and Red lines), confirming that end-to-end supervised learning captures disease signals more effectively than static embeddings.

Sample Size Artifacts: It is important to note the "staircase" or sawtooth pattern observed in the PR curves (Figure 4.1, Right). This visual artifact is characteristic of evaluating on small datasets (n=297). Since the test set in each cross-validation fold contains approximately 30 patients, a single misclassification results in a visible vertical drop in precision. Despite this

high-variance evaluation environment, the GSAGE models maintained a high precision envelope, demonstrating robustness even when data is scarce.

5.4 Deployment: End-to-End Risk Assessment Application

To demonstrate the practical utility of our inductive framework, we successfully deployed the Bipartite GSAGE as a real-time web application (Figure 5.1). Unlike transductive graph models that require retraining to recognize new nodes, our system leverages the Inductive GraphSAGE architecture to generate embeddings for new patients on-the-fly ($O(1)$ inference time).

The application features a user-friendly interface built with **Streamlit**, allowing clinicians to input standard physiological parameters (e.g., Age, Cholesterol, Thal) via interactive sliders and dropdowns. These inputs are sent to a FastAPI backend, which dynamically maps the new patient to the existing bipartite graph structure and executes the inference pipeline. As shown in Figure 5.1, the system outputs a probabilistic risk assessment and a binary classification, providing an accessible tool for rapid screening without requiring technical expertise in graph neural networks.

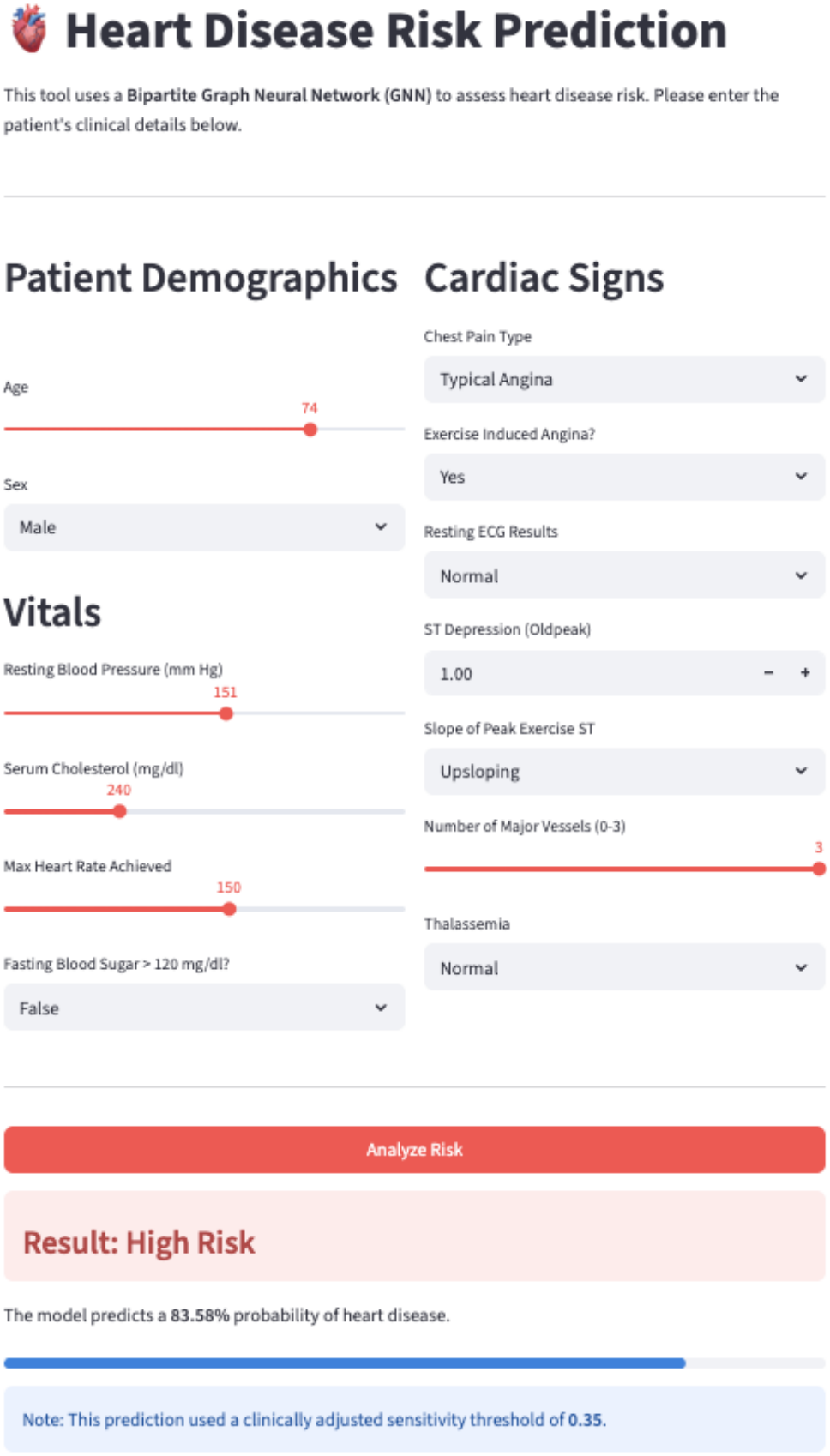

FIGURE 5.1: Interface of the Heart Disease Risk Prediction Web App.

Chapter 6

Future Work

The primary limitation of this study was the reliance on a single, small-scale dataset (Cleveland, n=297). Future work will aim to aggregate data from diverse geographical sources to test the model's generalization across populations. Our results also demonstrated a performance trade-off: the Patient Similarity Network (PSN) yielded higher recall, while the Bipartite graph offered superior explainability. Future research will explore Hybrid Graph Ensembles to achieve "the best of both worlds."

The Bipartite structure developed in this project provides a flexible foundation for integrating multi-modal data. Currently, the graph is limited to tabular clinical attributes. Future iterations could expand the node types to include unstructured data sources, such as billing codes (ICD-10), medication history, or genomic markers. Representing these diverse data types as distinct node classes in the graph would allow the GNN to learn complex, non-linear interactions that traditional tabular models cannot capture.

While the technical feasibility of our web application has been established, its clinical utility remains to be validated. A crucial next step is to conduct a usability study with medical professionals. Collecting qualitative feedback on the application's interface and the clarity of the GNNExplainer outputs will be essential for refining the tool. Furthermore, we aim to investigate the integration of our inference API directly into Electronic Health Record (EHR) systems moving the project from a standalone prototype to a production ready system

Bibliography

[1] Heloisa Oss Boll, Stefan Byttner, and Mariana Recamonde-Mendoza. "Graph Neural Networks for Heart Failure Prediction on an EHR-Based Patient Similarity Graph". In: *Anais Estendidos do XXV Simpósio Brasileiro de Computação Aplicada à Saúde (SBCAS 2025)*. SBCAS Estendido 2025. Sociedade Brasileira de Computação (SBC), June 2025, 121–126. DOI: 10.5753/sbcas_estendido.2025.7013. URL: http://dx.doi.org/10.5753/sbcas_estendido.2025.7013.

[2] John C. Gower. "A general coefficient of similarity and some of its properties". In: *Biometrics* 27.4 (1971), pp. 857–871. DOI: 10.2307/2528823.

[3] Aditya Grover and Jure Leskovec. "node2vec: Scalable Feature Learning for Networks". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 855–864. URL: https://arxiv.org/abs/1607.00653.

[4] William L. Hamilton, Rex Ying, and Jure Leskovec. "Inductive Representation Learning on Large Graphs". In: *Advances in Neural Information Processing Systems (NIPS)*. 2017. URL: https://arxiv.org/pdf/1706.02216.

[5] *Heart Disease Facts*. CDC. URL: https://www.cdc.gov/heart-disease/data-research/facts-stats/index.html (visited on 12/01/2025).

[6] Andras Janosi et al. *Heart Disease*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C52P4X. 1989.

[7] Sebastián Ramírez. *FastAPI: High performance, easy to learn, fast to code, ready for production*. Available at https://fastapi.tiangolo.com/. 2024.

[8] Mario Padilla Rodriguez and Mohamed Nafea. *Centralized and Federated Heart Disease Classification Models Using UCI Dataset and their Shapley-value Based Interpretability*. 2024. arXiv: 2408.06183 [cs.LG]. URL: https://arxiv.org/abs/2408.06183.

[9] Streamlit Inc. *Streamlit: The fastest way to build and share data apps*. Available at `https://streamlit.io/`. 2024.

[10] Rakhi Wajgi et al. "Heart Disease Prediction using Graph Neural Network". In: *International Journal of Intelligent Systems and Applications in Engineering* 12.12s (2024), 280–287. URL: `https://ijisae.org/index.php/IJISAE/article/view/4514`.

[11] Rex Ying et al. *GNNExplainer: Generating Explanations for Graph Neural Networks*. 2019. arXiv: `1903.03894 [cs.LG]`. URL: `https://arxiv.org/abs/1903.03894`.